



UNIVERSIDAD FRANCISCO DE VITORIA

ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA MATEMÁTICA

PROYECTO FINAL DE GRADO

MODALIDAD INVESTIGACIÓN

**ANÁLISIS DE IMPACTO DE LA
CRIPTOGRAFÍA POSTCUÁNTICA EN
ENTORNOS EMPRESARIALES**

Ana Robledano Abasolo
Convocatoria de julio 2025

CALIFICACIÓN DEL PROYECTO FINAL DE GRADO

CUALITATIVA:	
NUMÉRICA:	

Conforme Presidente:	Conforme Secretario:
Fdo.:	Fdo.:

Conforme Vocal:	Conforme Vocal:	Conforme Vocal:
Fdo.:	Fdo.:	Fdo.:

Lugar y fecha: Pozuelo de Alarcón, a 4 de julio de 2025

“Harvest now, decrypt later”

“The math works. It's the implementation that breaks.”

“Post-quantum cryptography is solving tomorrow's problems with today's tools.”

– *Conocimientos generales en criptografía*

A mis abuelos, por su apoyo incondicional.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a IBM, por darme la oportunidad de formar parte de su equipo de ciberseguridad, un entorno de alto nivel técnico y humano. Esta beca ha sido clave para poder desarrollar este trabajo. En especial, agradezco a José Cándido Carballido y Ginés Carrascal por acompañarme durante el proceso, compartiendo conmigo su profundo conocimiento en computación cuántica y ciberseguridad, así como su paciencia infinita para responder preguntas que solo ellos saben responder.

A mi tutora de prácticas, Raquel López Ruiz, por su apoyo y disposición durante mi formación, y por haber contribuido a mi desarrollo profesional en esta etapa.

Agradezco también a Jorge Andrés Plazas, mi profesor de criptografía, por introducirme al fascinante mundo de las retículas, los algoritmos que protegen (o comprometen) nuestros secretos digitales y por presentarme a Bob y a Alice.

Y, por supuesto, a mi familia, por su apoyo constante y por compartir conmigo el entusiasmo por los dilemas cuánticos, como el de ese gato que está vivo y muerto a la vez.

Resumen

El presente proyecto de investigación tiene como objetivo analizar el impacto de la implementación de mecanismos de encriptación postcuántica en entornos empresariales, evaluando su viabilidad, implicaciones técnicas y efectos en la seguridad.

Para llevar a cabo este estudio, se han implementado algoritmos de encriptación postcuánticos estandarizados por el National Institute of Standards and Technology (NIST) dentro del protocolo de seguridad Transport Layer Security (TLS), que permite establecer una conexión cifrada entre cliente y servidor.

El estudio incluye el despliegue de la infraestructura necesaria para establecer una conexión servidor-cliente compatible con algoritmos postcuánticos. Para ello, se utilizaron herramientas de LibOQS y se documentaron todas las etapas, facilitando así la replicación del proceso y su futura aplicación práctica.

Los resultados muestran que los algoritmos postcuánticos superan a esquemas tradicionales como RSA, al ofrecer una mayor resistencia frente a amenazas cuánticas y una mayor eficiencia. Por ello, se recomienda a las organizaciones adoptar de forma proactiva soluciones criptográficas híbridas como paso estratégico hacia una seguridad resiliente al entorno cuántico.

Palabras claves

Criptografía postcuántica, Algoritmos Quantum Safe, Criptografía híbrida, Transport Layer Security (TLS), Seguridad de la información, Computación cuántica.

Abstract

This research project aims to analyse the impact of implementing Quantum Safe encryption mechanisms in a business process, evaluating their feasibility, technical implications, and effects on security.

To carry out this study, post-quantum encryption algorithms standardized by the National Institute of Standards and Technology (NIST) have been implemented within the Transport Layer Security (TLS) protocol, which enables encrypted connections between client and server.

The study includes the deployment of the necessary infrastructure to establish such client-server connections compatible with post-quantum algorithms. For this purpose, tools provided by LibOQS have been used, and each step has been explained in detail in order to facilitate replication of the process and serve as a preliminary phase for practical implementation.

The results show that post-quantum algorithms outperform traditional schemes like RSA, offering both stronger resistance to quantum threats and greater efficiency. Organizations are encouraged to proactively adopt hybrid cryptographic solutions as a strategic step toward quantum-resilient security.

Keywords

Post-Quantum Cryptography, Quantum-Safe Algorithms, Hybrid Cryptography, Transport Layer Security (TLS), Information Security, Quantum Computing.

Índice de Contenidos

1. Introducción	11
2. Estado del Arte	13
2.1. Criptografía Moderna	13
2.1.1. Diffie-Hellman.....	14
2.1.2. Rivest, Shamir y Adleman.....	15
2.2. Computación Cuántica	17
2.3. Amenaza Cuántica.....	19
2.4 Criptografía post-cuántica (PQC).....	19
2.4.1. Lattices	21
2.4.2. Problema del ruido en muestras lineales	26
2.4.3. Kyber.....	29
2.5 Entidades de Seguridad postcuántica	31
2.5.1. NIST.....	31
2.5.2. Open Quantum Safe Project	31
2.6. LibOQS	32
2.7. Tecnologías	34
2.7.1. Protocolo TLS.....	34
2.7.1.OpenSSL	36
2.8. Trabajos similares.....	36
2.9. Conclusiones del Estado del Arte	40
3. Objetivos.....	41
3.1. Objetivo general	41
3.2. Lista de objetivos específicos.....	41
3.3. Métodos de Validación	41
4. Metodología y Plan de Trabajo	43
4.1. Materiales y recursos	46
4.2. Método	46
4.2.1. Infraestructura basada en Linux	46

4.2.2. Adaptación a Contenedores de Podman	47
4.2.3. Simulación conexiones TLS Servidor cliente	48
4.2.4. Validación de las conexiones	49
4.2.5. Exportación de rendimientos.....	53
4.2.5. Clasificación y representación de resultados.....	53
4.3. Plan de Trabajo.....	56
4.3.1. PT 1 Investigación previa	56
4.3.2. PT 2 Implementación Arquitectura basada en Linux.....	57
4.3.3. PT 3 Conexiones de prueba y recogida de rendimientos	57
4.3.4. PT 4 Procesamiento de los datos	58
4.3.5. PT 5 Representación de resultados y conclusiones.....	58
4.4. Condicionantes y Limitaciones.....	58
5. Resultados.....	60
5.1. Algoritmo de intercambio de clave Diffie-Hellman (DH)	61
5.1.1. Finite Field Diffie-Hellman (FFDH)	61
5.1.2. Elliptic Curve Diffie-Hellman (ECDH)	62
5.1.3. Comparativa.....	65
.....	65
5.2. Algoritmos Clásicos de Encapsulación de Clave (KEM)	66
5.1.1. Rivest Shamir Adleman (RSA)	66
5.1.2. Elliptic Curves (EC-KEM).....	68
5.3. Algoritmos Postcuánticos e Híbridos de Encapsulación de claves.....	70
5.1.3. Multivariant Lattice KEM / Kyber	70
5.1.4. Frodo	75
5.1.5. Bit Flipping Key Encapsulation (BIKE) y Hamming Quasi-Cyclic (HQC)	76
5.1.6. Comparativa	78
6. Implicaciones Éticas e Impacto Social.....	87
6.1. Valor del proyecto	88
6.2. Alcance	88
6.3. Implicaciones desde el punto social	89
6.4. Implicaciones desde el punto de vista económico	90
6.5. Implicaciones medioambientales	90
6.6. Marco legal.....	91

6.7. Riesgos del Proyecto.....	92
6.8. Conclusiones	95
7. Conclusiones y Trabajos Futuros.....	96
8. Otros Méritos del Proyecto.....	98
9. Bibliografía	99
Anexo A: Licencias de Software.....	107
Anexo B: Despliegue infraestructura	109
Anexo C: Tablas de Rendimientos	112
Anexo D: Código para el procesamiento de Datos y Visualización..	113
Anexo E: Tablas de Rendimiento Procesadas	114
Anexo F: Podman	115

Índice de Tablas

<i>Tabla 1. Niveles de seguridad de los algoritmos modernos [20]</i>	19
<i>Tabla 2. Algoritmos postcuánticos de firma digital ofrecidos por LibOQS en la version 0.12.0. Elaboración propia.....</i>	32
<i>Tabla 3. Algoritmos postcuánticos KEM ofrecidos por LibOQS en la version 0.12.0. Elaboración propia.....</i>	33
<i>Tabla 4. Curvas elípticas de algoritmos híbridos ofrecidos por LibOQS. Elaboración propia.</i>	33
<i>Tabla 5. Fases del protocolo TLS. Elaboración Propia.....</i>	35
<i>Tabla 6. Comparativa TLS Tradicional vs TLS Quantum Safe. Elaboración propia.</i>	36
<i>Tabla 7. Métricas recogidas en conexiones TLS Quantum Safe. Elaboración propia.....</i>	52
<i>Tabla 8. Métricas disponibles por categoría del algoritmo. Elaboración propia.....</i>	53
<i>Tabla 9. Desglose de dos algoritmos ejecutados Elaboración propia.....</i>	55
<i>Tabla 10. Algoritmo post procesamiento. Elaboración propia.</i>	55
<i>Tabla 11. Características de los algoritmos cuánticos y protocolos criptográficos a los que amenazan [64]......</i>	56
<i>Tabla 12. Funciones criptográficas en TLS. Elaboración propia.....</i>	56
<i>Tabla 13. Métricas recogidas en conexiones TLS Quantum Safe. Elaboración propia.....</i>	58
<i>Tabla 14. Requerimientos de espacio computacional del proyecto. Elaboración propia.....</i>	59
<i>Tabla 15. Elementos del entorno experimental.</i>	60
<i>Tabla 16. Niveles de seguridad NIST [65]</i>	60
<i>Tabla 17. Bits de seguridad Finite Fields Diffie Helman [67][68].</i>	61
<i>Tabla 18. Rendimiento Finite Fields Diffie Helman. Elaboración propia.</i>	61
<i>Tabla 19. Niveles de seguridad de curvas elípticas NIST [66]</i>	63
<i>Tabla 20. Niveles de seguridad de curvas elípticas del proyecto Curve25519 [66]</i>	63
<i>Tabla 21. Niveles de seguridad de curvas elípticas Brainpool [67].</i>	63
<i>Tabla 22. Rendimientos ECDH. Elaboración propia.</i>	64
<i>Tabla 23Bits de seguridad en RSA [75].</i>	67
<i>Tabla 24. Rendimientos RSA. Elaboración propia.....</i>	67
<i>Tabla 26. Bits de seguridad de curvas elípticas KEM [67].</i>	68
<i>Tabla 27. Rendimiento de EC-KEM. Elaboración propia.....</i>	68
<i>Tabla 28. Diferencia de tiempo entre ML-KEM y Kyber. Elaboración propia.</i>	71
<i>Tabla 29. Rendimientos de ML-KEM en AMD Ryzen 7 [82].</i>	72
<i>Tabla 30. Rendimientos de ML-KEM en procesador Intel i5-11367G. Elaboración propia.....</i>	72
<i>Tabla 31. Bits de Seguridad de ML-KEM [65].</i>	73

<i>Tabla 32. Aplicaciones de esquemas híbridos en distintos entornos empresariales. Elaboración propia.</i>	74
<i>Tabla 33. Bits de seguridad de BIKE [87].</i>	77
<i>Tabla 34. Bits de seguridad de HQC [89].</i>	77
<i>Tabla 35. Tabla comparativa de los algoritmos KEM. Elaboración propia.</i>	79

Índice de Figuras

<i>Figura 1 Usuario conectándose a un servidor utilizando criptografía postcuántica. Elaboración propia.</i>	12
<i>Figura 2. Esquema Diffie- Hellman. Elaboración propia.</i>	14
<i>Figura 3. IBM Quantum Roadmap [16].</i>	18
<i>Figura 4. Esquema de criptografía moderna vs postcuántica. Elaboración propia.</i>	20
<i>Figura 5. Adopción de criptografía postcuántica en Cloudflare España [79].</i>	20
<i>Figura 6. Lattices 2D definidas por vectores base. Elaboración propia.</i>	22
<i>Figura 7. Lattice normal cuadrada. Elaboración propia.</i>	22
<i>Figura 8. Lattice con vectores base (0,1) y (2,0). Elaboración propia.</i>	23
<i>Figura 9. Shortest Vector Problem y Closest Vector Problem.</i>	25
<i>Figura 10. Goldreich-Goldwasser-Halevi Cryptosystem. Elaboración propia.</i>	26
<i>Figura 11. Formulación LWE sin el error añadido. Elaboración propia.</i>	27
<i>Figura 12. Formulación LWE. Elaboración propia.</i>	27
<i>Figura 13. Representación geométrica del problema Learning With Errors (LWE): soluciones exactas desplazadas por ruido en el espacio modular. Elaboración propia.</i>	27
<i>Figura 14. Representación del grupo cíclico \mathbb{Z}_p en aritmética modular.</i>	29
<i>Figura 15. Representación matricial de Kyber. Elaboración propia.</i>	29
<i>Figura 16. Representación visual de fases de Kyber. [36]</i>	30
<i>Figura 17. Implementación de HTTP junto a TLS para crear una comunicación segura HTTPS. Elaboración propia.</i>	34
<i>Figura 18. Comparación de tiempo de ejecución de protocolos OQS [46]</i>	37
<i>Figura 19. Diagrama de Gantt del proy.</i>	45
<i>Figura 20. Contenedores de Docker modificados para soportar algoritmos de cifrado post-cuántico mediante la biblioteca LibOQS [56][57]</i>	47
<i>Figura 21. Repositorios de contenedores curl y nginx en Github [58].</i>	47
<i>Figura 22. Interacción de la librería LibOQS con los contenedores nginx y curl [59].</i>	48
<i>Figura 23. Conexión entre un cliente curl y un servidor nginx [63].</i>	49
<i>Figura 24 Ciclo del análisis de datos. Elaboración propia.</i>	54
<i>Figura 25. Diseño de la infraestructura. Elaboración propia.</i>	57
<i>Figura 26. Tiempos de ejecución de FFDH para distintos tamaños de clave. Elaboración propia</i>	62
<i>Figura 27. Tiempo de operación ECDH. Elaboración propia.</i>	64
<i>Figura 28. Comparativa tiempos de operación de Algoritmos de Intercambio de clave. Elaboración propia.</i>	65
<i>Figura 29 Key Encapsulation Mechanism [68].</i>	66

<i>Figura 30. Rendimiento medido en operaciones por segundo de RSA. Elaboración propia.....</i>	67
<i>Figura 31. Tiempo total de ejecución en algoritmos EC-KEM. Elaboración propia.....</i>	69
<i>Figura 32. Mapa de calor diferencia en operaciones por segundo para 2 procesadores distintos.</i>	72
<i>Figura 33. Mapa de calor de Rendimientos de ML-KEM en sus combinaciones híbridas.</i>	74
<i>Figura 34. Tiempos totales de ejecución de las variantes FrodoKEM. Elaboración propia.</i>	75
<i>Figura 35. Mapa de calor de tiempo total de ejecución para FrodoKEM. Elaboración propia.</i>	76
<i>Figura 36. Comparativa de rendimientos de HQC y BIKE.</i>	77
<i>Figura 37. Comparativa de rendimientos por Familia de Algoritmo y nivel de seguridad. Elaboración propia.....</i>	81
<i>Figura 38. Mapa de calor de rendimiento según la curva elíptica aplicada en los algoritmos KEM. Elaboración propia.....</i>	84
<i>Figura 39. Promedio de velocidad de operaciones para Algoritmos KEM. Elaboración propia.</i>	85
<i>Figura 40. Promedio de encapsulación y decapsulación por algoritmo KEM.....</i>	86
<i>Figura 41. Imagen de Deep Blue jugando contra Garry Kasparov [91].</i>	87
<i>Figura 42. Métricas de consumo energético por partición en servidores IBM LinuxONE [93].</i>	91

Índice de Outputs

<i>Output 1 fragmento de log de conexión TLS con Kyber 768.....</i>	50
<i>Output 2 listado de algoritmos de encapsulación de clave.....</i>	51
<i>Output 3 Conexiones TLS con algoritmo KEM kyber512 y algoritmo de firma dilithium3</i>	52
<i>Output 4 Conexiones TLS con algoritmo KEM kyber1024 y algoritmo de firma falcon512.....</i>	52
<i>Output 5 Log de ejecución.....</i>	55
<i>Output 6 Espacio de almacenamiento de los contenedores (en MegaBytes)</i>	59

Lista de Acrónimos

Acrónimo	Significado
NIST	National Institute of Standards and Technology
TLS	Transport Layer Security
PQC	Post Quantum Criptography
OQS	Open Quantum Safe
LibOQS	Library for Open Quantum Safe
DH	Diffie Hellman
ECC	Elliptic Curves Criptography
FFDH	Finite Fields Diffie Hellman
DSA	Digital Signature Algorithm
RSA	Rivest, Shamir y Adelman
LWE	Learning With Errors
SVP	Shortest Vector Problem
CVP	Closest Vector Problem
HQC	Hamming Quasi-Cyclic
BIKE	Bit Flipping Key Encapsulation
ML-KEM	Module Lattice–Key Encapsulation Mechanism

1. INTRODUCCIÓN

Cada vez son mayores los riesgos que presentan las tecnologías emergentes en la seguridad de los datos. Desde ataques cibernéticos hasta vulnerabilidades en dispositivos inteligentes, la necesidad de implementar estrategias sólidas de encriptación se ha vuelto esencial para proteger la información confidencial y garantizar la privacidad de los usuarios. En Estados Unidos los delitos cibernéticos y fraudes generaron pérdidas por 16.800 millones USD, un incremento del 33 % respecto a 2023 [1]. En España, el Instituto Nacional de Ciberseguridad (INCIBE) presentó un balance de ciberseguridad en 2024 con más de 97.000 incidentes gestionados reportando que los incidentes crecieron un 16 % en comparación con 2023, afectando al 45 % de las empresas españolas [2].

En noviembre de 2024, inicié mis prácticas académicas en el departamento de ciberseguridad de IBM en Madrid. Durante esta experiencia, tuve la oportunidad de observar de primera mano la importancia de la ciberseguridad y la urgente necesidad de adaptar los métodos tradicionales de encriptación a las nuevas tecnologías.

Uno de los principales riesgos asociados al desarrollo de la computación cuántica es su capacidad para comprometer los sistemas de cifrado asimétrico actualmente utilizados. En particular, el algoritmo de Shor permite la factorización eficiente de números primos grandes, lo que representa una amenaza directa para algoritmos como RSA y ECC [3].

Para hacer frente a este desafío, el Instituto Nacional de Estándares y Tecnología (NIST) ha seleccionado un conjunto de algoritmos resistentes a ataques cuánticos, conocidos como esquemas de criptografía postcuántica. Entre ellos destacan CRYSTALS-Kyber para el intercambio de claves, y CRYSTALS-Dilithium, Falcon y SPHINCS+ para la generación de firmas digitales. Estos algoritmos se basan en problemas matemáticos considerados difíciles incluso para los ordenadores cuánticos, como las redes euclidianas (lattices), los códigos corregidores de errores y las funciones hash. Se prevé que estos sistemas reemplacen gradualmente a las soluciones criptográficas tradicionales en los próximos años [4].

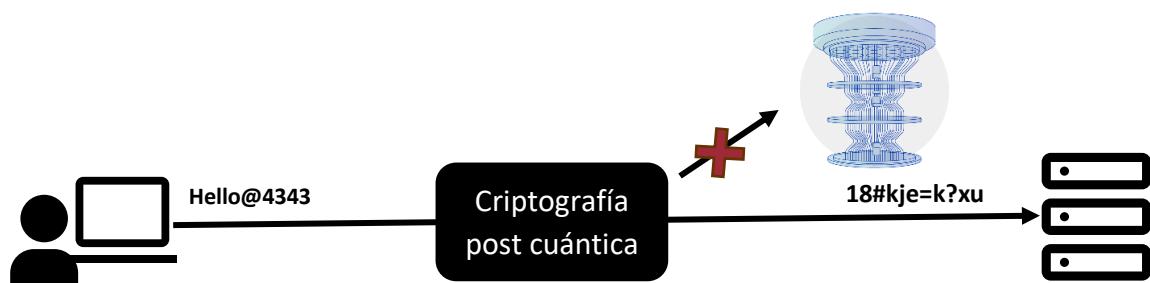


Figura 1 Usuario conectándose a un servidor utilizando criptografía postcuántica. Elaboración propia.

Este proyecto se centra específicamente en analizar dicha transición en una de las aplicaciones más críticas dentro del entorno empresarial: el protocolo Transport Layer Security (TLS) que protege la comunicación en Internet [5].

En este contexto, la biblioteca LibOQS (Library for Open Quantum Safe) proporciona implementaciones de referencia de múltiples algoritmos postcuánticos, permitiendo su integración en protocolos de seguridad como TLS [6]. Estas herramientas han sido desplegadas para la evaluación práctica del impacto de la transición hacia esquemas de seguridad Quantum Safe.

Mediante una serie de experimentos controlados, se han recopilado y procesado datos relacionados con el rendimiento, la seguridad y el esfuerzo de implementación. Posteriormente, estos datos se han visualizado y analizado para extraer conclusiones precisas. Este enfoque contribuye a validar la viabilidad técnica de los algoritmos postcuánticos y permite generar recomendaciones de negocio específicas que aportan valor a la investigación.

2. ESTADO DEL ARTE

Dado que este Trabajo Final de Grado (TFG) se enmarca en el ámbito de la investigación matemática aplicada a la criptografía postcuántica, la presente sección de Estado del Arte adquiere un papel fundamental. Su objetivo es doble: por un lado, contextualizar la problemática de los mecanismos criptográficos actuales; por otro, analizar los progresos en el campo de la criptografía postcuántica, prestando especial atención al impacto que supone su adopción en entornos reales, con un estudio de caso centrado en la transición del protocolo TLS hacia versiones resistentes a ataques cuánticos.

Para ello, se analizan las bases de la criptografía tradicional y sus vulnerabilidades ante la amenaza que representan las computadoras cuánticas. A continuación, se exponen los fundamentos y principios de la criptografía postcuántica, que buscan solventar estas problemáticas.

Se presta especial atención a los estándares y procesos liderados por el NIST para la estandarización de algoritmos postcuánticos, así como a las iniciativas de Open Quantum Safe, que facilitan la integración práctica de estos esquemas. Finalmente, se estudia la evolución del protocolo TLS en este contexto, valorando los retos técnicos y de implementación asociados.

2.1. CRIPTOGRAFÍA MODERNA

La criptografía es el estudio y práctica de técnicas para proteger la información mediante su transformación (cifrado), de modo que solo personas autorizadas puedan acceder a ella. Es fundamental en la seguridad digital ya que garantiza la confidencialidad, integridad y autenticación [7].

En la criptografía moderna se emplean principalmente dos enfoques: criptografía simétrica y asimétrica, que a menudo se combinan en protocolos como TLS [8][9].

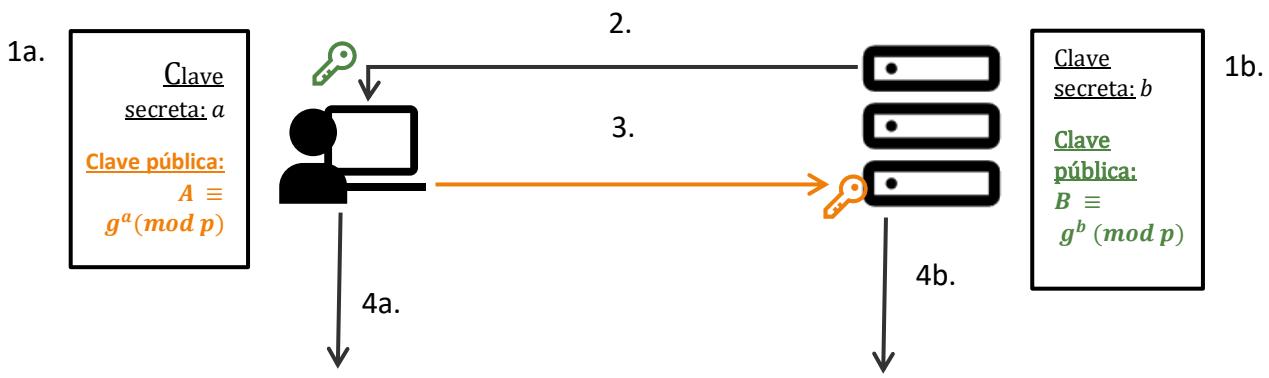
- Criptografía simétrica: utiliza la misma clave para cifrar y descifrar; rápida y eficiente [8].
- Criptografía asimétrica: utiliza un par de claves pública/privada; facilita el intercambio seguro de claves [8][9].

2.1.1. Diffie-Hellman

A continuación, se detalla el problema matemático Diffie-Hellman en el que se basan los esquemas de clave pública actuales utilizados en TLS [10].

Se escoge un número primo p y un elemento g en F_p^\times , es decir, g es un número entero no múltiplo de p . Donde F_p es el cuerpo con p elementos y F_p^\times es el conjunto de elementos invertibles en este cuerpo. En aplicaciones reales es importante que el número primo p sea grande y que g tenga un orden grande dentro del grupo F_p^\times .

1. a. El cliente genera un par de claves:
 - Clave secreta: a
 - Clave pública: $A \equiv g^a \pmod{p}$
1. b. El servidor genera un par de claves (clave secreta y pública)
 - Clave secreta: b
 - Clave pública: $B \equiv g^b \pmod{p}$
2. El servidor envía su clave pública efímera (B) al cliente.
3. El cliente responde con su clave pública efímera (A).
4. a. El cliente calcula la clave común como $A' \equiv B^a \pmod{p}$
4. b. El servidor calcula la clave común como $B' \equiv A^b \pmod{p}$



Clave compartida: $A' \equiv B^a \pmod{p}$

$B' \equiv A^b \pmod{p}$

Figura 2. Esquema Diffie- Hellman. Elaboración propia.

Donde:

- Claves públicas: A, B, g, p
- Claves secretas: $1 \leq a \leq p-2; 1 \leq b \leq p-2$
- Clave compartida: A'

Como resultado se obtiene una clave secreta común con la que cifrar mensajes, firmar mensajes o intercambiar claves simétricas.

Demostración $A' \equiv B' \pmod{p}$

$$\begin{aligned} A^b &\equiv B^a \pmod{p} \\ (g^a)^b &\equiv (g^b)^a \pmod{p} \\ g^{ab} &\equiv (g^b)^a \pmod{p} \end{aligned}$$

Esta clave compartida se usa para derivar las claves de cifrado simétrico que protegerán la sesión. Si un intermediario quisiera obtener la clave común, tendría que resolver:

$$a = \log_g A$$

o

$$b = \log_g B$$

Operación computacionalmente costosa para ordenadores clásicos

Problema del Logaritmo Discreto

Diffie-Hellman (DH) implica la generación de claves públicas de la forma:

$$Y = g^x \pmod{p}$$

donde la clave privada x es el logaritmo discreto.

Un atacante que pueda resolver una instancia del problema del logaritmo discreto (DLP, por sus siglas en inglés) puede exponer la clave privada de una de las dos partes y, al combinarla con la clave pública de la segunda parte, obtener acceso al secreto compartido.

En los sistemas informáticos clásicos, se cree que el DLP no tiene una solución en tiempo polinómico [11].

2.1.2. Rivest, Shamir y Adleman

Es un algoritmo de cifrado asimétrico que permite la transmisión segura de mensajes sin necesidad de compartir una clave secreta previamente [12].

1. El servidor (o la entidad que genera las claves) elige dos números primos grandes, p, q .
2. Calcula:
 - o $n = p \cdot q$
 - o $\varphi(n) = (p - 1) \cdot (q - 1)$

donde φ es la función totiente de Euler.

3. Elige un número e tal que:

- $1 < e < \varphi(n)$
- $\gcd(e, \varphi(n)) = 1$, es decir, que $(e, \varphi(n))$ sean coprimos.

4. Calcula el inverso multiplicativo de $e \pmod{\varphi(n)}$

Es decir, un d tal que: $d \cdot e \equiv 1 \pmod{\varphi(n)}$

3. Claves resultantes:

- Clave pública: (e, n)
- Clave privada: (e, d)

3. Cifrado y Descifrado

- Cifrado (con la clave pública):

Dado un mensaje m (donde $0 < m < n$, el emisor calcula:

$$c \equiv m^e \pmod{n}$$

- Descifrado (con la clave privada):

El receptor calcula:

$$m \equiv c^d \pmod{n}$$

Demostración de que el descifrado recupera el mensaje original

Queremos probar que $m \equiv c^d \pmod{n}$:

A partir de $c \equiv m^e \pmod{n}$, entonces:

$$m \equiv m^{ed} \pmod{n}$$

Por el Teorema de Euler y la forma en que se construyen e y d , se cumple que:

$$ed \equiv 1 \pmod{\varphi(n)} \Rightarrow ed = 1 + k\varphi(n)$$

Entonces:

$$m \equiv m^{1+k\varphi(n)} \pmod{n} \rightarrow m \equiv m \cdot m^{k\varphi(n)} \pmod{n}$$

Dividiendo por m a ambos lados: $1 \equiv m^{k\varphi(n)} \pmod{n}$

Y desarrollando la igualdad: $1 \equiv \underbrace{m^{\phi(y)} \cdot m^{\phi(n)} \cdots m^{\phi(n)}}_{k \text{ veces}} \pmod{n}$

Como m es coprimo con n , y por el teorema de Euler: $m^{\phi(n)} \equiv 1 \pmod{n}$

Por tanto, el descifrado recupera correctamente el mensaje original.

Factorización como producto de primos

El algoritmo RSA se basa en un problema diferente: la factorización de números enteros grandes. En concreto, la seguridad de RSA depende de la dificultad de obtener los factores primos p y q a partir del producto $n = p \cdot q$. Conocer estos factores permite calcular la clave privada a partir de la clave pública.

Al igual que con el DLP, en la actualidad no se conoce ningún algoritmo clásico eficiente para factorizar números grandes [11].

2.2. COMPUTACIÓN CUÁNTICA

La computación cuántica se fundamenta en los principios de la mecánica cuántica, tales como la superposición, el entrelazamiento cuántico y la teleportación cuántica. A diferencia de los ordenadores clásicos, que procesan información utilizando bits (0 o 1), los ordenadores cuánticos emplean qubits.

Los qubits pueden representar múltiples estados simultáneamente gracias a la superposición. Esta capacidad permite una paralelización masiva de operaciones, abriendo nuevas posibilidades en el tratamiento de problemas computacionales complejos [13].

“Nature isn’t classical... and if you want to make a simulation of nature, you’d better make it quantum mechanical.”

– Richard Feynman

Este tipo de computación ofrece ventajas significativas en áreas como la optimización, la simulación de materiales y reacciones químicas, y el aprendizaje automático, donde los métodos clásicos resultan ineficientes o directamente inviables [14].

Diversos organismos internacionales coinciden en que la computación cuántica será una de las tecnologías más transformadoras de la próxima década. El World Economic Forum (WEF) señala que los ordenadores cuánticos tienen el potencial de revolucionar sectores estratégicos como la salud, la industria, la logística, la energía y las finanzas, al permitir resolver problemas que son intratables para la computación clásica actual. En su informe “Quantum Computing in the Next Decade”, el WEF destaca aplicaciones como el descubrimiento acelerado de nuevos fármacos, la optimización de cadenas de suministro y el modelado de sistemas complejos en energía y materiales [17].

IBM ha desarrollado una infraestructura avanzada que permite el acceso remoto a sus ordenadores cuánticos a través de la plataforma en la nube IBM Quantum. Esta plataforma proporciona acceso a una variedad de procesadores cuánticos, desde sistemas de 5 qubits hasta procesadores más avanzados, como el IBM Quantum Eagle, con 127 qubits, lanzado en 2021 [15]. Además, IBM continúa ampliando su hoja de ruta tecnológica con la presentación del IBM Quantum System Two y una visión de sistemas de mayor capacidad, como parte de su estrategia hacia la era de la “utilidad cuántica” [16].

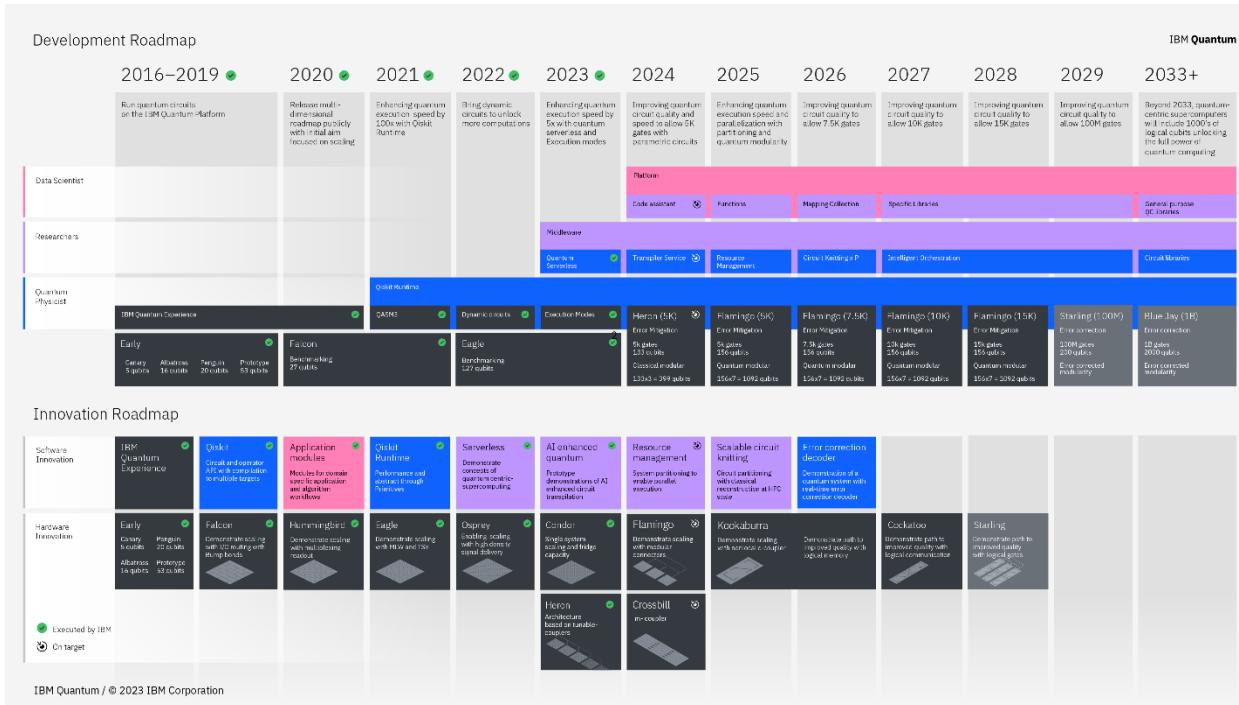


Figura 3. IBM Quantum Roadmap [16].

El algoritmo de Shor es un algoritmo cuántico desarrollado por Peter Shor en 1994, diseñado para factorizar números enteros grandes de manera eficiente, algo que resulta extremadamente difícil para los ordenadores clásicos [11].

El problema de factorización se puede convertir en un problema de encontrar el período en tiempo polinomial. Por tanto, el algoritmo de Shor para encontrar períodos también puede factorizar enteros eficientemente.

Dada una base a , elegida aleatoriamente tal que $1 < a < N$ y $\gcd(a, N) = 1$, se define la siguiente función:

$$f(x) = a^x \pmod{N}$$

Esta función es periódica, y el objetivo principal del algoritmo cuántico es encontrar su **período r** , es decir, el menor entero positivo tal que:

$$a^r \equiv 1 \pmod{N}$$

Una vez encontrado r , si r es par y $a^{\frac{r}{2}} \equiv -1 \pmod{N}$, se pueden obtener factores de N mediante:

$$\gcd(a^{\frac{r}{2}} \pm 1, N)$$

El enfoque principal del algoritmo de Shor es transformar el problema de la factorización en uno de encontrar el período de una función matemática específica. Shor usa la capacidad de los computadores cuánticos para encontrar ese período de forma rápida y eficiente [18].

Aunque queda fuera del alcance del proyecto, existen otros algoritmos cuánticos como el algoritmo de Grover que también plantean riesgos, en este caso para la criptografía simétrica. Grover permite reducir la complejidad de un ataque de fuerza bruta desde $O(2^n)$ a $O(2^{\frac{n}{2}})$, lo que obligaría a aumentar el tamaño de las claves simétricas (por ejemplo, en AES) para mantener un nivel de seguridad equivalente al actual [20].

2.3. AMENAZA CUÁNTICA

Si bien se prevé que la computación cuántica tendrá un impacto transformador en sectores como la salud, la industria, el comercio y la logística, también representa una amenaza significativa para los sistemas criptográficos actuales.

En particular, este proyecto se centra en la migración de los esquemas de criptografía de clave pública, como RSA y Diffie-Hellman. Estos esquemas se basan en la dificultad computacional de ciertos problemas matemáticos —la factorización de enteros y el logaritmo discreto, respectivamente— que resultan inabordables para los ordenadores clásicos con recursos razonables.

Sin embargo, se prevé que los ordenadores cuánticos pueden resolver estos problemas de forma eficiente utilizando el algoritmo de Shor, lo que permitiría romper estos sistemas criptográficos en tiempo polinómico [18].

Shor's algorithm can factor primes with $O(n^2)$ where n is the number of bits [19].

Esto supondría una pérdida completa de la confidencialidad, la autenticidad y la integridad de la información en muchas infraestructuras de seguridad digital. Algoritmos como DH, DSA y RSA, que actualmente se consideran seguros frente a atacantes clásicos, quedarían obsoletos en un escenario cuántico.

Cryptographic algorithm	After quantum computing
AES-256	Secure but weakened
SHA-256	Secure but weakened
RSA	No longer secure
ECDSA	No longer secure
DSA	No longer secure

Tabla 1. Niveles de seguridad de los algoritmos modernos [20]

2.4 CRIPTOGRAFÍA POST-CUÁNTICA (PQC)

Ante las amenazas que representa la computación cuántica para los sistemas criptográficos actuales, ha surgido un nuevo campo de estudio conocido como criptografía postcuántica (Post-Quantum Cryptography, PQC). Esta disciplina se centra en el desarrollo de algoritmos criptográficos que permanezcan seguros incluso frente a adversarios que dispongan de ordenadores cuánticos a gran escala [21].

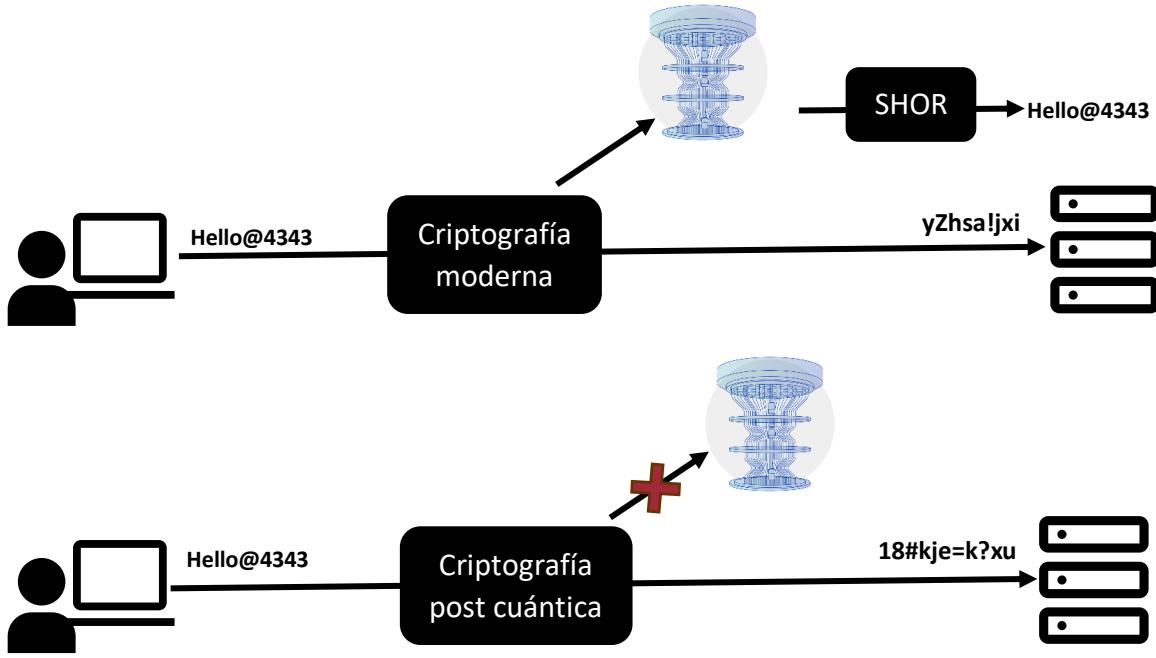


Figura 4. Esquema de criptografía moderna vs postcuántica. Elaboración propia.

"La criptografía post-cuántica busca identificar problemas matemáticos frente a los cuales los algoritmos cuánticos conocidos, como el de Shor o el de Grover, ofrecen poca o ninguna ventaja computacional. El principal reto de la PQC consiste en lograr soluciones que sean no solo seguras, sino también prácticas y eficientes para su uso generalizado en entornos reales" [22].

Cloudflare es un servicio que mejora el rendimiento y la seguridad web. En [su portal Radar](#), ofrece visibilidad en tiempo real sobre el uso de tecnologías como la criptografía postcuántica. Según sus datos, el 32,9 % del tráfico HTTPS en su red ya la emplea, aunque aún queda un largo camino para su adopción total.

Post-quantum encryption adoption in Spain

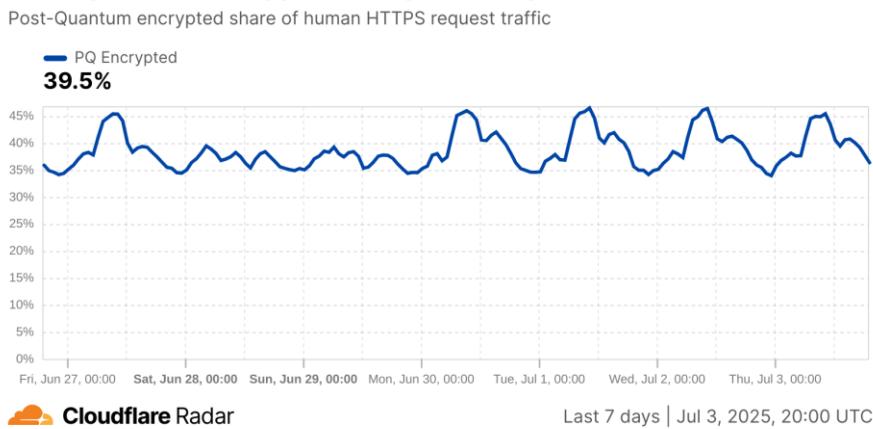


Figura 5. Adopción de criptografía postcuántica en Cloudflare España [79].

Hasta la fecha, se han identificado cinco grandes familias de algoritmos postcuánticos:

- Criptografía basada en retículas (Lattice-based): se apoya en problemas como LWE (Learning With Errors) y se utiliza para cifrado, intercambio de claves y firmas digitales.

- Criptografía basada en códigos (Code-based): como el sistema McEliece, se basa en problemas de corrección de errores.
- Criptografía basada en funciones hash (Hash-based)
- Criptografía multivariable (Multivariate-based): usa sistemas de ecuaciones polinómicas con múltiples variables.
- Criptografía basada en isogenias de curvas elípticas supersingulares (SIKE): aprovecha la dificultad de calcular isogenias entre curvas elípticas supersingulares.

[21]

Este trabajo se enfoca particularmente en la criptografía basada en retículas. En especial en el problema del Ruido de Muestras Lineales (LWE, *Learning With Errors*). Estos problemas matemáticos constituyen la base de algoritmos estandarizados como Kyber.

2.4.1. Lattices

A continuación, se estudia la criptografía basada en retículas, a las que se hará referencia como *lattices* a partir de este punto.

Una lattice L en \mathbb{R}^n es el conjunto de todas las combinaciones lineales enteras de m vectores linealmente independientes. $B = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_m\}$ en \mathbb{R}^n donde $m \leq n$. El set B se llama una base de L , y escribimos $L = L(B)$. La dimensión de L es n y el rango de L es m .

- En adelante, asumiremos que los vectores base $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_m$ están en \mathbb{Z}^n
- Entonces $L = \{x_1 \cdot \vec{v}_1 + x_2 \cdot \vec{v}_2 + \dots + x_n \cdot \vec{v}_m \mid x_1, x_2, \dots, x_n \in \mathbb{Z}\} \subseteq \mathbb{Z}^n$. L es una lattice entera.
- Sea B la matriz $n \times m$ cuyas columnas son los vectores base $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_m$:

$$B = \begin{pmatrix} | & | & & | \\ \vec{v}_1 & \vec{v}_2 & \dots & \vec{v}_m \\ | & | & & | \end{pmatrix} \rightarrow L = \{B \cdot x \mid x \in \mathbb{Z}^m\}$$

Redes de rango completo (Full-rank lattices)

Una red de rango completo L en \mathbb{R}^n es una red en \mathbb{R}^n de rango n .

Definición [23]. Sean L y L' redes en \mathbb{R}^n . Entonces, L' es una **sublattice** de L si $L' \subseteq L$.

- Se asume que todas las redes y subredes serán de **rango completo** (e **integer**, es decir, con coeficientes enteros).
- Una base $B = \{\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n\}$ para una lattice de rango completo en \mathbb{R}^n es también una base del espacio vectorial \mathbb{R}^n .

Ejemplo en un entorno simplificado (2-dimensiones):

Las lattice 2D siguientes se pueden generar a partir de 2 vectores base.

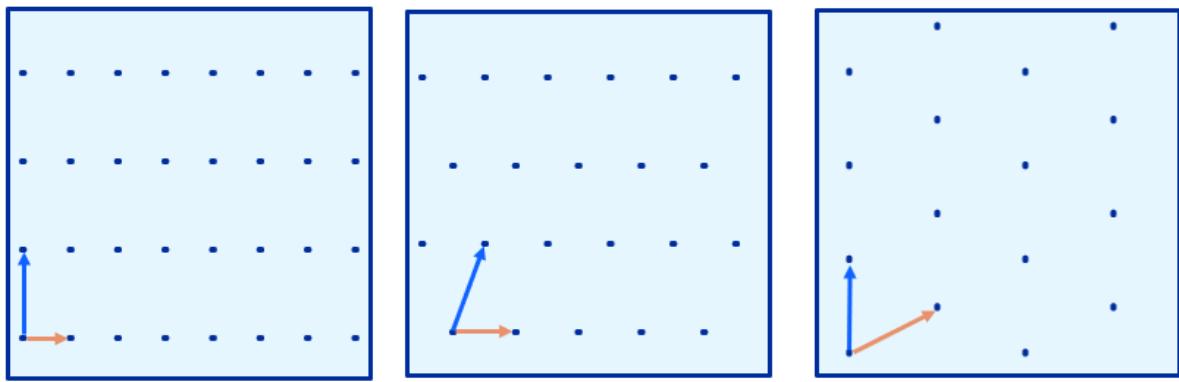


Figura 6. Lattices 2D definidas por vectores base. Elaboración propia.

- La base de la lattice, se utiliza para generar todos los puntos posibles.
- Un lattice está formada por todos los puntos a los que se puede llegar a partir de sus vectores base.
- A un mismo punto se puede llegar por más de un camino.
- 2 bases diferentes pueden generar la misma lattice.

Los siguientes ejemplos se desarrollan en \mathbb{Z}^2

Lattice Normal cuadrada

$$B = \{(1,0), (0,1)\}$$

$$L_1 = L(B_1) = \{B_1 \cdot x | x \in \mathbb{Z}^2\}$$

Entonces $L_1 = \mathbb{Z}^2$

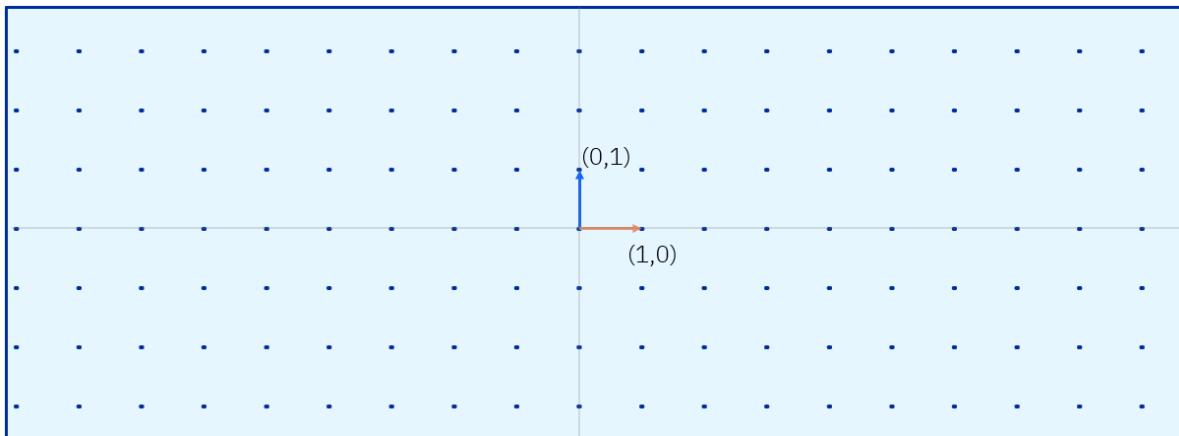


Figura 7. Lattice normal cuadrada. Elaboración propia.

Cualquier punto del lattice se obtiene como:

$$\begin{aligned}\vec{p} &= x_1 \cdot \vec{v}_1 + x_2 \cdot \vec{v}_2 \\ \vec{p} &= x_1 \cdot (1,0) + x_2 \cdot (0,1)\end{aligned}$$

Esto genera todos los puntos con coordenadas (x_1, x_2) que forman la clásica cuadrícula del plano cartesiano. $L = \{x_1 \cdot (1,0) + x_2 \cdot (0,1) | x \in \mathbb{Z}^2\}$

Lattice con B_2 = { (2, 0), (0, 1)}

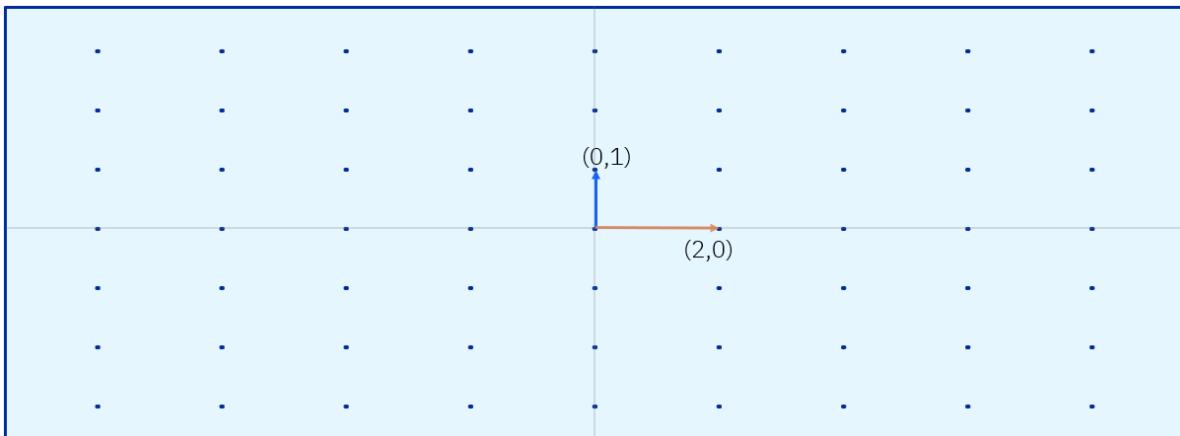


Figura 8. Lattice con vectores base $(0,1)$ y $(2,0)$. Elaboración propia.

Vectores base: $\vec{v}_1 = (2,0); \vec{v}_2 = (0,1)$.

$$L_2 = L(B_2) = \{B_2 \cdot x | x \in \mathbb{Z}^2\}, \text{ donde } B_2 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

Se responde a los siguientes problemas de lattices:

1. ¿Es L_2 una sublattice de L_1 ?

$$(2,0) = 2 \cdot (1,0) + 0 \cdot (0,1)$$

$$(0,1) = 0 \cdot (1,0) + 1 \cdot (0,1)$$

Sí, $L_2 \subseteq L_1$, ya que las coordenadas de la base de L_2 pueden representarse como una combinación lineal entera de las coordenadas de la base de L_1 .

2. ¿Es L_1 una sublattice de L_2 ?

$$(1,0) = \frac{1}{2} \cdot (2,0) + 0 \cdot (0,1) \notin L_2$$

$$(0,1) \in L_2$$

No, $L_1 \not\subseteq L_2$, ya que los coeficientes de la combinación lineal deben ser enteros para el contexto que nos ocupa de lattices enteras. $\frac{1}{2} \notin \mathbb{Z}$

3. ¿Son L_1 y L_2 la misma lattice?

Como $L_2 \subseteq L_1$ pero $L_1 \not\subseteq L_2$, se demuestra que: L_2 está contenida en L_1 pero no son la misma lattice.

$L_2 \neq L_1$ ya que B_1, B_2 no generan la misma lattice.

Lattice con $B_3 = \{(2,1), (7,3)\}$

Vectores base: $\vec{v}_1 = (2,1); \vec{v}_2 = (7,3)$.

$$L_3 = L(B_3) = \{B_3 \cdot x | x \in \mathbb{Z}^2\}, \text{ donde } B_3 = \begin{pmatrix} 2 & 1 \\ 7 & 3 \end{pmatrix}$$

Se responde a los siguientes problemas de lattices:

1. ¿Es L_1 una sublattice de L_3 ?

$$(1,0) = a \cdot (2,1) + b \cdot (7,3)$$

$$(0,1) = c \cdot (2,1) + d \cdot (7,3)$$

Solución: $a = -3; b = 1; c = 7; d = -2$

Sí, $L_1 \subseteq L_3$, ya que las coordenadas de la base de L_1 pueden representarse como una combinación lineal entera de las coordenadas de la base de L_3 . En este caso la solución del sistema únicamente tiene componentes enteros $a, b, c, d \in \mathbb{Z}$.

2. ¿Es L_3 una sublattice de L_1 ?

$$(2,1) = 2 \cdot (1,0) + 1 \cdot (0,1)$$

$$(7,3) = 7 \cdot (1,0) + 3 \cdot (0,1)$$

Sí, $L_3 \subseteq L_1$, ya que las coordenadas de la base de L_3 pueden representarse como una combinación lineal entera de las coordenadas de la base de L_1 .

3. ¿Son L_1 y L_3 la misma lattice?

$$L_3 = L(\{(2,1), (7,3)\})$$

$$L_1 = L(\{(1,0), (0,1)\})$$

Sí son la misma lattice, $L_1 = L_3$ ya que, los vectores base de B_1 se pueden calcular como una combinación lineal de los vectores base de B_2 y viceversa, con coeficientes enteros.

Conclusiones:

- L_1 y L_3 son la misma lattice, pero descritas sobre bases distintas.
- La base $B_1 = \{(1,0), (0,1)\}$ es ‘mejor’ que la $B_3 = \{(2,1), (7,3)\}$ ya que los vectores en la B_1 son de menor norma y ortogonales.

La norma de un vector o distancia euclídea se calcula como:

$$\|\vec{v}\| = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2}$$

Con $\vec{v} = (v_1, v_2, \dots, v_n) \in \mathbb{R}^n$

[23][24]

PROBLEMAS DE LATTICES:

1. Shortest Vector Problem (SVP): dada una lattice $L(B) \subseteq \mathbb{Z}^n$, el problema consiste en encontrar el vector no nulo más corto en L . Es decir, un $v \in L \setminus \{0\}$ tal que $\|v\|$ sea mínimo.
- Para la lattice normal cuadrada del ejemplo anterior, los vectores más cortos no nulos en la red son los vectores base y sus opuestos: $v_{sol} = \{\pm(1,0), \pm(0,1)\}$. Estos tienen norma 1, y no existe ningún otro vector no nulo en la red con menor longitud.
2. Closest Vector Problem (CVP): dada una lattice $L(B) \subseteq \mathbb{Z}^n$ y un vector objetivo $t \in \mathbb{R}^n$ que no necesariamente pertenece a la red: Encontrar un vector $v \in L(B)$ tal que $\|v - t\|$ sea mínimo. En otras palabras, se busca el vector de la red que esté más cerca del vector objetivo t , usando típicamente la norma euclídea.
- Para la lattice normal cuadrada, dado un vector objetivo $t = (t_x, t_y) \in \mathbb{R}^2$, la solución equivale a redondear cada coordenada de t al entero más cercano, es decir: $v_{sol} = (round(t_x), round(t_y))$ donde *round* denota el redondeo al entero más próximo.

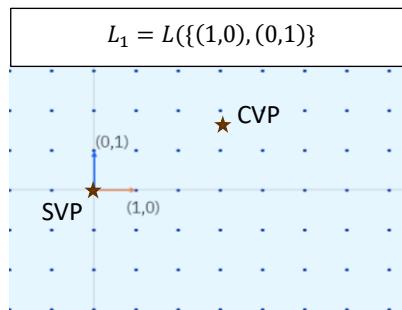


Figura 9. Shortest Vector Problem y Closest Vector Problem.

En el caso de una red cuadrada regular en dos dimensiones, tanto el **Shortest Vector Problem (SVP)** como el **Closest Vector Problem (CVP)** pueden resolverse de manera directa e intuitiva, ya que la geometría de la red es simple y visualmente clara.

Sin embargo, a medida que se aumenta la dimensión o se trabaja con bases más complejas y no ortogonales, estos problemas se vuelven computacionalmente difíciles. En tales escenarios, se recurre a técnicas como el **algoritmo de reducción de bases Lenstra–Lenstra–Lovász**, que permite aproximar soluciones eficientes al SVP y CVP en espacios de mayor dimensión [25].

Los problemas de lattices, como el Shortest Vector Problem (SVP), son NP-difíciles (NP-hard), lo que significa que resolverlos eficientemente implicaría poder resolver también todos los problemas de NP bajo ciertas reducciones.

- El algoritmo clásico más rápido que resuelve SVP tiene un tiempo heurístico computacional igual a $2^{0.292n+o(n)}$
- El algoritmo cuántico más rápido que resuelve SVP tiene un tiempo heurístico computacional igual a $2^{0.265n+o(n)}$

[26]

Goldreich–Goldwasser–Halevi Cryptosystem

Es un sistema criptográfico basado en lattices propuesto en 1997 por: Oded Goldreich, Shafi Goldwasser, Shai Halevi [27].

Es un criptosistema de clave pública cuya seguridad se basa en la dificultad del problema del vector más cercano (CVP) en redes (lattices). Aunque no es seguro hoy, GGH fue uno de los primeros esquemas prácticos de criptografía basada en lattices y abrió la puerta a la criptografía postcuántica [28].

Su procedimiento es el siguiente:

El emisor conoce una buena base B_1 de la retícula (una base "corta", con vectores pequeños y casi perpendiculares). Y pública una mala base (una base larga, con vectores más grandes y casi paralelos) B_2 que genera la misma retícula $L(B_1) = L(B_2)$.

El receptor utiliza la clave pública, B_2 , para cifrar un mensaje m . El cifrado se calcula como:

$$c = m \cdot B_2 + r ;$$

donde r es un pequeño vector de error o "ruido".

Lo que se envía es un punto cercano a la retícula (no exactamente en la retícula generada por la base pública), es decir, un punto desplazado o "shifted".

Para descifrar, el receptor usa la base privada corta para resolver un problema de tipo Closest Vector Problem (CVP) y recuperar m [24][29]

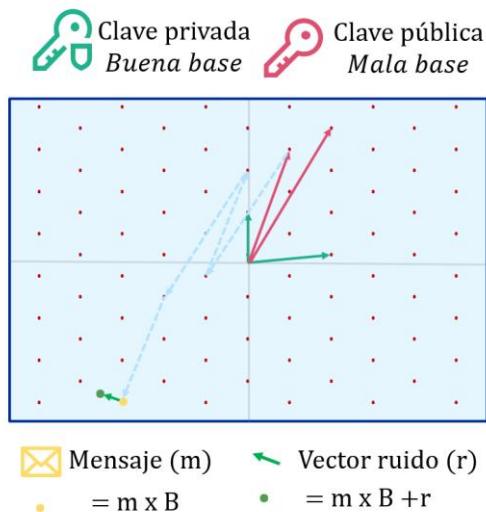


Figura 10. Goldreich-Goldwasser-Halevi Cryptosystem. Elaboración propia.

2.4.2. Problema del ruido en muestras lineales

Introducido por Regev en 2005, es un problema que consiste en resolver sistemas lineales ligeramente perturbados por errores aleatorios. Encontrar una solución aproximada a sistemas de ecuaciones ruidosas en espacios vectoriales sobre anillos modulares [30].

LWE es considerado uno de los fundamentos más sólidos de la criptografía post-cuántica y ha dado lugar a múltiples esquemas criptográficos seguros. Consiste en encontrar s tal que $A \cdot s \approx b \pmod{p}$ [31]. El proceso es el siguiente:

A partir de un vector solución s se crea un Sistema de n ecuaciones.

A un ordenador le llevaría mucho tiempo calcular la clave privada a partir de la pública, es decir, encontrar una solución al Sistema de ecuaciones para un n grande.

sistema de ecuaciones	vector secreto solución
$\begin{cases} a_{11}s_1 + a_{12}s_2 + \dots + a_{14}s_n = b_1 \\ \dots \\ \dots \\ a_{m1}s_1 + a_{m2}s_2 + \dots + a_{m4}s_n = b_m \end{cases}$	$\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}$

Figura 11. Formulación LWE sin el error añadido. Elaboración propia.

Incorporando errores a cada ecuación, se consigue un problema aún más Complejo.

Public Key sistema de ecuaciones con errores	Private key vector secreto solución
$\begin{cases} a_{11}s_1 + a_{12}s_2 + \dots + a_{14}s_n = b_1 + (e_1) \pmod p \\ \dots \quad \dots \quad \dots \\ \dots \quad \dots \quad \dots \\ a_{m1}s_1 + a_{m2}s_2 + \dots + a_{m4}s_n = b_m + (e_m) \pmod p \end{cases}$	$\begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}$

Figura 12. Formulación LWE. Elaboración propia.

Este Sistema de ecuaciones tiene muy altas probabilidades de no tener solución. Ya que es un sistema completo indeterminado al que se le ha añadido un error. Hay demasiadas ecuaciones que deben satisfacer las variables [33].

Además, estas ecuaciones están definidas sobre un campo de Galois $GF(p)$, un conjunto finito con un número primo p de elementos. Si $m \gg n$, lo más probable es que solo haya una solución s [32].

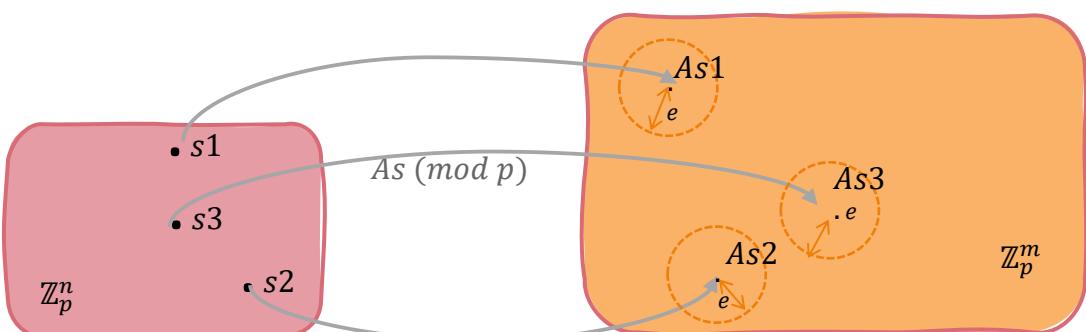


Figura 13. Representación geométrica del problema Learning With Errors (LWE): soluciones exactas desplazadas por ruido en el espacio modular. Elaboración propia.

Esto se puede aprovechar en la criptografía de clave pública de la siguiente manera:

Ejemplo simplificado de LWE:

1. El cliente toma un conjunto de ecuaciones de la clave pública (Sistema de Ecuaciones con un pequeño error).

$$\left\{ \begin{array}{l} 10x + 30y + z + 20w \equiv 34 \quad + 4 \pmod{89} \\ 20x + 25y + 0z + 28w \equiv 37 \quad + 2 \pmod{89} \\ 19x + 30y + 2z + 30w \equiv 68 \quad - 5 \pmod{89} \end{array} \right.$$

2. Suma las ecuaciones para crear una única ecuación

$$\left\{ 49x + 85y + 3z + 78w = 80 \quad + 1 \pmod{89} \right.$$

El servidor tiene un vector solución de las ecuaciones SIN ERROR; que también es solución de la ecuación resultante de sumarlas.

A. Bit a encriptar “0”

Se suma 0 al lado derecho

$$49x + 85y + 3z + 78w = 81 + 0 \pmod{89}$$

B. Bit a encriptar “1”

Se suma $p/2$ al lado derecho. En este caso $89/2 = 44$, utilizando división entera¹.

$$49x + 85y + 3z + 78w = 81 + 44 \pmod{89}$$

3. Cuando al servidor le llega una nueva ecuación (con el error y el bit encriptado). Por ejemplo, enviando el “0”

$$49x + 85y + 3z + 78w = 81 \pmod{89}$$



El servidor sustituye en la ecuación su vector secreto: $[10, 82, 50, 5]^T$ que es una solución al sistema de ecuaciones sin errores.

¿Cómo sabe si representa un 1 o un 0?

Tiene que separar el bit codificado de la solución real.

El servidor sustituye en la ecuación su vector secreto: $[10, 82, 50, 5]^T$

$$49 \cdot 10 + 85 \cdot 82 + 3 \cdot 50 + 78 \cdot 5 = 79 \pmod{89}$$

Compara con el lado derecho de la ecuación del cliente.

$$81 - 79 = 2 \rightarrow \text{bit codificado con el error (Bit real = 0)}$$

¹ La división entera da como resultado solo la parte entera del cociente, descartando el resto o la parte decimal

El bit codificado debería ser 0 o 44. En este caso el bit real es 0 ya que el bit codificado con el error está más cercano al 0 que al 44. En aritmética modular, $p/2$ es el número más alejado del 0. Entonces el servidor debería ser capaz de determinar entre algo codificado cercano a 0 o cercano a $p/2$.[32][33]

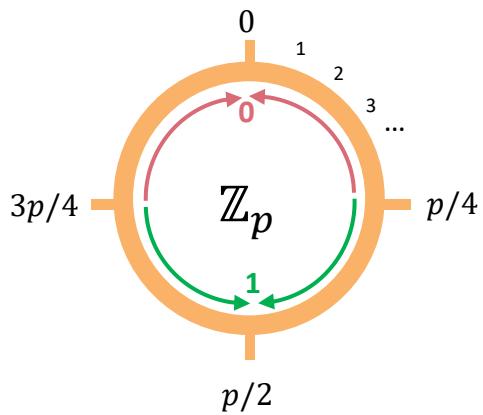


Figura 14. Representación del grupo cíclico \mathbb{Z}_p en aritmética modular.

2.4.3. Kyber

Kyber fue seleccionado como el algoritmo ganador para cifrado e intercambio de claves en la tercera ronda del concurso de criptografía post-cuántica del NIST, cuyos resultados se anunciaron el 5 de julio de 2022 [34].

En Kyber, el mensaje se codifica como un punto cercano a una retícula, en el sentido de que:

- Se parte de una retícula implícita generada por una clave pública,
 - Se añade ruido (error) controlado para ocultar el mensaje y el secreto,
 - El receptor, con una clave secreta, puede cancelar ese error y recuperar el mensaje [35].

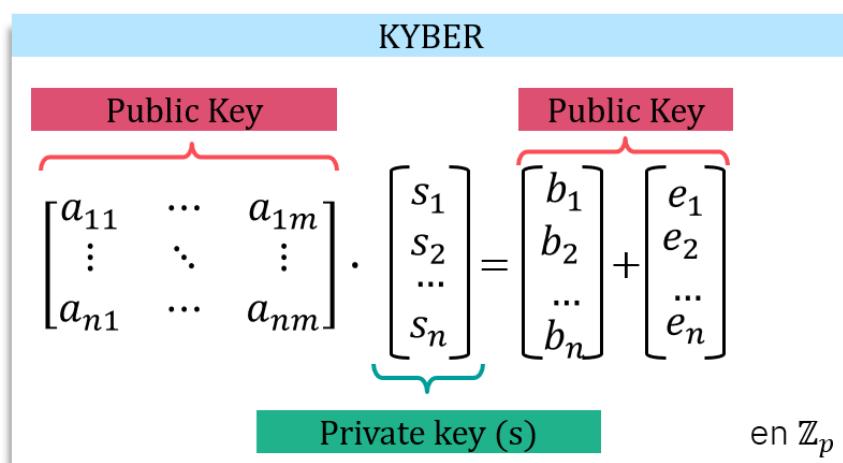


Figura 15. Representación matricial de Kyber. Elaboración propia.

Podemos ver la retícula como el conjunto de todos los vectores:

$$\mathcal{L}_A = \{A \cdot s \mid s \in \mathbb{Z}^n\}$$

donde:

$A \cdot s$ es un punto de la retícula generada por las columnas de A

e : desplaza ese punto de la retícula

$t = b + e$: es el punto perturbado, que ve el atacante.

Entonces:

Si tomamos un $s \in \mathbb{Z}^n$, el producto $A \cdot s$ sí está sobre la retícula. $A \cdot s + e$ está cerca de un punto de la retícula, pero no sobre ella [35].

Cifrado

Para cifrar un mensaje m , el emisor genera un vector secreto aleatorio r (coeficientes pequeños) y calcula:

$$u = A^T \cdot r + e_1$$

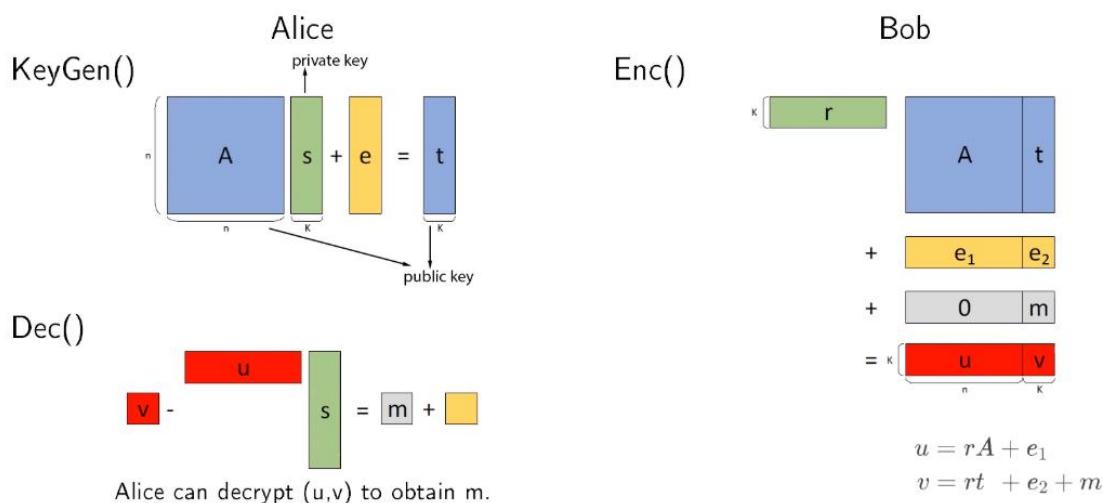
$$v = t^T \cdot r + e_2 + \text{encode}(m)$$

El par (u, v) es el ciphertext. Ambos contienen puntos cercanos a la retícula generada por A , desplazados por errores [36].

Descifrado

El receptor, que conoce s , puede computar: $v - u^T \cdot s \approx \text{encode}(m) + \text{pequeño error}$

Al aplicar una operación de redondeo (decodificación), recupera el mensaje m [36].



$$\begin{aligned}
 m_{\text{noisy}} &= v - us \\
 &= rt + e_2 + m - (rA + e_1)s \\
 &= r(As + e) + e_2 + m - (rA + e_1)s \\
 &= re + e_2 + m - e_1s \\
 &= m + (re + e_2 - e_1s)
 \end{aligned}$$

Figura 16. Representación visual de fases de Kyber. [36]

Aunque Kyber no usa una "base buena" como GGH, el secreto s permite cancelar los errores introducidos durante el cifrado, lo que emula el proceso de encontrar el punto más cercano en la retícula. Por eso, la seguridad de Kyber se basa en la dificultad de resolver el problema LWE, que se ha demostrado reducible a problemas de retículas difíciles, como: Bounded Distance Decoding (BDD), Shortest Vector Problem (SVP), o el Decisional LWE [33].

2.5 ENTIDADES DE SEGURIDAD POSTCUÁNTICA

Las entidades de seguridad postcuántica juegan un papel clave en el desarrollo de nuevos algoritmos criptográficos resistentes a los ordenadores cuánticos. En esta sección se describen algunas de las principales entidades que lideran este esfuerzo.

2.5.1. NIST

Uno de los mayores impulsores de la transición hacia la criptografía post cuántica es el Instituto Nacional de Estándares y Tecnología (NIST), que desde 2016 lidera un proceso internacional de evaluación y estandarización de nuevos algoritmos basados en problemas matemáticos como Lattices y LWE. Este proyecto tiene como objetivo garantizar la continuidad de la seguridad digital en un futuro donde los ordenadores cuánticos puedan romper los esquemas criptográficos actuales [34].

Según la directora del NIST, Laurie E. Locascio, “Our post-quantum cryptography program has leveraged the top minds in cryptography — worldwide — to produce this first group of quantum-resistant algorithms that will lead to a standard and significantly increase the security of our digital information” [37].

A través de un proceso abierto y competitivo, el NIST ha evaluado múltiples propuestas y ha seleccionado un conjunto de algoritmos finalistas y candidatos recomendados. Mientras el estándar aún se encuentra en desarrollo, la propia institución recomienda a los expertos en seguridad que estudien los nuevos algoritmos y evalúen cómo se integrarán en sus aplicaciones. No obstante, advierte que todavía no deben incorporarse de forma definitiva, ya que podrían sufrir ajustes antes de su estandarización final [34].

2.5.2. Open Quantum Safe Project

El Proyecto Open Quantum Safe (OQS) es una iniciativa de código abierto cuyo objetivo es facilitar la adopción de algoritmos de criptografía post-cuántica (PQC) en aplicaciones reales. Para ello, proporciona herramientas, bibliotecas y entornos de prueba donde experimentar con los algoritmos candidatos a la estandarización por parte del NIST [6].

Gracias a su filosofía de código abierto, el proyecto OQS fomenta la colaboración entre investigadores, profesionales de la seguridad y desarrolladores de software, contribuyendo activamente a la validación y adopción de estándares de criptografía post cuántica a nivel global. Su componente principal es LibOQS (Library for Open Quantum Safe).

2.6. LIBOQS

LibOQS (Library for Open Quantum Safe) es una biblioteca de C desarrollada por el proyecto Open Quantum Safe. LibOQS está integrado en OpenSSL a través de su API en C y ofrece una interfaz unificada que permite a los desarrolladores implementar algoritmos post cuánticos en aplicaciones existentes, sin necesidad de modificar profundamente sus arquitecturas criptográficas [6].

El catálogo completo de algoritmos que ofrece LibOQS puede consultarse en su repositorio oficial en GitHub: <https://github.com/open-quantum-safe/liboqs>. A continuación, se ha hecho una clasificación de los algoritmos listados para un mejor entendimiento.

Algoritmos de firma digital post-cuánticos

	Algoritmos	Descripción	Concurso NIST	Algoritmos Híbridos
CRYSTALS-Dilithium	dilithium2, dilithium3, dilithium5	Basado en retículos (lattices).	Ganador	p256_dilithium2 (L2), p384_dilithium3 (L3), p521_dilithium5 (L5) rsa3072_dilithium2
Falcon	falcon512, falcon1024	Basado en retículos (NTRU).	Finalista	p256_falcon512 (L1) p521_falcon1024 (L5) rsa3072_falcon512,
SPHINCS+ SHA2	sphincssha2128 fsimple, sphincssha2128 ssimple, sphincssha2192 fsimple, etc.	Basado en árboles hash.	Finalista	p256_sphincssha2128fsimple (L1) p384_sphincssha2192fsimple (L3), p521_sphincssha2256fsimple (L5) rsa3072_sphincssha2128fsimple,
SPHINCS+ SHAKE	sphincsshake128fsimple, sphincsshake128ssimple, sphincsshake192fsimple, etc.	Variante de SPHINCS+ con función hash SHAKE en lugar de SHA2.	Finalista	rsa3072_sphincsshake128fsimple, p256_sphincsshake128fsimple (L1), p384_sphincsshake192fsimple (L3), p521_sphincsshake256fsimple (L5)

Tabla 2. Algoritmos postcuánticos de firma digital ofrecidos por LibOQS en la versión 0.12.0. Elaboración propia.

Algoritmos de Encapsulación de clave (KEM) Quantum Safe:

	Algoritmos	Descripción	Concurso NIST	Combinaciones híbridas
BIKE	bikel1, bikel3, bikel5	Basado en códigos binarios (bit-flipping).	Finalista	p256_bikel1, p384_bikel3, p521_bikel5
CRYSTALS-Kyber*	kyber512, kyber768, kyber1024	Basado en redes euclidianas (lattices).	Ganador	p256_kyber512, p384_kyber768, p521_kyber1024
FrodoKEM	frodo640aes, frodo640shake, frodo976aes, frodo976shake, frodo1344aes, frodo1344shake	Basado en matrices aleatorias (Learning With Errors, sin estructura).	Candidato	p256_frodo640aes, p384_frodo976aes, p521_frodo1344aes
HQC	hqc128, hqc192, hqc256 [†]	Basado en códigos (code-based).	Finalista	p256_hqc128, p384_hqc192, p521_hqc256

Tabla 3. Algoritmos postcuánticos KEM ofrecidos por LibOQS en la versión 0.12.0. Elaboración propia.

Curva	Seguridad aproximada	Uso típico
P256	~128 bits (NIST L1)	Navegadores, TLS, móviles
P384	~192 bits (NIST L3)	Entornos con mayor exigencia
P521	~256 bits (NIST L5)	Seguridad máxima, entornos críticos

Tabla 4. Curvas elípticas de algoritmos híbridos ofrecidos por LibOQS. Elaboración propia.

Para ejecutar los algoritmos listados anteriormente, se han utilizado contenedores en configuración servidor-cliente provistos por el proyecto LibOQS, los cuales permiten probar implementaciones de algoritmos postcuánticos integrados en OpenSSL.

2.7. TECNOLOGÍAS

A continuación, se describe el protocolo de seguridad TLS y su infraestructura criptográfica, con énfasis en su adaptación a entornos post-cuánticos mediante la incorporación de algoritmos resistentes a ataques cuánticos.

2.7.1. Protocolo TLS

Transport Layer Security (TLS) es un protocolo que incluye una serie de normas sobre cómo comunicarse de manera segura por internet. Está definido por organizaciones y hay varias versiones, siendo TLS 1.3 la más actual. Cuando un cliente (navegador) se conecta a un servidor (sitio web), TLS garantiza que la comunicación sea segura y cifrada [38].

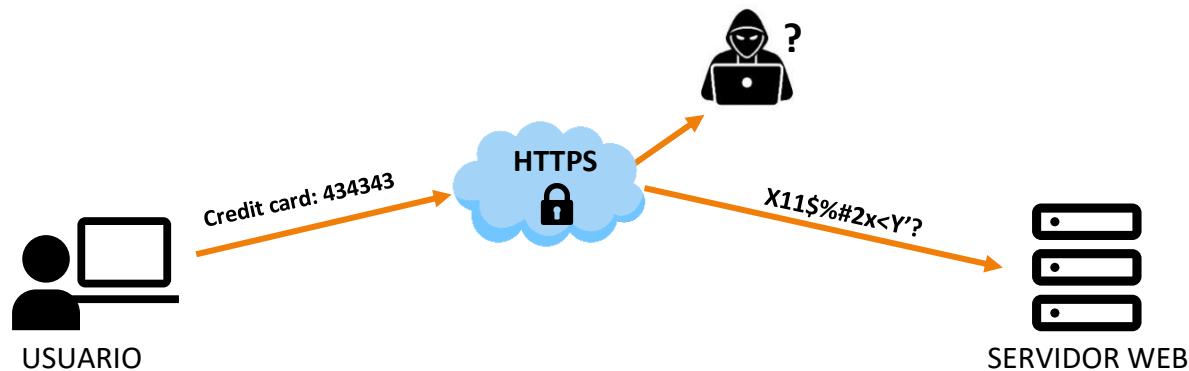


Figura 17. Implementación de HTTP junto a TLS para crear una comunicación segura HTTPS. Elaboración propia.

TLS es una de las opciones más utilizadas para comunicaciones en internet, ya que protege las comunicaciones en aplicaciones web, correo electrónico y servicios internos. Grandes empresas como Facebook, Twitter y Google utilizan TLS para mantener seguras las conexiones de miles de millones de personas [39].



In January this year (2010), Gmail switched to using HTTPS for everything by default. Previously it had been introduced as an option, but now all of our users use HTTPS to secure their email between their browsers and Google, all the time. On our production frontend machines, SSL/TLS accounts for less than 1% of the CPU load.... - Adam Langley (Google) [40]



We have deployed TLS at a large scale using both hardware and software load balancers. We have found that modern software-based TLS implementations running on commodity CPUs are fast enough to handle heavy HTTPS traffic load without needing to resort to dedicated cryptographic hardware... - Doug Beaver (Facebook) [41]



TLS connections are required in order to access X API endpoints. Communicating over TLS preserves user privacy and security by protecting information between the user and the X API as it travels across the public Internet. Connections to the X API require TLS version 1.2. [42]

TLS utiliza criptografía para intercambiar claves de forma segura, autenticar a los participantes, cifrar los datos transmitidos y garantizar la integridad de los mensajes. Esto se logra mediante algoritmos de intercambio de claves, firmas digitales, cifrado simétrico, y códigos de autenticación [43].

La criptografía simétrica y asimétrica, se combina en el protocolo TLS de la siguiente manera:

Protocolo de Negociación (Handshake)	Negocia la versión de TLS a usar Selecciona los algoritmos criptográficos (cipher suites). Establece una clave secreta compartida mediante criptografía asimétrica (clave pública). Autentica al servidor (y opcionalmente al cliente) mediante firmas digitales.
Protocolo de Transmisión de datos	Utiliza criptografía simétrica para Cifrar y descifra mensajes usando la clave secreta compartida. Garantiza la integridad y autenticidad de los mensajes mediante códigos de autenticación.

Tabla 5. Fases del protocolo TLS. Elaboración Propia.

Durante la fase inicial de negociación, TLS utiliza criptografía de clave asimétrica para intercambiar de manera segura la clave secreta, que luego servirá para cifrar el resto de la comunicación [44]. Sin embargo, este esquema podría volverse vulnerable en el futuro frente a ordenadores cuánticos, que podrían romper los algoritmos asimétricos actuales y comprometer la seguridad del intercambio de claves. **TLS Quantum Safe** es una variante del protocolo TLS que incorpora nuevos algoritmos criptográficos de clave asimétrica que están diseñados para resistir ataques de ordenadores cuánticos (postcuánticos).

	TLS Tradicional	TLS Quantum-Safe
Algoritmos	RSA, ECDSA, ECDH	Kyber, Dilithium, FrodoKEM, ...
Seguridad contra cuántica	Vulnerable al algoritmo de Shor	Resistente al algoritmo de Shor
Implementación	OpenSSL	OpenSSL modificado

Tabla 6. Comparativa TLS Tradicional vs TLS Quantum Safe. Elaboración propia.

TLS es un protocolo, es decir, un conjunto de normas sobre cómo debe realizarse la comunicación segura; mientras que OpenSSL es una implementación práctica mediante código de ese protocolo. La biblioteca LibOQS está diseñada para integrarse con OpenSSL a través de su API en C, permitiendo incorporar algoritmos post-cuánticos en implementaciones prácticas de TLS (como HTTPS).

2.7.1. OpenSSL

OpenSSL es una biblioteca de software de código abierto que proporciona herramientas y funciones para implementar comunicaciones seguras mediante criptografía. Es utilizado por servidores web, clientes de correo electrónico, navegadores, VPNs y aplicaciones personalizadas. Sus funciones criptográficas son eficientes y compatibles con diferentes SO como Linux, Windows y macOS [45].

Al ser de código abierto, OpenSSL puede ser modificado para integrar nuevas funcionalidades. En este proyecto, se ha utilizado una versión personalizada que incorpora la biblioteca LibOQS, lo que permite habilitar algoritmos criptográficos postcuánticos que no están disponibles en la versión oficial de OpenSSL.

2.8. TRABAJOS SIMILARES

El campo de la criptografía post-cuántica es un área activa de investigación dada la inminente amenaza que la computación cuántica representa para los sistemas criptográficos tradicionales. Varios proyectos e investigaciones se han centrado en el desarrollo y la evaluación de algoritmos resistentes a ataques cuánticos.

[46] El artículo titulado “**Post-quantum public key algorithms selected in liboqs**”, presentado en el Simposio SCIS 2018 en Kanazawa, Japón, ofrece un análisis exhaustivo de

los algoritmos de intercambio de claves resistentes a ataques cuánticos integrados en la biblioteca de código abierto liboqs del proyecto Open Quantum Safe (OQS).

Este estudio proporciona una base para investigaciones que buscan implementar soluciones criptográficas resistentes a ataques cuánticos. La evaluación comparativa de los protocolos en liboqs ofrece información valiosa sobre las fortalezas y debilidades de cada enfoque, facilitando la selección informada de algoritmos para aplicaciones específica

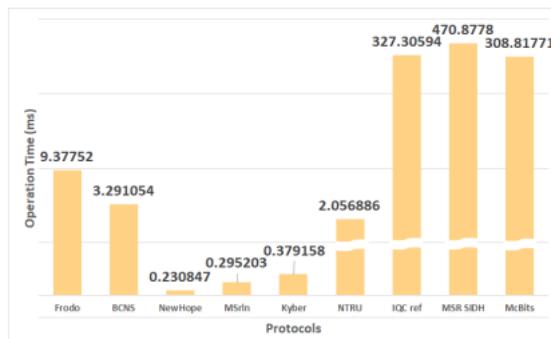


Figura 18. Comparación de tiempo de ejecución de protocolos OQS [46]

Entorno experimental:

- Procesador: Intel(R) Core i7-5500U
- Memoria RAM: 16 GB
- Sistema Operativo: Ubuntu 16.04
- Compilador: GCC v5.4.0
- Código fuente: Se utilizó la versión de referencia de liboqs descargada desde GitHub

Resultados Clave

Los protocolos basados en retículas, como Kyber y NewHope, mostraron un equilibrio favorable entre seguridad y eficiencia.

El artículo concluye que, si bien varios protocolos muestran promesas, es esencial continuar con la investigación y optimización para garantizar una transición segura hacia sistemas criptográficos post-cuánticos. Se sugiere explorar combinaciones de diferentes enfoques criptográficos y mejorar la integración de estos protocolos en infraestructuras existentes.

[47] El artículo titulado "**A Comparison of Security and its Performance for Key Agreements in Post-Quantum Cryptography**" (IEEE, 2020) ofrece un análisis comparativo de diversos algoritmos de intercambio de claves diseñados para resistir ataques de computadoras cuánticas.

Entorno experimental

- Procesador: Intel Core i7-6500U @ 2.50 GHz (cuatro núcleos)
- Memoria RAM: 16 GB

- Sistema Operativo: Ubuntu 20.04 (64 bits)
- Código fuente: programas en lenguaje SAGE

Resultados Clave

- Kyber y NewHope: Destacan por su equilibrio entre seguridad y eficiencia, siendo candidatos fuertes para la estandarización.
- FrodoKEM: Ofrece altos niveles de seguridad, pero a costa de mayores recursos computacionales.
- NTRU: Presenta un buen rendimiento, aunque su seguridad ha sido objeto de debate en la comunidad criptográfica.

El estudio concluye que, aunque no existe una solución única que se adapte a todas las necesidades, algunos algoritmos como Kyber y NewHope muestran un equilibrio prometedor entre seguridad y rendimiento. La elección del algoritmo adecuado dependerá del contexto específico de aplicación y de los requisitos de seguridad y eficiencia.

[48] La tesis de Kiros Meles, titulada "**Implementation and Performance Evaluation of Quantum-Safe Cryptography in C: A Comparative Study Using LibOQS and OpenSSL**" (Universidad de Padua, 2023–2024), ofrece un análisis comparativo entre algoritmos criptográficos tradicionales y post-cuánticos, centrándose en su implementación y rendimiento utilizando las bibliotecas LibOQS y OpenSSL.

La investigación involucró la implementación de diversos algoritmos criptográficos, tanto clásicos como post-cuánticos, utilizando las bibliotecas LibOQS y OpenSSL.

Entorno de prueba

Las pruebas se realizaron en un equipo Lenovo IdeaPad 330-15IKB, con las siguientes especificaciones:

- Procesador: Intel Core i5 de 8^a generación a 2.5 GHz
- Almacenamiento: Disco duro de 1 TB (HDD)
- Memoria RAM: 4 GB
- Sistema Operativo: Ubuntu 22.04 LTS (64 bits)

Para las evaluaciones se utilizaron las siguientes versiones de bibliotecas:

- LibOQS: Versión 0.10.0, lanzada el 23 de marzo de 2024
- OpenSSL: Versión 3.0.13, lanzada el 30 de enero de 2024

Ambas bibliotecas fueron evaluadas bajo estas condiciones de sistema, utilizando la implementación descrita en la Sección 4 de la tesis.

La tesis concluye que tanto LibOQS como OpenSSL proporcionan soluciones viables para la criptografía postcuántica. Sin embargo, la elección del algoritmo adecuado debe basarse en

los requisitos específicos de la aplicación, priorizando la velocidad o la seguridad según el caso. Se recomienda continuar con la optimización y pruebas en entornos reales para mejorar la aplicabilidad práctica de estos algoritmos.

[49] El **proyecto de benchmarking de Open Quantum Safe (OQS)** proporciona una plataforma para visualizar el rendimiento de algoritmos criptográficos post-cuánticos, centrándose en aspectos como la velocidad de ejecución y el consumo de memoria.

“This project is not currently maintained, and these measurements are not up to date. The data contained here are not representative of the current liboqs main branch, nor are they representative of the liboqs main branch at the time they were collected.” – Open Quantum Safe [49]

Aunque el proyecto de perfilado original ya no se mantiene y sus datos no representan el estado actual de la biblioteca liboqs, los datos disponibles ofrecen una visión general de las capacidades de diferentes algoritmos en diversas arquitecturas.

2.9. CONCLUSIONES DEL ESTADO DEL ARTE

Este proyecto parte de los avances de investigaciones previas, centrándose en los algoritmos postcuánticos que han mostrado el mejor rendimiento en pruebas anteriores. Concretamente, los algoritmos basados en Lattices y Learning With Errors (LWE) son los más destacados en trabajos anteriores y por ello se han analizado en mayor profundidad.

Por otra parte, la librería Liboqs está en constante actualización. Realizar pruebas con las últimas actualizaciones ayuda en una tarea de seguimiento que estudia el desempeño de los nuevos algoritmos. Este proyecto se lleva a cabo sobre la [versión 0.12.0](#) de LibOQS, publicada en diciembre de 2024.

A partir de esta base, se ha desarrollado un proyecto que abarca desde la infraestructura necesaria para la simulación, hasta la ejecución y estudio de los algoritmos en un entorno controlado. Todas las ejecuciones se han realizado en un ordenador estándar, proporcionando una referencia clara sobre los tiempos que los usuarios podrían experimentar al operar con sus equipos promedio. Además, los resultados se han comparado con los de otro proyecto realizado en un equipo de diferentes características. Asimismo, se ha documentado el espacio de almacenamiento requerido por cada componente del entorno de pruebas, aportando una visión más completa del impacto computacional.

Este proyecto contribuye al campo de la criptografía al analizar en profundidad el protocolo TLS Quantum Safe, explicando cómo se asegura la comunicación en línea y detallando los recursos necesarios para la transición. A través de este enfoque, se ha descrito el impacto de la criptografía postcuántica y detallado recomendaciones para su implementación práctica en la empresa.

3. OBJETIVOS

En este apartado se definen los objetivos que orientan el desarrollo del proyecto, desde una meta general hasta objetivos específicos y los métodos que permitirán validar su cumplimiento.

3.1. OBJETIVO GENERAL

El objetivo general del proyecto fue analizar la viabilidad de la adopción de algoritmos postcuánticos. Teniendo en cuenta los esfuerzos de despliegue, rendimiento y seguridad que ofrecen respecto a los esquemas actuales. Con ello, se busca contribuir a la transición criptográfica de las empresas de manera controlada y consciente.

3.2. LISTA DE OBJETIVOS ESPECÍFICOS

Para alcanzar el objetivo general planteado, se definieron varios objetivos específicos que permitieron organizar el proyecto en etapas concretas y evaluables:

- Conocer la base matemática de los algoritmos criptográficos actuales y comprender cómo la computación cuántica pone en riesgo estos esquemas.
- Analizar las iniciativas globales y las entidades que desarrollan criptografía postcuántica. Estudiar los algoritmos en proceso de estandarización por el NIST y entender su fundamentación matemática.
- Crear un entorno de pruebas en el que poder realizar conexiones TLS Quantum Safe. Y utilizar las herramientas proporcionadas por LibOQS para la implementación de los algoritmos.
- Analizar los resultados y ofrecer conclusiones de valor para las empresas, orientadas a la selección de algoritmos postcuánticos según el nivel de seguridad requerido y la capacidad computacional disponible.

3.3. MÉTODOS DE VALIDACIÓN

Con el fin de asegurar que los objetivos específicos del proyecto han sido alcanzados, se establecerán distintos métodos de validación. Estos métodos permitirán comprobar la viabilidad técnica, la calidad del desarrollo y la comprensión del impacto que los protocolos de encriptación Quantum Safe pueden tener en el contexto de la ciberseguridad moderna.

A continuación, se detallan los métodos asociados a cada objetivo específico:

- Documentar la formulación matemática de los algoritmos y exponerla ante profesores para evaluar la comprensión y fundamentación.
- Recopilación bibliográfica exhaustiva de documentos, artículos y estándares oficiales relacionados con criptografía postcuántica y estandarización.
- Verificación mediante comandos que confirmen el despliegue correcto, tales como revisión de versiones, estado de imágenes activas y conectividad entre contenedores. Documentación de los pasos y resultados del despliegue.
- Ejecución exitosa de conexiones TLS Quantum Safe entre cliente y servidor, recogida sistemática de métricas de rendimiento, presentación de análisis estadísticos y gráficos que evidencien los resultados.
- Redacción y presentación de un informe final con conclusiones fundamentadas, basado en el análisis empírico y teórico desarrollado, validado mediante revisión por expertos en ciberseguridad.

4. METODOLOGÍA Y PLAN DE TRABAJO

Para el desarrollo de este proyecto se ha optado por una combinación de las metodologías Cascada y Diseño Experimental, ya que se ajustan de manera óptima a las características técnicas y al enfoque empírico del trabajo.

Dado que el objetivo central del proyecto es evaluar la viabilidad de algoritmos de criptografía postcuántica en el protocolo TLS, se ha adoptado el diseño experimental como metodología para la fase de pruebas y análisis. Esta metodología permite definir variables controladas (como el tipo de algoritmo y tamaño de clave) y medir sus efectos sobre las variables de interés (tiempo de ejecución y seguridad).

Entorno experimental:

Procesador: Intel® Core™ i5-1135G7 (11^a generación) @ 2.40 GHz (8 CPUs, ~2.6 GHz)

Memoria RAM: 8 GB

Sistema Operativo: Ubuntu 22.04 LTS

Código fuente: contenedores LibOQS descargados de Docker Hub.

Por otra parte, la metodología en cascada es un modelo secuencial clásico que permite organizar el proyecto en fases ordenadas: análisis, diseño, implementación, pruebas y evaluación [50]. En este proyecto, las fases se han organizado por paquetes de trabajo y cada una ha sido completada antes de pasar a la siguiente.

ANÁLISIS DE IMPACTO DE LA CRIPTOGRAFÍA POSTCUÁNTICA

Universidad Francisco de Vitoria
Grupo B

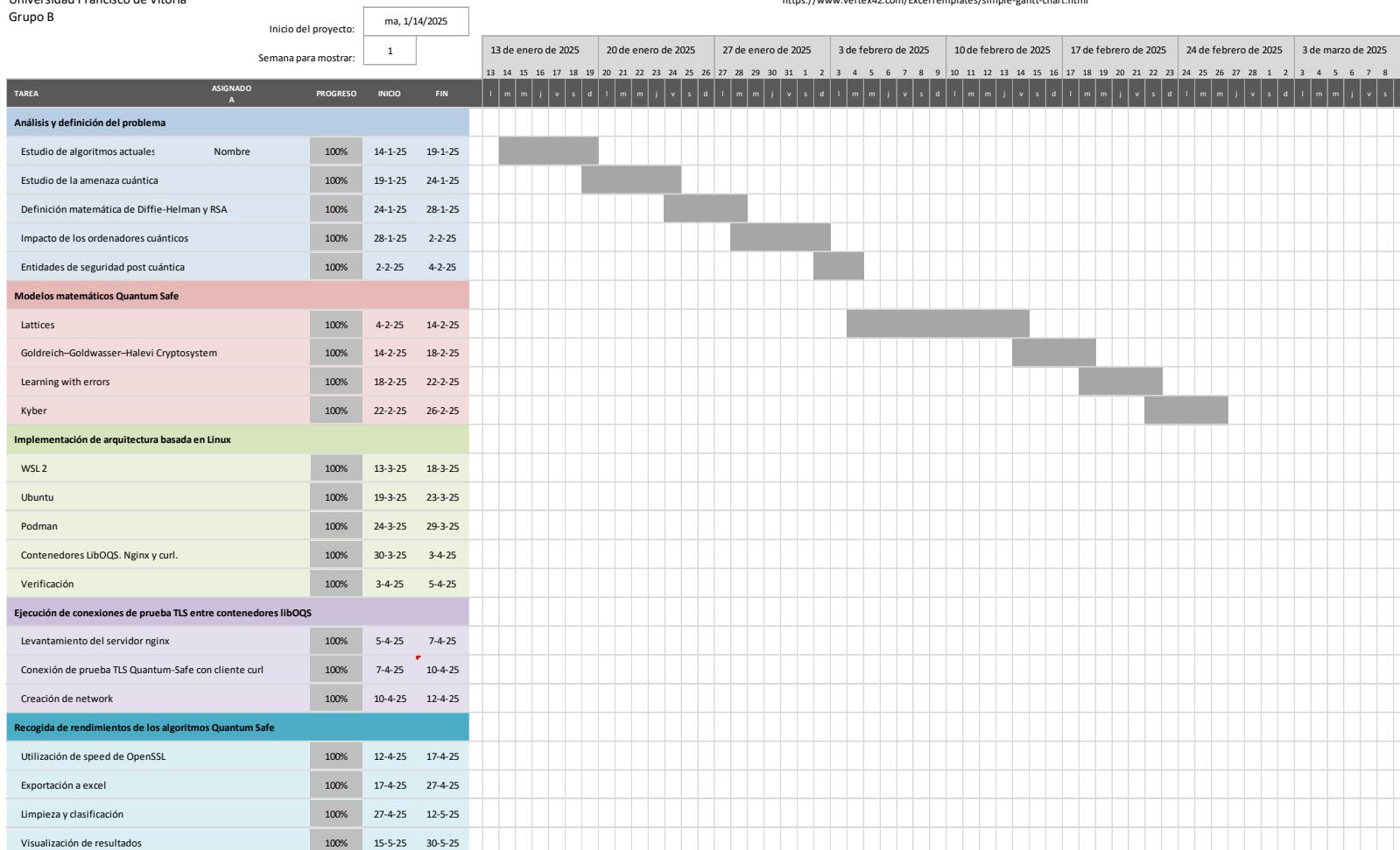


Figura 19. Diagrama de Gantt del proyecto

4.1. MATERIALES Y RECURSOS

Para el desarrollo del entorno de pruebas y la ejecución de experimentos, se han empleado los siguientes recursos:

- WSL2 junto con una distribución Ubuntu 22.04, que sirvió como sistema base para desplegar y ejecutar las herramientas necesarias en un entorno Linux.
- Podman como motor de contenedores, utilizado para la gestión eficiente de entornos aislados. [Ver Anexo F].
- Imágenes de contenedores y bibliotecas criptográficas proporcionadas por el proyecto Open Quantum Safe (OQS).
- Repositorio liboqs en Github, que ofrece soporte para algoritmos de criptografía post-cuántica y documentación asociada.
- Aplicación Python, para el procesamiento y análisis de resultados. Concretamente, se han utilizado las bibliotecas Seaborn y Matplotlib para la generación de gráficos, así como Pandas y NumPy para la manipulación de datos y estructuras tipo dataframe.
- Portátil Lenovo 11th Gen Intel ® Core ™ i5-1135G7 @ 2.40GHz (8CPUs), con procesador Intel Core i5, ~2.6GHz. 8 GB de RAM.

Además, se ha realizado el curso online " Practical Introduction to Quantum-Safe Cryptography", ofrecido por IBM Quantum Learning [51]. Esta formación ha servido de guía para la comprensión e implementación de los conceptos aplicados en el proyecto.

En cuanto a recursos humanos se refiere, el proyecto ha sido desarrollado de manera íntegra por Ana Robledano, contando con la supervisión técnica por parte de Ginés Carrascal y José Cándido Carballido.

4.2. MÉTODO

En esta sección se describe la metodología que se ha empleado para llevar a cabo el proyecto, se incluyen explicaciones sobre los algoritmos criptográficos utilizados, así como las herramientas y tecnologías implementadas para construir el entorno experimental.

También se define el desarrollo de la prueba de concepto que se ha enfocado en la implementación práctica y validación de conexiones TLS Quantum Safe. Posteriormente, se han recogido métricas de rendimiento a gran escala para su análisis.

4.2.1. Infraestructura basada en Linux

Linux es un núcleo (kernel) de sistema operativo de código abierto, creado por Linus Torvalds en 1991. Es el componente central de muchos sistemas operativos, conocidos como

distribuciones de Linux, que incluyen software adicional para hacer que el sistema sea completo y funcional [52].

El núcleo Linux es responsable de gestionar hardware como el procesador, la memoria, el almacenamiento, etc y es el sistema operativo dominante en servidores web [53]. Por este motivo, se utiliza en el proyecto como entorno donde desplegar contenedores y realizar conexiones TLS.

Concretamente se ha elegido la distribución de Linux, Ubuntu debido a su flexibilidad y compatibilidad con herramientas de criptografía post-cuántica. Sin embargo, el mismo proceso podría replicarse de manera similar en cualquier otra distribución de Linux, ya que todas comparten el mismo núcleo (kernel).

Para utilizar el Sistema Operativo Linux, se necesita una Maquina Virtual (VM) que emplea software en lugar de hardware para ejecutar programas e implementar aplicaciones. Concretamente, se ha utilizado WSL 2 (Windows Subsystem for Linux), que cuenta con el hipervisor² de Windows llamado Hyper-V. En este caso, Hyper-V permite ejecutar un kernel de Linux real en una máquina virtual liviana dentro de Windows.

4.2.2. Adaptación a Contenedores de Podman

Los contenedores son unidades ejecutables de software que empaquetan el código de la aplicación junto con sus bibliotecas y dependencias. Permiten que el código se ejecute en cualquier entorno informático, ya sea de sobremesa, TI tradicional o infraestructura en la nube [54]. Las organizaciones habitualmente despliegan contenedores sobre máquinas virtuales para mejorar el aislamiento y la seguridad [55].

En este trabajo se han utilizado contenedores cliente-servidor que soportan algoritmos post-cuánticos mediante la biblioteca LibOQS, las imágenes oficiales se encuentran disponibles en Docker Hub:

`openquantumsafe/curl`

`openquantumsafe/nginx`

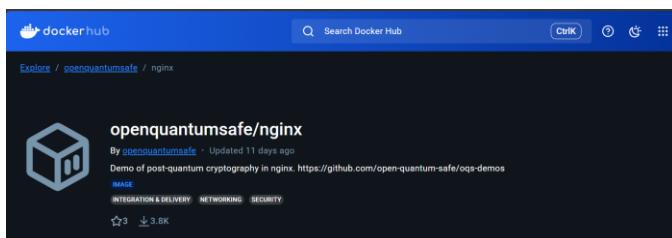


Figura 20. Contenedores de Docker diseñados para soportar algoritmos de cifrado post-cuántico mediante la biblioteca LibOQS [56][57]

Support <code>curl</code> passing Maintained: @baentsch, @bhess, @pi-314159	<code>nginx</code> passing Maintained: @baentsch, @p-314159
--	--

Figura 21. Repositorios de contenedores curl y nginx en Github [58].

² El software de hipervisor permite que múltiples sistemas operativos invitados virtuales se ejecuten simultáneamente en un solo equipo físico anfitrión.

Aunque estas imágenes están diseñadas para usarse con Docker, en el marco de este trabajo se ha optado por el uso de Podman como motor de contenedores, ya que es una alternativa compatible y sin daemon. (Anexo G: Podman)

Cada contenedor incluye:

- OpenSSL modificado con soporte para algoritmos postcuánticos.
- LibOQS como biblioteca instalada que implementa algoritmos criptográficos resistentes a la computación cuántica.
- Herramientas de línea de comandos para hacer pruebas.

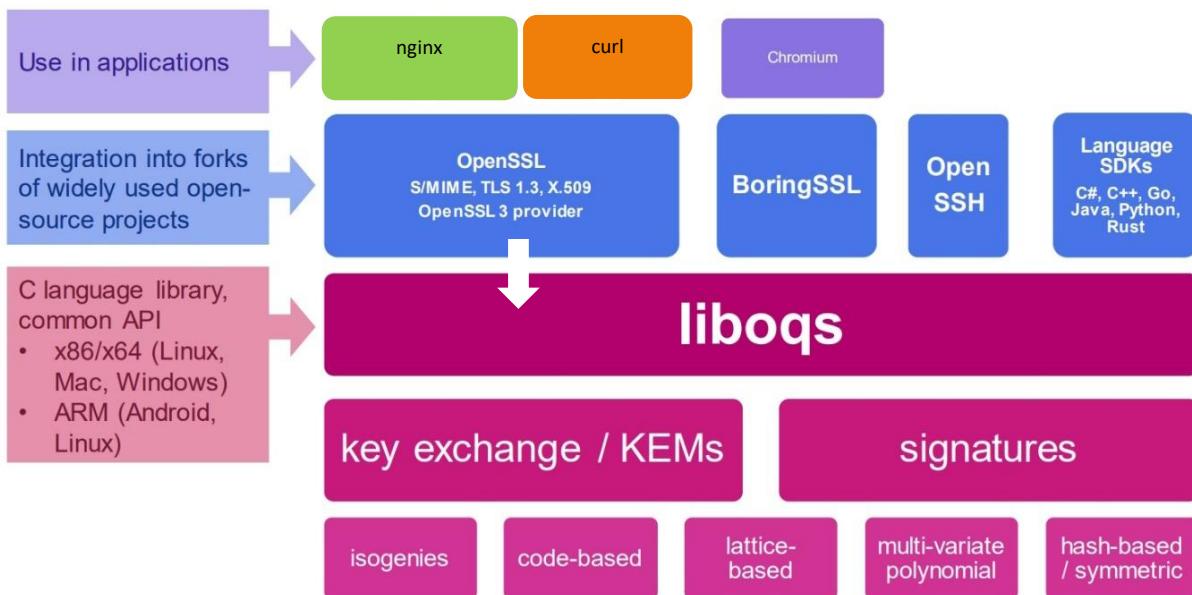


Figura 22. Interacción de la librería LibOQS con los contenedores nginx y curl [59].

OpenSSL dentro del contenedor nginx llama a las funciones de LibOQS cuando necesita usar algoritmos post-cuánticos como Kyber (para intercambio de claves) que no están disponibles en la versión estándar de OpenSSL. OpenSSL sigue haciendo la gestión de las conexiones TLS, pero cuando se usa un algoritmo post-cuántico, delega esa parte concreta en LibOQS. Es decir, OpenSSL llamará a las funciones de LibOQS para procesar esas operaciones específicas Quantum Safe.

4.2.3. Simulación conexiones TLS Servidor cliente

Las imágenes de curl y nginx explicadas anteriormente permiten experimentar con conexiones TLS basadas en el estándar TLS 1.3, utilizando algoritmos híbridos y resistentes a ataques cuánticos.

- Nginx (servidor) es un software muy utilizado en internet que actúa como servidor, es decir, el lugar que envía información cifrada a los usuarios que se conectan. Cuando miles de personas acceden al mismo servicio al mismo tiempo, NGINX reparte las solicitudes para que no colapse [60].

- Curl (cliente) es el programa que solicita la información a un servidor y comprueba que la conexión es segura, igual que haría un navegador o una aplicación cuando accede a un sitio web [61].

Páginas web de grandes empresas como Netflix, Adobe, Spotify o Dropbox usan NGINX para gestionar las conexiones [62]. Los contenedores de libOQS permiten probar en un entorno controlado conexiones TLS Quantum Safe entre un servidor y un cliente.

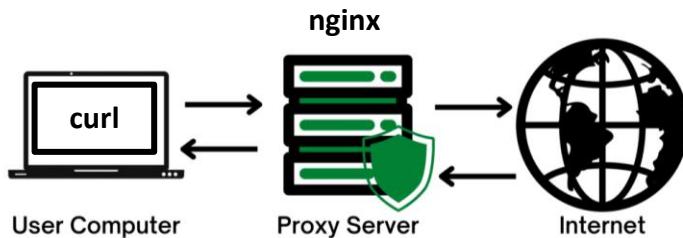


Figura 23. Conexión entre un cliente curl y un servidor nginx [63]

4.2.4. Validación de las conexiones

Para realizar conexiones TLS entre los contenedores curl y nginx con soporte para criptografía post-cuántica, se siguen los siguientes pasos:

1. Levantar el servidor NGINX post-cuántico

Ejecutar el siguiente comando para iniciar un contenedor con la imagen modificada de NGINX, exponiendo el puerto 4433:

```
podman run -d -p 4433:4433 docker.io/openquantumsafe/nginx
```

Esto inicia un servidor HTTPS accesible en: <https://localhost:4433>

2. Abrir la terminal interactiva de curl post-cuántico

Para ejecutar comandos dentro del contenedor curl con soporte para algoritmos cuánticos, se utiliza:

```
podman run -it docker.io/openquantumsafe/curl
```

3. Realizar la conexión TLS usando un algoritmo post-cuántico

Dentro del contenedor curl, se ejecuta:

```
curl --curves kyber768 https://localhost:4433
```

En este comando, curl utiliza el algoritmo de intercambio de claves Kyber768, que forma parte de la criptografía post-cuántica, para negociar la conexión TLS. La URL <https://localhost:4433> corresponde al servidor NGINX post-cuántico previamente iniciado.

Como resultado se obtiene un html con el registro de la conexión.

```

Supported groups: kyber768
Shared groups: kyber768
---
New, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
SSL-Session:
    Protocol : TLSv1.3
    Cipher   : TLS_AES_256_GCM_SHA384
    Session-ID: 701B6151C6E2B2A9393A9DFC2CD6345B92BFB4A51E72573EA33A3E86D75CB6C8
    Session-ID-ctx: 01000000
    Resumption PSK: 980C21911AE4A202E4E7E423EBACE5CA795944347E46422DD1E6C26D2A64233E3D721EE1F0950A95A89FD7A5B24B405B
    PSK identity: None
    PSK identity hint: None
    SRP username: None
    Start Time: 1746518838
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
    Extended master secret: no
    Max Early Data: 0
---
0 items in the session cache
0 client connects (SSL_connect())
0 client renegotiates (SSL_connect())
0 client connects that finished
1 server accepts (SSL_accept())
0 server renegotiates (SSL_accept())
1 server accepts that finished
0 session cache hits
0 session cache misses
0 session cache timeouts
0 callback cache hits
0 cache full overflows (128 allowed)
---
no client certificate available

```

Output 1 fragmento de log de conexión TLS con Kyber 768

El servidor admite el algoritmo Kyber768, que es un mecanismo de intercambio de claves post-cuántico (KEM). El cliente y el servidor negociaron con éxito el uso de Kyber768 como grupo de intercambio de claves.

Mediante comandos en los contenedores de Open Quantum Safe, también es posible y registrar el rendimiento en términos de tiempos de ejecución y cantidad de conexiones establecidas.

Los comandos:

[Podman run -it docker.io/openquantumsafe/curl openssl list -kem-algorithms](https://podman.io/command-reference/podman-run.html)

[Podman run -it docker.io/openquantumsafe/curl openssl list -signature-algorithms](https://podman.io/command-reference/podman-run.html)

Muestran listas de los algoritmos de encapsulación de clave y de firma digital disponibles.

```

anruki@MSI:~$ podman run -it docker.io/openquantumsafe/curl openssl list -kem-algorithms
{ 1.2.840.10045.2.1, EC, id-ecPublicKey } @ default
{ 1.2.840.113549.1.1.1, 2.5.8.1.1, RSA, rsaEncryption } @ default
{ 1.3.101.111, X448 } @ default
{ 1.3.101.110, X25519 } @ default
frodo640shake @ oqsprovider
x25519_frodo640shake @ oqsprovider
frodo976aes @ oqsprovider
p384_frodo976aes @ oqsprovider
p256_frodo640aes @ oqsprovider
frodo976shake @ oqsprovider
p384_frodo976shake @ oqsprovider
x448_frodo976shake @ oqsprovider
frodo1344aes @ oqsprovider
p521_frodo1344aes @ oqsprovider
frodo640aes @ oqsprovider
p521_frodo1344shake @ oqsprovider
mlkem512 @ oqsprovider
p256_mlkem512 @ oqsprovider
x25519_mlkem512 @ oqsprovider
mlkem768 @ oqsprovider
p384_mlkem768 @ oqsprovider
x448_mlkem768 @ oqsprovider
X25519MLKEM768 @ oqsprovider
SecP256r1MLKEM768 @ oqsprovider
x25519_frodo640aes @ oqsprovider
p256_frodo640shake @ oqsprovider
x448_frodo976aes @ oqsprovider
frodo1344shake @ oqsprovider
mlkem1024 @ oqsprovider
p256_bikel1 @ oqsprovider
bikel3 @ oqsprovider
p384_bikel3 @ oqsprovider
x448_bikel3 @ oqsprovider
SecP384r1MLKEM1024 @ oqsprovider
p521_mlkem1024 @ oqsprovider
bikel1 @ oqsprovider
x25519_bikel1 @ oqsprovider
bikel5 @ oqsprovider
p521_bikel5 @ oqsprovider

```

Output 2 listado de algoritmos de encapsulación de clave..

Una prueba de ejecución para evaluar el rendimiento es el siguiente:

```

podman run -e TEST_TIME=5 -e KEM_ALG=kyber768 -e SIG_ALG=dilithium3 -it
docker.io/openquantumsafe/curl perftest.sh

```

Este comando realiza intercambios de claves TLS durante 5 segundos utilizando:

- Kyber768 como algoritmo de intercambio de claves (KEM — Key Encapsulation Mechanism)
- Dilithium3 como algoritmo de firma digital (SIG — Signature Algorithm)

El script perftest.sh mide cuántas conexiones TLS exitosas se pueden establecer en ese intervalo de tiempo, proporcionando datos útiles para comparar el rendimiento y eficiencia de distintas combinaciones criptográficas.

Se ejecutan varios ejemplos con diferentes pares de algoritmos para validar la correcta recogida de rendimientos.

```
anruki@IBM-PF31MENJ:~$ podman run -e TEST_TIME=5 -e KEM_ALG=kyber512 -e SIG_ALG=dilithium3 -it openquantumsafe/curl perftest.sh
-----
Certificate request self-signature ok
subject=CN=localhost
Running /opt/oqssa/bin/perftest.sh with SIG_ALG=dilithium3 and KEM_ALG=kyber512
Using default temp DH parameters
ACCEPT
s_time: verify depth is 1
5329 connections in 2.04s; 2612.25 connections/user sec, bytes read 0
5329 connections in 6 real seconds, 0 bytes read per connection
```

Output 3 Conexiones TLS con algoritmo KEM kyber512 y algoritmo de firma dilithium3

```
anruki@IBM-PF31MENJ:~$ podman run -e TEST_TIME=5 -e KEM_ALG=kyber1024 -e SIG_ALG=falcon512 -it openquantumsafe/curl perftest.sh
-----
Certificate request self-signature ok
subject=CN=localhost
Running /opt/oqssa/bin/perftest.sh with SIG_ALG=falcon512 and KEM_ALG=kyber1024
Using default temp DH parameters
ACCEPT
s_time: verify depth is 1
3895 connections in 2.27s; 1715.86 connections/user sec, bytes read 0
3895 connections in 6 real seconds, 0 bytes read per connection
```

Output 4 Conexiones TLS con algoritmo KEM kyber1024 y algoritmo de firma falcon512

Se recogen manualmente los resultados de las pruebas de validación anteriores:

	Prueba 1	Prueba 2
<i>KEM utilizado</i>	kyber512	kyber1024
<i>Algoritmo de firma (SIG)</i>	dilithium3	falcon512
<i>Tiempo de prueba (TEST_TIME)</i>	5 segundos	5 segundos
<i>Conexiones realizadas</i>	5329	3895
<i>Tiempo de CPU utilizado</i>	2.04 segundos	2.27 segundos
<i>Conexiones por segundo (CPU)</i>	2612.25	1715.86
<i>Tiempo real transcurrido</i>	6 segundos	6 segundos
<i>Datos transferidos por conexión</i>	0 bytes	0 bytes
<i>Entorno de contenedor</i>	Podman	Podman

Tabla 7. Métricas recogidas en conexiones TLS Quantum Safe. Elaboración propia.

Las pruebas duraron 6 segundos de tiempo real, incluyendo el arranque, overhead y finalización. El resultado de 0 bytes leídos por conexión es normal, ya que el objetivo principal de la prueba es medir el rendimiento del establecimiento de conexiones TLS, enfocado en la negociación y autenticación (criptografía asimétrica), y no en la transferencia de datos, que normalmente utiliza criptografía simétrica.

El sistema logró manejar más de 5000 conexiones en un entorno controlado usando algoritmos postcuánticos. Esto demuestra que el proyecto es viable y los algoritmos postcuánticos como Kyber, Dilithium y Falcon pueden ser ejecutados en operaciones de establecimiento de sesión TLS entre contenedores de LibOQS.

4.2.5. Exportación de rendimientos

Para medir y exportar el rendimiento de todos los algoritmos criptográficos disponibles se ha utilizado openssl speed dentro del contenedor:

```
podman run -it docker.io/openquantumsafe/curl openssl speed
```

Este comando ejecuta una serie de pruebas de rendimiento que incluyen tanto algoritmos clásicos, postcuánticos e híbridos. No obstante, de todas las ejecuciones, este proyecto se centra en las métricas obtenidas para los protocolos de criptografía asimétrica, que incluyen: Intercambio de Clave, Firma Digital y Encapsulación de Clave. Cada prueba proporciona métricas en base al tipo de algoritmo ejecutado:

Categoría	Métrica	Descripción
<i>Firma Digital</i>	Keygen / Keygens/s	Generación de claves para firma y número de claves generadas por segundo.
	Sign / Signs/s	Tiempo y frecuencia para firmar mensajes digitalmente.
	Verify / Verifys/s	Tiempo y frecuencia para verificar firmas digitales.
<i>Intercambio de Clave</i>	Time per op	Tiempo medio necesario para completar una operación de derivación de clave.
	Ops per second	Número de operaciones de intercambio de clave realizadas por segundo.
<i>Encapsulación de Claves (KEM)</i>	Keygen / Keygens/s	Generación de claves para KEM y número de operaciones por segundo.
	Encaps / Encaps/s	Tiempo y frecuencia para encapsular (cifrar) una clave compartida.
	Decaps / Decaps/s	Tiempo y frecuencia para recuperar (descifrar) la clave encapsulada.

Tabla 8. Métricas disponibles por categoría del algoritmo. Elaboración propia.

4.2.5. Clasificación y representación de resultados

Los resultados obtenidos en las pruebas de rendimiento criptográfico (descritas en el apartado anterior) y los datos procesados se encuentran en anexos. Partiendo de un output de texto con distintos apartados y métricas sin normalizar se ha seguido el siguiente método para el procesamiento y análisis:

```

fetch.c:355:Global default library context, Algorithm (DSA-SHA1 : 109),
Properties: (null)
version: 3.4.0
built: Tue Jun 17 14:47:38 2025 UTC
options: bn(64,64)
compiler: gcc -fPIC -pthread -m64 -Wa,-noexecstack -Wall -O3 -
DOPENSSL_USE_NODELETE -DOPENSSL_THREADS -DOPENSSL_PIC
CPUINFO: OPENSSL_1a3
The 'numbers' are in
type      16
bytes    16384 bytes
bits     16384 bits
kbytes   8192 kbytes
871398.66k 879132.57k 1151558.54k 1175612.39k 351794.52k 352051.45k 532534.25k 532163.24k 34164.48k 769548.41k 31916.11k 519418.02k 536496.48k 352326.75k 36634.70k 36654.85k 36298.94k
35974.04k 36462.87k

```

Log .txt de resultados openSSL

Tablas excel filtradas por función criptográfica



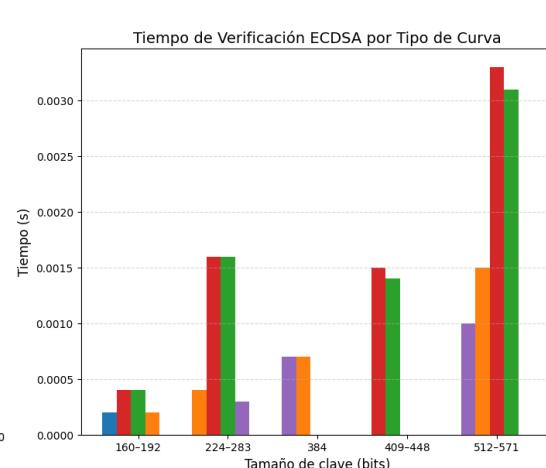
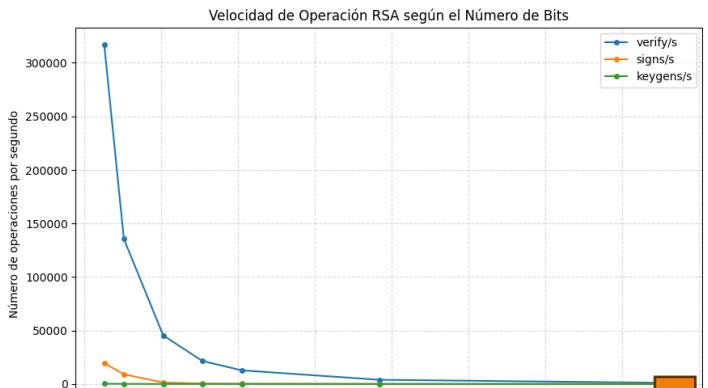
DATOS PROCESADOS: limpieza de errores tipográficos, normalización de las unidades, estandarización de las métricas y extracción de nuevas columnas como Bits, Familia y Tipo de algoritmo.

Familia	Tipo	bits	keygen	encaps	decaps	keygens/n	encaps/n	decaps/n	Algoritmo
ECC	Tradicional	253	0.000081	0.000023	0.000044	22768.4	84277.7	12200.3	X25319
ECC	Tradicional	255	0.000012	0.000081	0.000065	84277.7	12200.3	15439	ECP-256
ECC	Tradicional	384	0.000784	0.001577	0.000819	1275.6	884.2	1220.3	ECP-384
ECC	Tradicional	448	0.001194	0.000357	0.000159	5147.5	2803.6	6290.6	X448
ECC	Tradicional	512	0.001892	0.003889	0.000359	528.7	257.1	510.4	ECP-512
FrodoKEM	Post-Cuántico	640	0.000293	0.000383	0.000374	3414.2	2811.7	2673.4	frodo640aes
FrodoKEM	Post-Cuántico	640	0.000994	0.001072	0.001086	1005.8	932.9	920.8	frodo640shake
FrodoKEM	Híbrido	640	0.000772	0.000464	0.000588	1296.1	2154.9	1700.4	p256_frodo40aes
FrodoKEM	Híbrido	640	0.001546	0.001133	0.001331	646.7	882.3	763.1	p256_frodo40shake
FrodoKEM	Híbrido	640	0.000322	0.000456	0.000427	3103.9	2193.5	2344.4	x25519_frodo40shake
FrodoKEM	Híbrido	640	0.000998	0.001134	0.001162	1002.2	882.1	860.4	x25519_frodo640shake
FrodoKEM	Post-Cuántico	976	0.000593	0.000737	0.0007	1686.9	1357.4	1427.9	frodo976aes
FrodoKEM	Post-Cuántico	976	0.002153	0.002228	0.002285	484.5	449.3	441.6	frodo976shake
FrodoKEM	Híbrido	976	0.001999	0.002466	0.001656	500.3	405.5	601.8	p384_frodo76aes
FrodoKEM	Híbrido	976	0.003414	0.003888	0.003251	292.9	257.8	307.6	p384_frodo76shake
FrodoKEM	Híbrido	976	0.002661	0.002675	0.002531	440.2	951.9	980	x448_frodo76aes
FrodoKEM	Híbrido	1344	0.000109	0.001024	0.00123	991.2	795	822.8	frodo134aes
FrodoKEM	Post-Cuántico	1344	0.003827	0.003978	0.004121	261.3	251.4	242.6	frodo134shake
FrodoKEM	Híbrido	1344	0.003675	0.005017	0.003273	272.1	199.3	305.5	p521_frodo134aes
FrodoKEM	Híbrido	1344	0.006166	0.006066	0.006157	162.2	123.9	162.4	p521_frodo134shake

Familia	bits total	sign (s)	verify (s)	keygens/s	sign/s	verify/s	Tiempo total (s)	Tipo	Grupo bits	Algoritmo
DSA	1024	0.000084	0.000107	4415.4	9435.3	9442.3	-	-	1024	dsa1024
DSA	2048	0.000048	0.000116	1459	2183.3	3157.8	0.000774	-	2048	dsa2048
ECCDSA	256	0.0003	0.0003	2944.4	3314.3	0.0006	Curve512M2	256	Curve512	
ECCDSA	253	0	0.0001	24899.5	8974.2	0.0001	Ed25519	253	Ed25519	
ECCDSA	456	0.0002	0.0002	4899.4	4790.4	0.0004	Ed448	456	Ed448	
ECCDSA	256	0.0003	0.0003	3094.6	2995.0	0.0006	brainpool1P256r1	256	brainpool1P256r1	
ECCDSA	384	0.0008	0.0007	1203.1	1403.2	0.0015	brainpool1P384r1	384	brainpool1P384r1	
ECCDSA	512	0.0012	0.0001	850.2	1000.1	0.0022	brainpool1P512r1	512-571	brainpool1P512r1	
ECCDSA	163	0.0002	0.0004	5114.2	2247.7	0.0006	nistb163	160-192	nistb163	
ECCDSA	233	0.0003	0.0005	3411.6	1867.3	0.0006	nistb233	224-283	nistb233	
ECCDSA	281	0.0005	0.0011	2134.7	1892.2	0.0016	nistb281	224-283	nistb281	
ECCDSA	401	0.0008	0.0015	1180.3	656.1	0.0023	nistb409	409-448	nistb409	
ECCDSA	571	0.0018	0.0033	541.1	301.4	0.0031	nistb571	512-571	nistb571	
ECCDSA	16	0.0002	0.0003	5339.1	2835.6	0.0006	nistk16	160-192	nistk16	
ECCDSA	233	0.0003	0.0005	3901.5	2031.9	0.0008	nistk233	224-283	nistk233	
ECCDSA	283	0.0004	0.0011	2284.3	951.1	0.0015	nistk283	224-283	nistk283	
ECCDSA	469	0.0007	0.0014	1397.5	699.4	0.0021	nistk469	409-448	nistk469	
ECCDSA	571	0.0017	0.0031	589.9	322.2	0.0048	nistk571	512-571	nistk571	
ECCDSA	192	0.0002	0.0002	4519.5	5117.1	0.0004	nistp192	160-192	nistp192	
ECCDSA	224	0.0003	0.0003	3168	3505.3	0.0006	nistp224	224-283	nistp224	
ECCDSA	256	0	0.0001	48614.5	15834.1	0.0001	nistp256	224-283	nistp256	



VISUALIZACIÓN DE LOS DATOS PROCESADOS MEDIANTE GRÁFICAS



EXTRACCIÓN DE CONCLUSIONES

El proceso de limpieza y estandarización de los datos mediante scripts en Python cumple las siguientes funciones:

- Detección y análisis de outliers: por ejemplo, en algunos casos, los tiempos de ejecución son tan reducidos que en el registro aparecen como 0 segundos.
- Unificación de unidades métricas: se ha normalizado el formato de todas las métricas para que incluyan las unidades de forma coherente y estandarizada, lo que facilita su interpretación y comparación.
- Enriquecimiento de los datos: se han añadido columnas adicionales como el número de bits, la curva criptográfica (cuando aplica) y la familia del algoritmo, con el fin de contextualizar mejor cada dato.

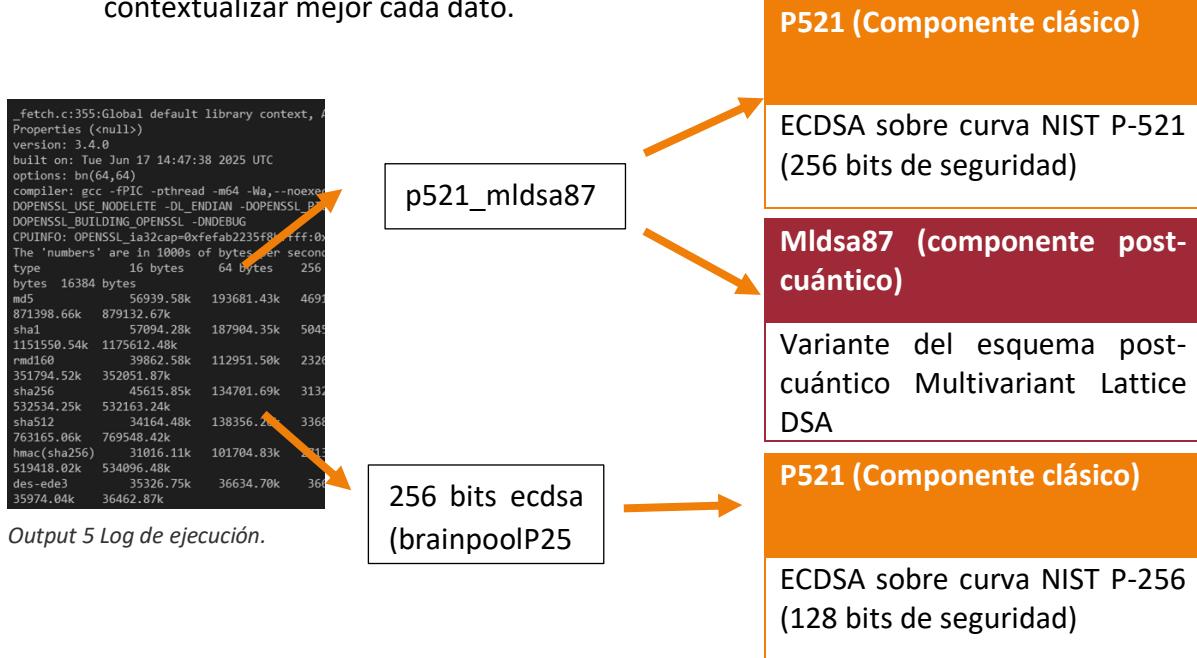


Tabla 9. Desglose de dos algoritmos ejecutados Elaboración propia.

Familia	Bits total	Tipo	Grupo bits	Algoritmo
ECDSA	256	brainpool	224–283	brainpoolP256r1

Tabla 10. Algoritmo post procesamiento. Elaboración propia.

4.3. PLAN DE TRABAJO

En este apartado se detalla el plan de trabajo general, alineado con la metodología anteriormente definida. Además, se desglosan los distintos paquetes de trabajo (PT) en los que se organiza el proyecto, especificando las actividades y objetivos asociados a cada uno. La hoja de ruta para el desarrollo del proyecto se encuentra en Anexos.

4.3.1. PT 1 Investigación previa

Se ha analizado la base matemática de los protocolos criptográficos tradicionales (RSA, DH), así como los algoritmos cuánticos que podrían comprometer su seguridad y las medidas disuasorias.

Algoritmo Cuántico	Funcionalidad	Tiempo de resolución ($n = \text{bits}$)	Protocolos Criptográficos	Mitigación
Shor	<i>factoring</i>	$\text{poly}(n)$	RSA	<i>Migrar a PQC</i>
Shor	<i>discrete logarithm</i>	$\text{poly}(n)$	DH, DSA, ECC	<i>Migrar a PQC</i>
Grover	<i>key search</i>	$2^{\frac{n}{2}}$	Symmetric key algorithms (e.g., AES)	Aumentar clave
Grover	<i>pre – image attack</i>	$2^{\frac{n}{2}}$	Hash functions (e.g., SHA – 256)	Aumentar hash
BHT	<i>collision attack</i>	$2^{\frac{n}{3}} \text{ or } 2^{\frac{2n}{5}}$	Hash functions (e.g., SHA – 256)	Aumentar hash

Tabla 11. Características de los algoritmos cuánticos y protocolos criptográficos a los que amenazan [64].

Los protocolos de criptografía asimétrica son los más expuestos al avance de la computación cuántica, ya que el algoritmo de Shor puede resolver en tiempo polinómico los problemas que sustentan su seguridad. En el caso de estos protocolos, el aumento del tamaño de clave no constituye una solución sostenible a largo plazo. Por ello, se hace necesaria una migración hacia esquemas criptográficos resistentes a la computación cuántica con la mayor urgencia posible.

Para centrar el análisis en un caso de uso concreto, se ha seleccionado el protocolo TLS debido a su amplia adopción en las comunicaciones por internet y soporte en OpenSSL. A continuación, se presenta un resumen del impacto potencial de los algoritmos cuánticos sobre las principales funciones criptográficas empleadas en TLS 1.3:

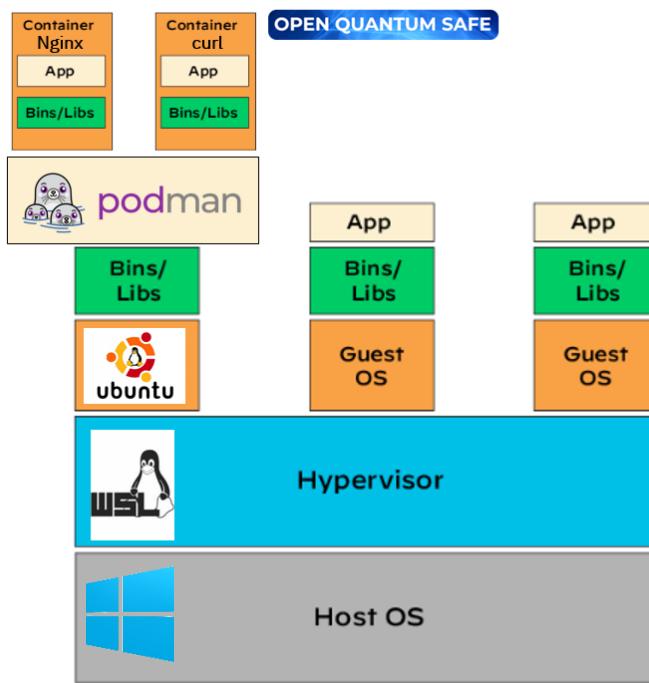
Función criptográfica	Afectación por ataques	Frecuencia de uso en TLS v1.3	Algoritmo cuántico que amenaza	Criticidad
Cifrado simétrico	Baja	Muy alta	Grover	Baja
Hash	Moderada	Muy alta	Grover	Media
Cifrado asimétrico	Muy alta	Baja	Shor	Alta
Firma Digital	Muy alta	Muy alta	Shor	<i>Muy alta</i>
Intercambio de claves	Muy alta	Muy alta	Shor	<i>Muy alta</i>

Tabla 12. Funciones criptográficas en TLS. Elaboración propia.

Tras esta investigación previa, se concluye que el caso de estudio del proyecto es la migración de la criptografía asimétrica de intercambio de claves en el protocolo TLS. Siendo una de las aplicaciones críticas de la empresa con mayor necesidad de migración.

4.3.2. PT 2 Implementación de Arquitectura basada en Linux

Se despliegan todas las tecnologías necesarias para llevar a cabo simulaciones de conexiones TLS, tanto con algoritmos actuales como con algoritmos postcuánticos. Se sigue la metodología de infraestructura basada en Linux y adaptación de contenedores a Podman.



Para el despliegue del entorno, se ha utilizado Podman como motor de contenedores sobre la distribución Ubuntu 22.04, ejecutada a través de WSL 2 en Windows. Esta configuración aprovecha la eficiencia de WSL 2 para ejecutar un entorno Linux real, junto con las ventajas de Podman, que permite gestionar contenedores sin requerir un daemon centralizado.

Finalmente, en este entorno se han descargado los contenedores de nginx y curl provistos por el proyecto OpenQuantumSafe.

Figura 25. Diseño de la infraestructura. Elaboración propia.

Esta arquitectura ha facilitado el trabajo con tecnologías avanzadas de criptografía postcuántica en un entorno accesible y optimizado.

4.3.3. PT 3 Conexiones de prueba y recogida de rendimientos

Se ha verificado la correcta descarga de los contenedores y el funcionamiento del protocolo TLS mediante conexiones de prueba realizadas desde la línea de comandos con curl, accediendo al servidor nginx. Además, se recopilaron algunas métricas de rendimiento utilizando el script perftest.sh.

	Prueba 1	Prueba 2
KEM utilizado	kyber512	kyber1024
Algoritmo de firma (SIG)	dilithium3	falcon512
Tiempo de prueba (TEST_TIME)	5 segundos	5 segundos
Conexiones realizadas	5329	3895
Tiempo de CPU utilizado	2.04 segundos	2.27 segundos

<i>Conexiones por segundo (CPU)</i>	2612.25	1715.86
<i>Tiempo real transcurrido</i>	6 segundos	6 segundos

Tabla 13. Métricas recogidas en conexiones TLS Quantum Safe. Elaboración propia.

Una vez validado el correcto funcionamiento de la infraestructura, se empleó el comando de OpenSSL speed para ejecutar todos los algoritmos disponibles en el entorno de pruebas. La ejecución completa tuvo una duración aproximada de 2 horas.

4.3.4. PT 4 Procesamiento de los datos

Con el fin de procesar adecuadamente los datos obtenidos, se realizó un estudio de los algoritmos postcuánticos ejecutados, así como de sus distintas variantes.

En el marco de este paquete de trabajo, se ha llevado a cabo una clasificación de dichos algoritmos mediante tablas, lo que ha facilitado su comprensión y filtrado sobre los resultados obtenidos [Ver Anexo C]. Posteriormente se han utilizado scripts de Python y finalmente se han obtenido 2 tablas con el formato necesario para su correcta visualización [Ver Anexo E].

4.3.5. PT 5 Representación de resultados y conclusiones

Se han realizado diversas gráficas que ayudan a comprender el rendimiento de los algoritmos y se ha comprobado que los nuevos algoritmos postcuánticos e incluso sus variantes híbridas con curvas elípticas son más eficientes que algoritmos tradicionales como RSA y FFDH. También ha sido objeto de estudio la afectación de cada tipo de curva elíptica en el tiempo total de los algoritmos híbridos.

4.4. CONDICIONANTES Y LIMITACIONES

Durante el desarrollo del proyecto se presentaron diversos condicionantes y limitaciones que influyeron tanto en el proceso como en los resultados obtenidos.

Uno de los factores más relevantes durante proyectos de este tipo es la constante evolución del entorno tecnológico. En particular, los estándares de criptografía postcuántica continúan desarrollándose y actualizándose, por lo que es fundamental mantenerse al día con los cambios para asegurar que las implementaciones sean seguras y compatibles con las futuras normativas y recomendaciones internacionales.

Desde el punto de vista computacional, se presentaron limitaciones relacionadas con el espacio de almacenamiento. El equipo portátil inicialmente utilizado agotó su capacidad. Esto se debe a que el proyecto involucraba la evaluación de algoritmos criptográficos tradicionales, híbridos y postcuánticos, además de la infraestructura necesaria para simular un cliente y un servidor.

Sin embargo, esta situación ha impulsado la optimización del uso de recursos. Por ejemplo, se instaló la interfaz de línea de comandos (CLI) de Podman desde los repositorios oficiales de Ubuntu, dado que su tamaño es considerablemente reducido en contraste con Podman Desktop con un tamaño de la instalación de más de 800 MB en entornos Windows.

A continuación, se detallan los principales componentes instalados y sus respectivos requerimientos de almacenamiento:

Elemento	Tamaño en MB
Python 3.11	1300 MB
Ubuntu	131000 MB
Visual Studio Code	416 MB
Podman	128 MB
Archivos generados	31.2 MB
Contenedor: curl	66.8 MB
Contenedor: nginx	96.1 MB
TOTAL: 133.04 GB	

Tabla 14. Requerimientos de espacio computacional del proyecto. Elaboración propia.

```
anruki@MSI:~$ podman images
REPOSITORY                      TAG      IMAGE ID      CREATED     SIZE
docker.io/openquantumsafe/curl   latest   904bc9ddc1e9  9 days ago  66.8 MB
docker.io/openquantumsafe/nginx  latest   276dd4202795  9 days ago  96.1 MB
```

Output 6 Espacio de almacenamiento de los contenedores (en MegaBytes)

El espacio total aproximado ocupado durante el proyecto fue de 133,04 GB, lo que representa una carga significativa para equipos de gama media. Por ejemplo, en un ordenador ThinkPad con procesador i5 y una capacidad de almacenamiento de 235 GB, este volumen supone más del 50 % del espacio disponible. No obstante, es importante destacar que este nivel de demanda corresponde a un entorno de pruebas exhaustivo; un usuario final que aplique criptografía postcuántica en su sistema con un solo sistema operativo no debería experimentar un impacto de almacenamiento tan significativo.

Realizar las pruebas en un equipo estándar ha asegurado que personas con recursos limitados también puedan replicar el experimento.

5. RESULTADOS

En este apartado se muestran los resultados más significativos de los algoritmos con mayor proyección, el código junto con todos los resultados se encuentra en anexos.

Los experimentos se realizaron en un equipo con procesador Intel® Core™ i5-1135G7, 16 GB de RAM, bajo el sistema operativo Ubuntu 22.04. Las implementaciones criptográficas fueron desarrolladas utilizando las bibliotecas OpenSSL, liboqs y sus herramientas de medición de tiempo.

PROCESADOR	Intel® Core™ i5-1135G7 (11ª generación) @ 2.40 GHz (8 CPUs, ~2.6 GHz)
MEMORIA RAM	16 GB
SISTEMA OPERATIVO	Ubuntu 22.04 LTS
CÓDIGO FUENTE	Contenedores LibOQS descargados de Docker Hub

Tabla 15. Elementos del entorno experimental.

Para el análisis de resultados, es fundamental considerar que, en los esquemas criptográficos, a mayor tamaño de clave, mayor es la seguridad, pero también mayor el coste computacional asociado a las operaciones. Por ello, existe un equilibrio entre rendimiento y nivel de protección que debe ser evaluado según el contexto de uso.

En este sentido, el concepto de bit de seguridad (security strength) mide la dificultad de romper un sistema criptográfico. Por ejemplo: 128 bits de seguridad significa que romper el sistema requeriría 2^{128} operaciones [65].

Bits de seguridad	Recomendación NIST SP 800-131A Rev.2 (2019)
< 112 bits	No permitido
112 bits	Aceptable Uso mínimo aceptable hasta ~2030
128 bits	Recomendado Seguridad de largo plazo
192-256 bits	Alta seguridad Recomendado para sistemas de alta seguridad o post-cuánticos

Tabla 16. Niveles de seguridad NIST [65]

Estas recomendaciones permiten comparar algoritmos con distintos tamaños de clave y asegurar que cumplen con los estándares mínimos de seguridad exigidos por organismos internacionales.

5.1. ALGORITMO DE INTERCAMBIO DE CLAVE DIFFIE-HELLMAN (DH)

Los algoritmos de intercambio de claves permiten que dos partes generen una clave secreta compartida para cifrar comunicaciones, sin necesidad de haberla acordado previamente, incluso si se comunican por un canal inseguro. Entre los métodos más destacados para lograr esto se encuentra el algoritmo de Diffie-Hellman (DH), que fue uno de los primeros protocolos prácticos para el intercambio seguro de claves sin necesidad de un canal previo confiable [66].

5.1.1. Finite Field Diffie-Hellman (FFDH)

FFDH es una variante clásica del algoritmo de intercambio de claves Diffie-Hellman, basada en exponenciación modular en campos finitos. Se utiliza en entornos donde se requiere el establecimiento de claves simétricas seguras a través de canales inseguros [68].

El tamaño de la clave (2048, 3072, etc.) se refiere al número de bits del módulo primo utilizado en la operación y presentan los siguientes bits de seguridad:

Tamaño clave FFDH (bits)	Bits de seguridad aproximados
2048	112
3072	128
4096	152
6144	176
8192	200

Tabla 17. Bits de seguridad Finite Fields Diffie Hellman [67][68].

Aunque FFDH ha sido ampliamente usado en TLS tradicional, ha sido progresivamente reemplazado por ECDH y algoritmos postcuánticos, debido a su menor eficiencia computacional y mayores requerimientos de tamaño de clave para niveles equivalentes de seguridad [69].

Nombre algoritmo	Time per op (s)	Ops per second	Bits
Finite Fields Diffie Hellman	0.0003	3715.8	2048
Finite Fields Diffie Hellman	0.0007	1491.7	3072
Finite Fields Diffie Hellman	0.0014	729.3	4096
Finite Fields Diffie Hellman	0.0034	297.4	6144
Finite Fields Diffie Hellman	0.0063	157.7	8192

Tabla 18. Rendimiento Finite Fields Diffie Hellman. Elaboración propia.

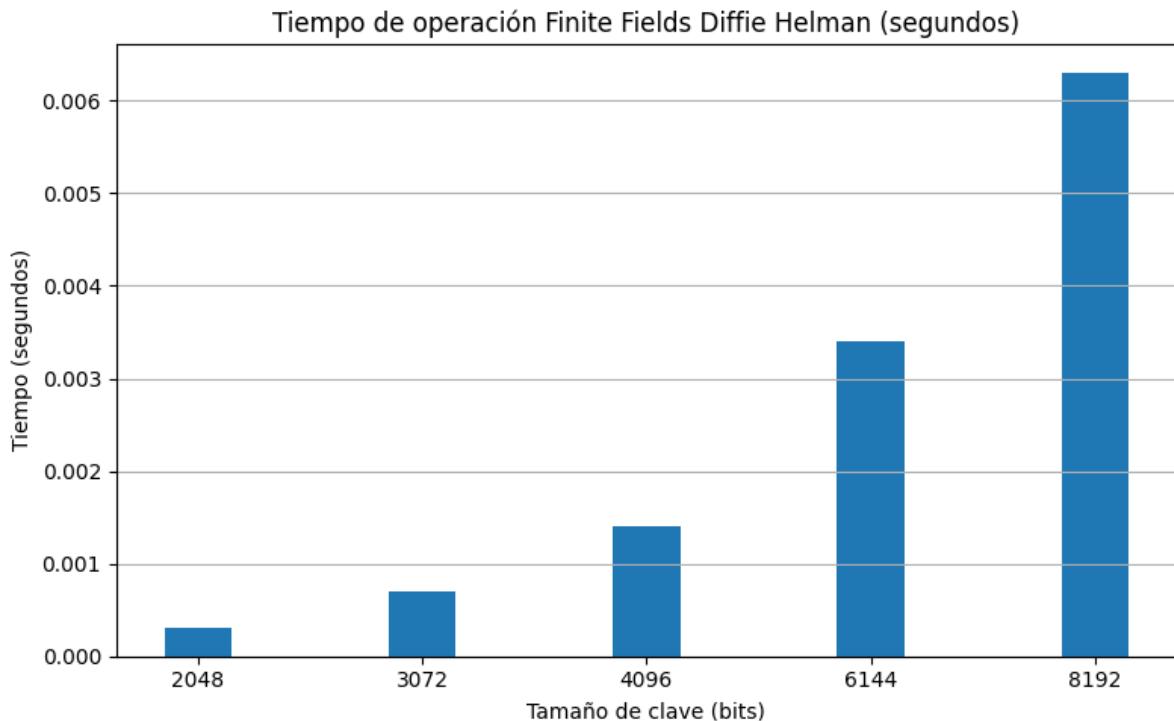


Figura 26. Tiempos de ejecución de FFDH para distintos tamaños de clave. Elaboración propia

DHFF con tamaño de clave 2048 bits es el mínimo aceptable por NIST, ya que aporta 112 bits de seguridad. No obstante, a partir de 2030 se recomienda aumentar el tamaño de clave a 3072. Esto supondría más del doble de tiempo de ejecución (para casos como el entorno de pruebas utilizado).

En conclusión, las empresas que utilizan DHFF con 2048 bits deberán estudiar si cambiar el tamaño de clave en el futuro o utilizar otros algoritmos que aporten mayor número de bits de seguridad para mantenerse alineados con el NIST.

5.1.2. Elliptic Curve Diffie-Hellman (ECDH)

El algoritmo ECDH (Elliptic Curve Diffie-Hellman) se utiliza principalmente para generar una clave compartida segura entre dos partes, en TLS [69]. Además, existen distintas curvas elípticas y variantes que hacen necesario su análisis. Las curvas se diferencian por el tamaño de clave (en bits) y la familia o estándar que las define (NIST, Brainpool, X25519/X448).

Las curvas NIST (National Institute of Standards and Technology) son un conjunto de curvas elípticas estandarizadas para criptografía, ampliamente utilizadas debido a su respaldo oficial por el NIST. Se clasifican en tres familias principales según el tipo de campo sobre el que están definidas:

- nistp: Curvas sobre campos primos
- nistk: Curvas Koblitz sobre campos binarios, optimizadas para eficiencia.
- nistb: Curvas binarias no-Koblitz, también sobre campos binarios.

Con sus respectivos bits de seguridad:

Curve Name	Estimated Security Strength
nistp256	128 bits
nistp384	192 bits
nistp521	512 bits

Tabla 19. Niveles de seguridad de curvas elípticas NIST [66]

Las curvas X25519 y X448 fueron desarrolladas por Daniel J. Bernstein junto con colaboradores como Tanja Lange y otros expertos en criptografía [70]. Estas curvas son actualmente las más recomendadas para nuevas implementaciones criptográficas y se utilizan en las versiones más recientes de TLS [71]. Sus niveles de seguridad son:

Curve Name	Estimated Security Strength
X25519	128 bits
X448	224 bits

Tabla 20. Niveles de seguridad de curvas elípticas del proyecto Curve25519 [66]

Las curvas Brainpool fueron creadas por el grupo de trabajo alemán ECC Brainpool. Aunque no tienen el mismo nivel de adopción global que X25519 o las NIST P-curves, han sido implementadas en sectores gubernamentales alemanes para securizar datos personales como pasaportes [72].

Concretamente, en este proyecto se ha ejecutado la variante brainpool_r1 (random) como representante de la familia Brainpool, por ser la curva estándar y más ampliamente utilizada en la práctica [72][73]. Esta curva tiene los siguientes niveles de seguridad:

Curve Name	Estimated Security Strength
brainpoolP160	80 bits
brainpoolP192	96 bits
brainpoolP224	112 bits
brainpoolP256	128 bits
brainpoolP320	160 bits
brainpoolP384	192 bits
brainpoolP512	256 bits

Tabla 21. Niveles de seguridad de curvas elípticas Brainpool [67].

Tras ejecutar los algoritmos anteriores de curvas elípticas y procesar los datos resultantes, se han obtenido las siguientes métricas:

Algoritmo	Operation Time (s)	Ops per second	Bits	Curva

X25519 ³	0.0000	31223.3	253	X25519
X448	0.0002	6331.0	448	X448
brainpoolP256r1	0.0003	3171.5	256	brainpool
brainpoolP384r1	0.0008	1227.8	384	brainpool
brainpoolP512r1	0.0012	843.0	512	brainpool
nistb163	0.0002	5508.9	163	nistb
nistb233	0.0002	4160.6	233	nistb
nistb283	0.0004	2313.6	283	nistb
nistb409	0.0007	1430.5	409	nistb
nistb571	0.0016	640.9	571	nistb
nistk163	0.0002	5751.2	163	nistk
nistk233	0.0002	4341.1	233	nistk
nistk283	0.0004	2394.7	283	nistk
nistk409	0.0007	1478.4	409	nistk
nistk571	0.0015	687.5	571	nistk
nistp192	0.0002	5176.2	192	nistp
nistp224	0.0003	3537.7	224	nistp
nistp256	0.0000	21321.0	256	nistp
nistp384	0.0008	1278.4	384	nistp
nistp521	0.0018	541.9	521	nistp
secp160r1	0.0002	6536.1	160	secp

Tabla 22. Rendimientos ECDH. Elaboración propia.

Gracias a esta organización, se ha podido graficar el tiempo de operación de cada curva en base al número de bits. A continuación, se muestra la gráfica resultante:

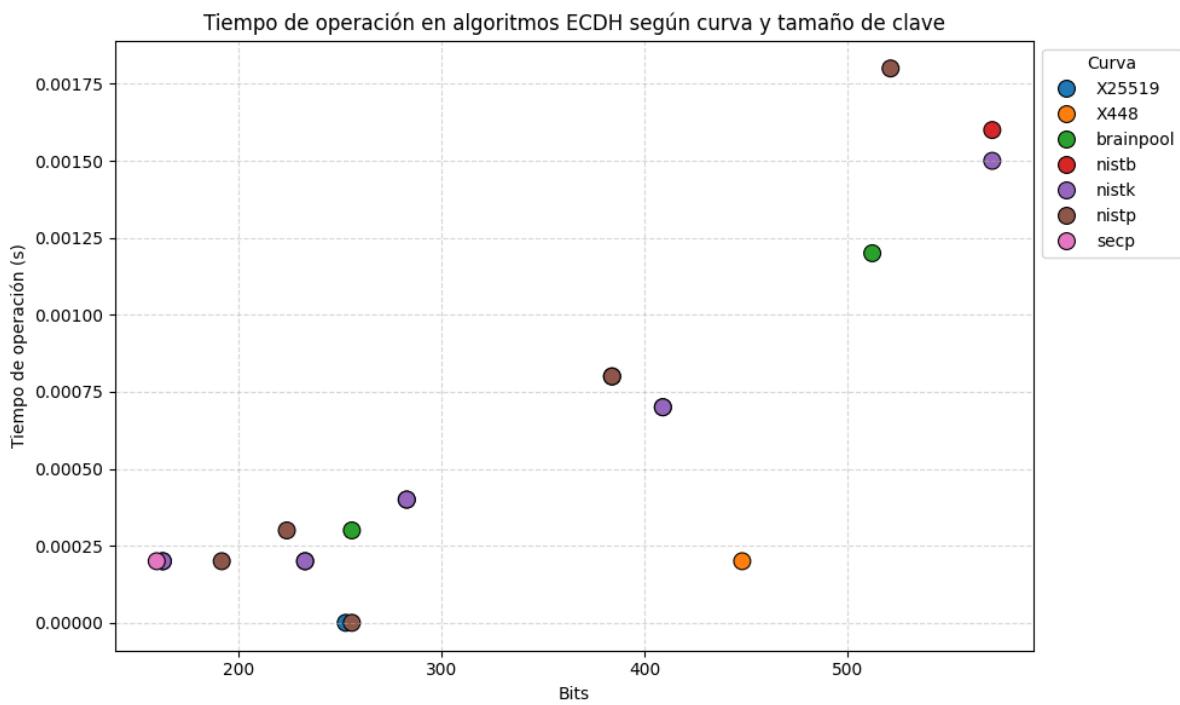


Figura 27. Tiempo de operación ECDH. Elaboración propia.

En términos de rendimiento de los algoritmos en el entorno de pruebas, las curvas nistp y X25519 fueron las que obtuvieron menores tiempos de ejecución. Por este motivo, sería

³ El algoritmo de curva elíptica X25519 tiene tiempo de ejecución casi 0 para 253 bits.

beneficioso implementarlas respecto a otras curvas con mayor costo computacional y menores bits de seguridad.

Por otra parte, también reportó tiempos de ejecución bajos la curva **X448**, sobre todo con respecto a otras curvas de similar tamaño de clave. La curva X448 está incluida en las suites de cifrado recomendadas por el RFC 8446 para TLS 1.3 [71] y en este experimento se confirma su buen desempeño.

5.1.3. Comparativa

Después de analizar los resultados de forma individual, se compararon los algoritmos de intercambio de clave ECDH y FFDH en función de los niveles de seguridad que ofrecen, medidos en bits de seguridad.

La figura siguiente presenta los tiempos de operación de los algoritmos Diffie-Hellman en su versión clásica sobre grupos finitos (FFDH) y en su variante basada en curvas elípticas (ECDH). Los intervalos de bits de seguridad se muestran mediante bandas de color, siguiendo las recomendaciones del NIST descritas previamente.

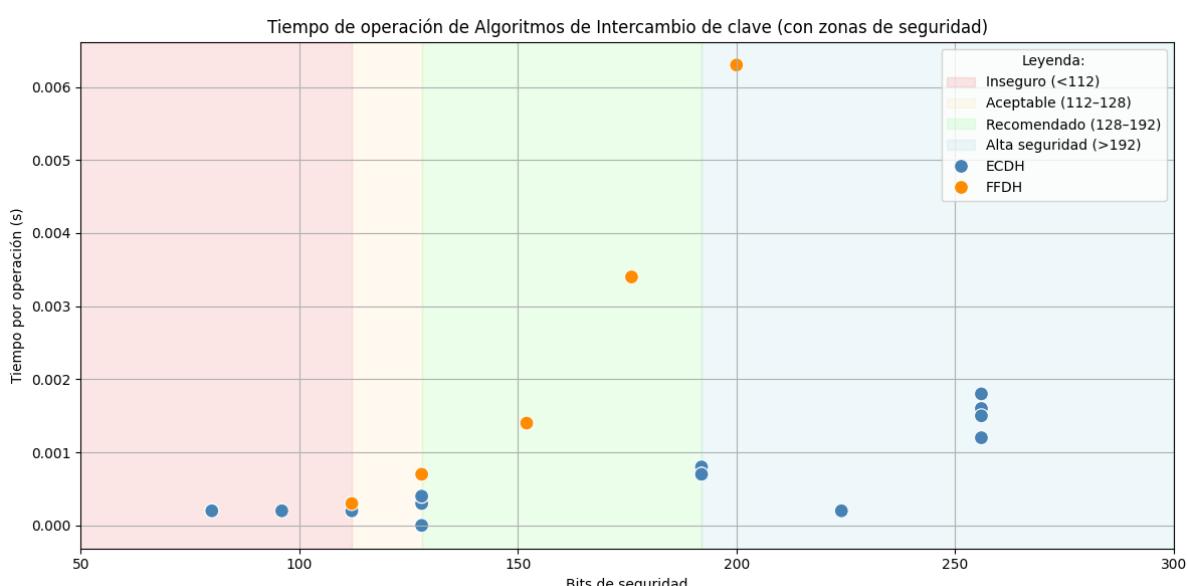


Figura 28. Comparativa tiempos de operación de Algoritmos de Intercambio de clave. Elaboración propia.

Se observa que ECDH mantiene tiempos significativamente más bajos y para mayor número de bits de seguridad. Por otra parte, FFDH necesita claves mucho más largas (miles de bits) para lograr niveles de seguridad comparables a los de ECDH, lo que lo hace menos eficiente.

Dado que los algoritmos clásicos como Diffie-Hellman son vulnerables a ataques cuánticos, la migración a criptografía postcuántica debe buscar un equilibrio entre seguridad y eficiencia operativa. En este sentido, las curvas elípticas, por su alta eficiencia y bajo costo computacional, se perfilan como buenas candidatas para formar parte de soluciones híbridas que combinen la seguridad clásica y postcuántica.

5.2. ALGORITMOS CLÁSICOS DE ENCAPSULACIÓN DE CLAVE (KEM)

Varias organizaciones y proyectos internacionales promueven activamente la adopción de KEMs postcuánticos, especialmente en el contexto de la migración hacia algoritmos resistentes a ataques cuánticos. El proyecto de código abierto Open Quantum Safe, que implementa y evalúa algoritmos PQC en bibliotecas como OpenSSL y liboqs, apoya el uso de KEMs como Kyber para reemplazar ECDH en TLS y otros protocolos. Asimismo, la IETF trabaja activamente en la integración de KEMs híbridos en protocolos como TLS 1.3 [74].

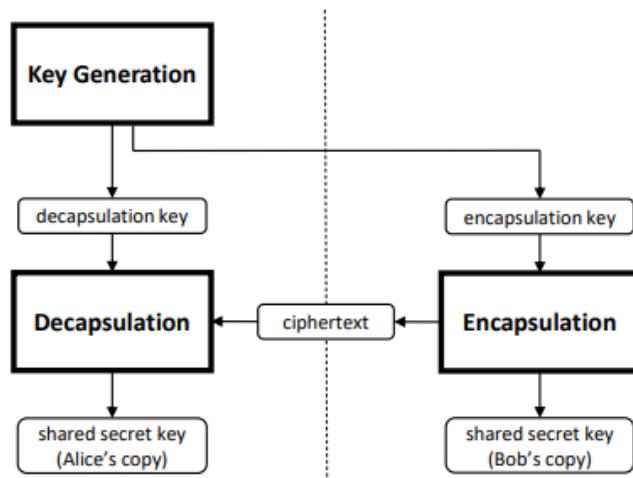


Figura 29 Key Encapsulation Mechanism [68].

El proceso implica generar un par de claves (pública y privada), encapsular una clave secreta usando la clave pública, y luego desencapsularla utilizando la clave privada para recuperar la clave compartida. A diferencia del intercambio de claves tradicional, en KEM, la clave se genera y envía de forma cifrada (encapsulada), en vez de derivarse conjuntamente como en Diffie-Hellman. Esto es muy útil para construir sistemas híbridos y postcuánticos, y para integrarse fácilmente en protocolos como TLS.

5.1.1. Rivest Shamir Adleman (RSA)

RSA puede utilizarse como mecanismo de encapsulación de claves (KEM), encapsulando una clave secreta mediante operaciones RSA tradicionales. Sin embargo, la seguridad de RSA KEM depende fuertemente del tamaño de la clave. Según estimaciones del NIST la equivalencia aproximada entre tamaño de clave RSA y bits de seguridad es la siguiente:

Tamaño clave RSA (bits)	Bits de seguridad aproximados
1024	80 bits
2048	112 bits
3072	128 bits
4096	140 bits

7680	192 bits
15360	256 bits

Tabla 23 Bits de seguridad en RSA [75].

Actualmente, claves RSA menores a 2048 bits se consideran inseguras para aplicaciones modernas. De hecho, en el año 2024, Microsoft ha anunciado la próxima descontinuación del soporte para claves RSA de 1024 bits en TLS en sus plataformas, buscando mejorar la seguridad ante amenazas actuales [76].

Por lo tanto, para que RSA KEM sea considerado seguro hoy en día, es recomendable utilizar claves de al menos 2048 bits. No obstante, a medio plazo habría que aumentar el tamaño de clave a 3072 bits para conseguir 128 bits de seguridad recomendados por el NIST. Y a largo plazo, el esquema podría volverse completamente inseguro debido al avance de los ordenadores cuánticos.

En el ámbito del proyecto se han obtenido métricas de ejecución que han sido procesadas para generar la siguiente tabla:

Algoritmo	keygens/s	encaps/s	decaps/s	keygen	encaps	decaps
rsa512	357.1	282509.2	29611.3	0.002800	0.000004	0.000034
rsa1024	118.8	147800.3	11230.6	0.008418	0.000007	0.000089
rsa2048	23.7	51410.3	3718.2	0.042263	0.000019	0.000269
rsa3072	7.0	25905.6	1330.7	0.142162	0.000039	0.000752
rsa4096	1.9	15991.2	620.8	0.521000	0.000063	0.001611
rsa7680	0.1	4831.1	28.0	8.000000	0.000207	0.035679
rsa15360	0.0	1244.1	5.6	21.810000	0.000804	0.177759

Tabla 24. Rendimientos RSA. Elaboración propia.

En la gráfica siguiente, se representa la velocidad de cada operación para cada tamaño de clave de RSA.

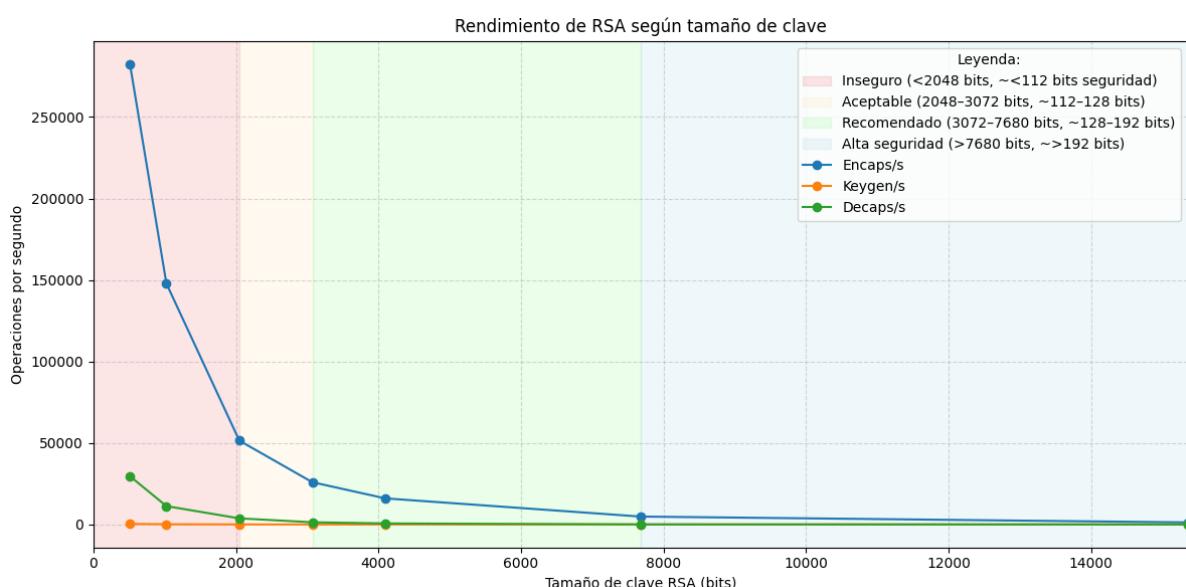


Figura 30. Rendimiento medido en operaciones por segundo de RSA. Elaboración propia.

Como se puede observar en los resultados experimentales, la velocidad de operación de RSA desciende drásticamente a medida que aumenta el número de bits en la clave. Siendo la generación de claves la operación más costosa.

Esto concuerda con estudios en criptografía que muestran cómo el coste computacional de RSA crece de forma exponencial con el tamaño de clave debido a las operaciones de factorización de enteros grandes, que sustentan su seguridad [77], [78]. Actualmente, la longitud mínima recomendada para claves RSA es de al menos 2048 bits para mantener un nivel aceptable de seguridad frente a los ataques modernos.

Sin embargo, este aumento de bits implica un coste computacional significativo. Por ello, RSA se considera ineficiente a largo plazo para aplicaciones que requieren alta velocidad y seguridad frente a la amenaza cuántica.

5.1.2. Elliptic Curves (EC-KEM)

EC-KEM utiliza conceptos derivados de ECDH para generar y encapsular una clave simétrica, pero en lugar de que ambas partes colaboren para derivar la clave, el emisor encapsula la clave usando la clave pública del receptor, y el receptor la recupera con su clave privada.

Como se observó anteriormente en Elliptic Curves Diffie Helman, el uso de curvas elípticas ofrece ventajas significativas respecto a otros algoritmos tradicionales. Las curvas elípticas requieren de un menor tamaño de clave para ofrecer un alto nivel de seguridad con un coste computacional reducido. Por ejemplo, en ECC, una clave de 256 bits proporciona un nivel de seguridad comparable al de una clave RSA de 3072 bits, gracias a la estructura matemática más compleja de las curvas elípticas [75]. A continuación, se muestran los bits de seguridad que ofrecen las curvas elípticas ejecutadas:

Algoritmo	Bits de seguridad
x25519	128 bits
ECP-256	128 bits
x448	224 bits
ECP-384	192 bits
ECP-521	256 bits

Tabla 25. Bits de seguridad de curvas elípticas KEM [67].

Algunas de las métricas obtenidas para los algoritmos de curvas elípticas KEM son:

Curva	keygens/s	encaps/s	decaps/s	keygen	encaps	decaps
x25519	29940.0	12768.4	28585.4	0.000033	0.000078	0.000035
ecp-256	84277.7	12400.3	15439.0	0.000012	0.000081	0.000065
ecp-384	1275.6	634.2	1220.3	0.000784	0.001577	0.000819
x448	5147.5	2803.6	6290.6	0.000194	0.000357	0.000159
ecp-521	528.7	257.1	510.4	0.001892	0.003889	0.001959

Tabla 26. Rendimiento de EC-KEM. Elaboración propia.

Utilizando estas métricas y los bits de seguridad de cada curva, se ha realizado la siguiente gráfica que muestra el tiempo total⁴ de ejecución (en escala logarítmica):

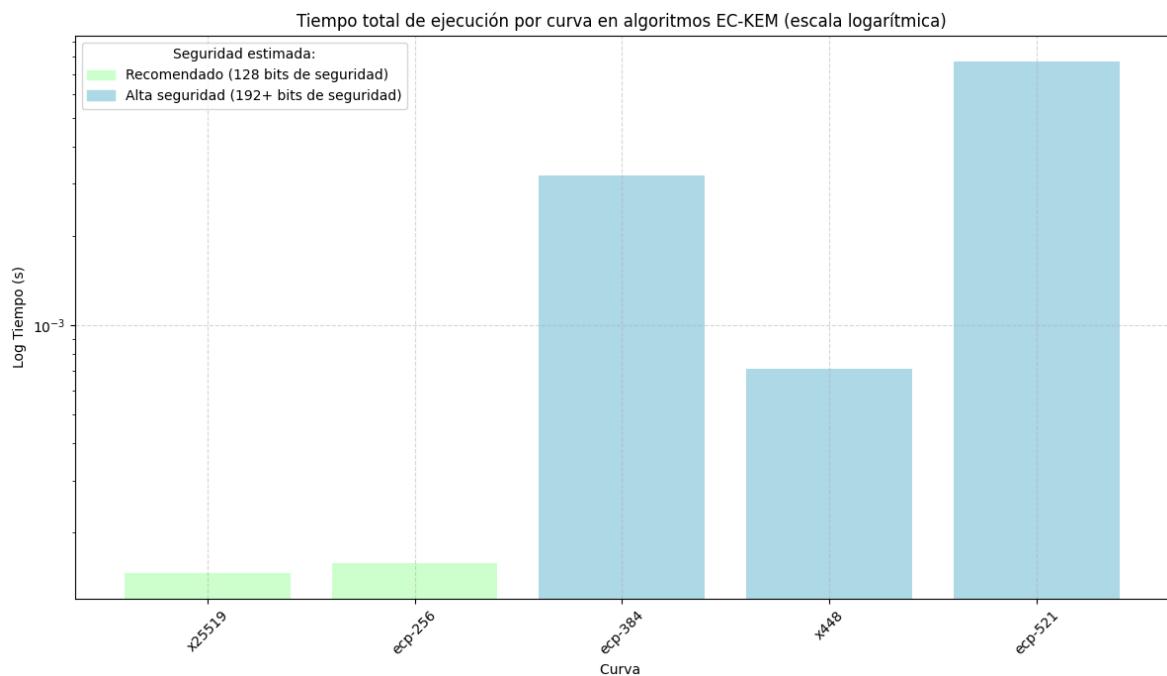


Figura 31. Tiempo total de ejecución en algoritmos EC-KEM. Elaboración propia.

La gráfica muestra que la curva X25519, una de las más utilizadas en la empresa, mostró el mejor rendimiento. Aunque la curva con mayor nivel de seguridad es ecp-521, su implementación conlleva un aumento significativo en el tiempo de ejecución, pasando de 0.000146 segundos (X25519) a 0.007740 segundos (ecp-521). Por otro lado, la curva X448 ofrece un equilibrio razonable entre seguridad y tiempo de ejecución.

No obstante, es importante tener en cuenta que tanto RSA como la criptografía de curvas elípticas, son vulnerables a ataques de computación cuántica mediante algoritmos como el de Shor. Esto significa que, a medio-largo plazo, la seguridad que ofrecen podría verse comprometida. Por tanto, para sistemas que requieran protección a largo plazo, es recomendable considerar la transición a algoritmos postcuánticos o híbridos que combinan seguridad clásica y cuántica.

Es fundamental evaluar cada caso particular considerando el flujo de conexiones y la naturaleza del negocio. Esto permitirá determinar si el incremento en tiempo de ejecución puede traducirse en pérdidas operativas o afectaciones en la experiencia del usuario. Además, el tipo de datos manejados debe guiar la prioridad en la selección de curvas, equilibrando la necesidad de mayor seguridad con los costos en rendimiento. En algunos escenarios, optar por curvas con mayor seguridad como ecp-521 puede ser justificado, mientras que, en otros, la eficiencia de X25519 o ecp-256 es preferible.

⁴ El tiempo total está compuesto por la suma de operaciones: generación de claves, encapsulación y desencapsulación.

5.3. ALGORITMOS POSTCUÁNTICOS E HÍBRIDOS DE ENCAPSULACIÓN DE CLAVES

Una de las principales aplicaciones actuales la criptografía de curvas elípticas en KEM es su integración en sistemas híbridos, donde se combinan algoritmos clásicos con algoritmos postcuánticos. De esta manera se da una transición controlada que garantiza la seguridad frente a adversarios tanto clásicos como cuánticos.

Los algoritmos postcuánticos, aún se encuentran en desarrollo, y no han tenido la suficiente exposición en el mercado para garantizar su fiabilidad. Por ello, se han creado algoritmos híbridos que utilizan la criptografía moderna y la postcuántica.

El uso de KEM híbridos ha sido especialmente impulsado por actores industriales como Cloudflare y Google, que han implementado versiones experimentales de TLS usando combinaciones como X25519 + Kyber-512, destacando su rendimiento y facilidad de integración [79].

5.1.3. Multivariant Lattice KEM / Kyber

Kyber, y su variante estandarizada ML-KEM (Modular Lattice Key Encapsulation Mechanism), se basan en criptografía de rejillas, concretamente en el problema denominado Module Learning With Errors (MLWE) explicado en secciones anteriores.

Histórico de los algoritmos

Ambos algoritmos son esencialmente equivalentes en su diseño criptográfico, pero ML-KEM representa una versión modularizada y más estrictamente definida de Kyber, adoptada como estándar por el NIST bajo la publicación oficial FIPS 203. Esta estandarización incluye especificaciones más detalladas para interoperabilidad, validación y cumplimiento en sistemas gubernamentales y comerciales.

Una diferencia práctica destacada es que:

- Kyber (NIST Round 3) fue el candidato antes de su estandarización.
- ML-KEM (FIPS 203) es la versión final, con ajustes formales y de implementación, recomendada para nuevos desarrollos.

Por ejemplo, el proyecto Open Quantum Safe (liboqs) anunció que dejará de incluir Kyber en sus futuras versiones en favor de ML-KEM, indicando una transición clara hacia el nuevo estándar [80].

En resumen, ML-KEM no cambia la seguridad ni la base matemática de Kyber, pero sí formaliza y refuerza su implementación. Por ello, se recomienda migrar cuanto antes a ML-KEM para garantizar compatibilidad futura y cumplimiento con estándares postcuánticos.

En la siguiente tabla se ha hecho una comparativa de rendimiento entre ML-KEM y Kyber, calculando la diferencia de tiempo para cada variante:

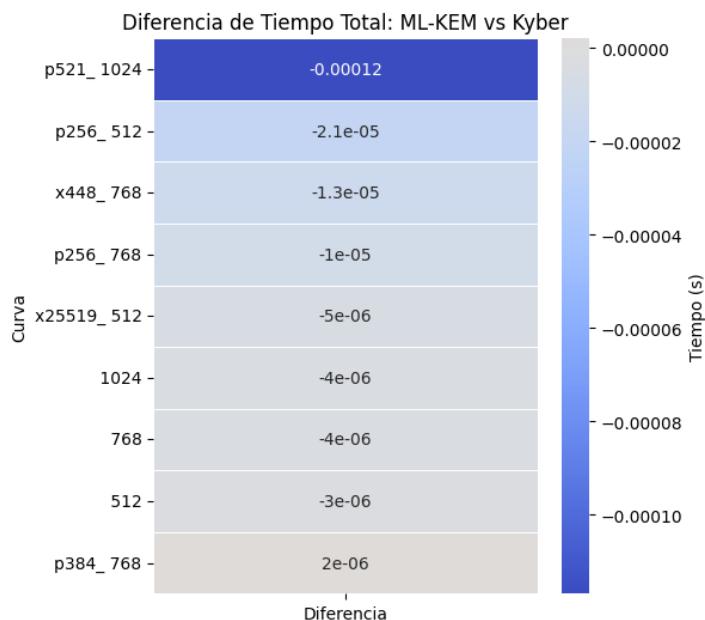


Tabla 27. Diferencia de tiempo entre ML-KEM y Kyber. Elaboración propia.

Los resultados experimentales muestran que ML-KEM tiende a ser igual o más eficiente que su predecesor Kyber en la mayoría de los casos evaluados. Sobre todo, para la variante con mayor complejidad, p521_MLKEM1024 tarda 0.00012s menos en ejecutarse que p521_Kyber1024.

Esto valida su elección por parte del NIST como nuevo estándar criptográfico postcuántico (FIPS 203) [81]. Las mejoras, aunque pequeñas, refuerzan su idoneidad para entornos donde cada milisegundo cuenta, especialmente al integrarse con curvas como P-521 o X25519 en combinaciones híbridas.

Comparación de rendimientos en distintos procesadores

En esta sección se presenta una comparación del rendimiento de las operaciones criptográficas del esquema ML-KEM entre dos procesadores diferentes: un AMD Ryzen 7 7700, cuyos resultados se encuentran publicados en el borrador de la IETF [82], y el procesador Intel i5-1135G7 de un equipo Lenovo ThinkPad utilizado en este proyecto. Las siguientes tablas muestran ambos resultados:

	Keygen/s	Encaps/s	Decaps/s
ML-KEM-512	244000	153000	202000
ML-KEM-768	142000	103000	134000
ML-KEM-1024	109000	77000	99000

Tabla 28. Rendimientos de ML-KEM en AMD Ryzen 7 [82].

	Keygens/s	Encaps/s	Decaps/s
ML-KEM-512	111886.2	131239.6	132171.1
ML-KEM-768	70344.8	81650.1	83975.1
ML-KEM-1024	51541.5	59101.4	56161.5

Tabla 29. Rendimientos de ML-KEM en procesador Intel i5-11367G. Elaboración propia.

El objetivo es analizar las diferencias de rendimiento en términos de operaciones por segundo para las tres funciones principales de ML-KEM: generación de claves (key generation), encapsulación (encapsulation) y desencapsulación (decapsulation). Para ello, se ha generado el siguiente mapa de calor que muestra la diferencia de operaciones por segundo en ambos equipos:

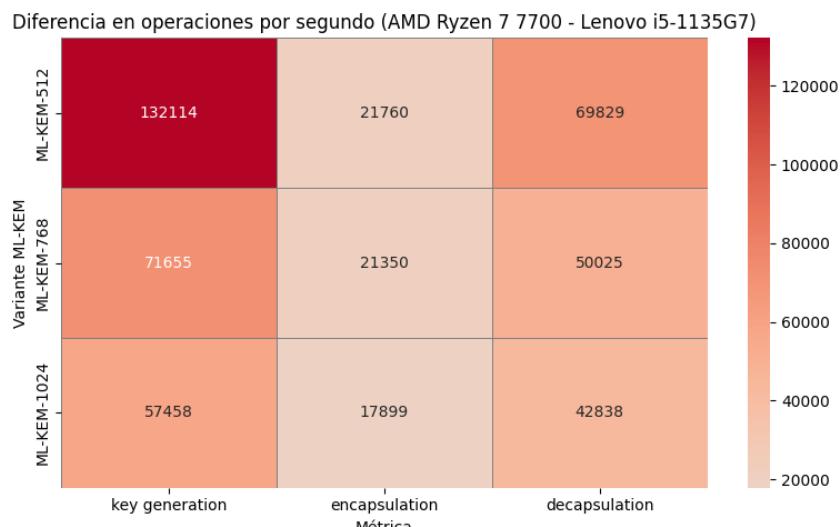


Figura 32. Mapa de calor diferencia en operaciones por segundo para 2 procesadores distintos.

El análisis comparativo entre el AMD Ryzen 7 7700 y el Lenovo i5-1135G7 en el rendimiento de ML-KEM revela que el Ryzen ofrece un rendimiento significativamente superior en todas las fases clave del proceso criptográfico: generación de claves, encapsulación y desencapsulación. Esta diferencia es especialmente notable en la generación de claves y en la variante con parámetros más pequeños (ML-KEM-512), donde la potencia y eficiencia del Ryzen le permiten manejar operaciones complejas de forma mucho más rápida.

En entornos donde se requiere procesamiento criptográfico intensivo y de baja latencia, como servidores que gestionan múltiples conexiones seguras simultáneas o infraestructuras de clave pública (PKI) a gran escala, el Ryzen 7 7700 puede mejorar significativamente la capacidad de respuesta, reduciendo tiempos de espera y aumentando la eficiencia.

Para dispositivos clientes, como laptops o estaciones de trabajo personales, la generación de claves puede ser menos frecuente o estar optimizada para ocurrir en momentos de baja demanda, por lo que un procesador como el i5-1135G7 puede ser suficiente. En general, los clientes suelen realizar operaciones de encapsulación y desencapsulación más que generación masiva de claves, lo que es viable para el i5 aunque con menor rendimiento absoluto.

Seguridad

En lo que respecta a la seguridad del esquema MLKEM (Module-Lattice-based Key Encapsulation Mechanism), incluso su variante con menor número de parámetros cumple con las recomendaciones de seguridad establecidas por el Instituto Nacional de Estándares y Tecnología (NIST). A continuación, se muestran las variantes estandarizadas de MLKEM y su correspondiente estimación de seguridad medida en bits:

Variante ML-KEM	Bits de Seguridad	Nivel de Seguridad NIST (PQC)
ML-KEM-512	128 bits	Nivel 1 (AES-128)
ML-KEM-769	192 bits	Nivel 3 (AES-192)
ML-KEM-1024	256 bits	Nivel 5 (AES-256)

Tabla 30. Bits de Seguridad de ML-KEM [65].

Una de las principales ventajas de MLKEM es su resistencia comprobada frente a ataques basados en el algoritmo de Shor, el cual es capaz de romper esquemas criptográficos tradicionales basados en factorización o logaritmos discretos (como RSA o ECDH) en un ordenador cuántico [Shor, 1994]. Esto convierte a MLKEM en una solución más duradera de cara al futuro post-cuántico.

Dado que todas las variantes seleccionadas cumplen con los criterios de seguridad establecidos por el NIST, se ha procedido a evaluar el impacto de las combinaciones híbridas en términos de rendimiento computacional, con especial atención a los entornos de despliegue con recursos limitados.

El siguiente mapa de calor representa el tiempo total (en segundos) empleado por diferentes combinaciones de curvas elípticas y variantes del esquema MLKEM, lo que permite analizar el impacto del esquema híbrido en el rendimiento.

- El eje **vertical** indica la curva elíptica utilizada (por ejemplo, p256, p384, x25519, etc.).
- El eje **horizontal** muestra la variante MLKEM (mlkem512, mlkem768, mlkem1024).
- Los valores en cada celda indican el **tiempo total de operación** (generación de claves, encapsulación y desencapsulación) expresado en **segundos**.

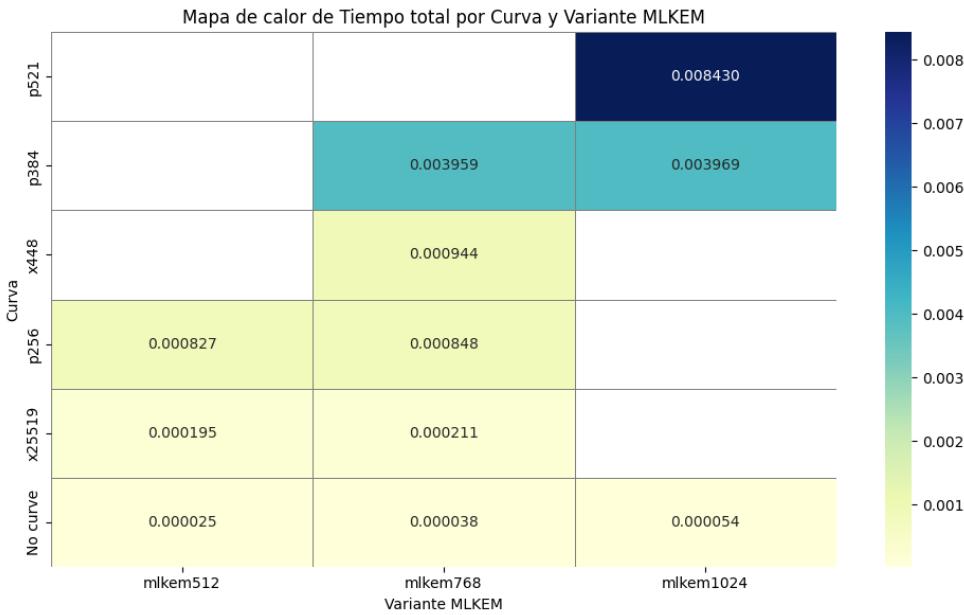


Figura 33. Mapa de calor de Rendimientos de ML-KEM en sus combinaciones híbridas.

La figura 31 revela una relación directa entre el nivel de seguridad y el coste computacional. Las combinaciones que integran curvas grandes como P-521 junto con ML-KEM-1024 alcanzan los mayores tiempos (~ 8.4 ms), mientras que el uso exclusivo de ML-KEM sin curva (post-cuántico puro) reduce el tiempo a unos pocos microsegundos. Además, se observa que combinaciones como X25519 + ML-KEM-768 o P-256 + ML-KEM-768 ofrecen un buen equilibrio entre seguridad y eficiencia, con tiempos inferiores a 1 ms. Estas combinaciones híbridas son especialmente relevantes, ya que permiten una migración gradual a esquemas post-cuánticos manteniendo compatibilidad con criptografía clásica.c

Cabe destacar que los resultados son coherentes con los tiempos reportados en implementaciones de referencia como wolfSSL y Cloudflare, que adoptan combinaciones similares para TLS 1.3, incluyendo P-256 + Kyber512 y X25519 + Kyber512/768 [83],[84]. Teniendo en cuenta los resultados obtenidos en este proyecto y los publicados por wolfSSL y Cloudflare, se han generado las siguientes recomendaciones:

Perfil de aplicación	Combinación recomendada	Comentario
Crítico / gubernamental	P-521 + MLKEM-1024	Seguridad máxima, alto coste
Equilibrado (cloud, web)	P-384 + MLKEM-768	Seguridad sólida, rendimiento razonable
Eficiente (IoT, móviles)	X25519 + MLKEM-512 o solo MLKEM512	Seguridad base, baja latencia

Tabla 31. Aplicaciones de esquemas híbridos en distintos entornos empresariales. Elaboración propia.

Estas decisiones deben contextualizarse según el entorno de ejecución, los requisitos regulatorios (por ejemplo, cumplimiento FIPS 203) y las amenazas a largo plazo derivadas de la computación cuántica.

5.1.4. Frodo

FrodoKEM también se apoya en criptografía de rejillas, pero a diferencia de ML-KEM, evita deliberadamente cualquier estructura algebraica optimizada. Se basa en el problema clásico de Learning With Errors (LWE), sin estructuras que pudieran ser explotadas por ataques avanzados. Esto lo hace menos eficiente en comparación con ML-KEM, pero conceptualmente más simple y, en teoría, más conservador en términos de seguridad, ya que reduce el riesgo de futuras debilidades estructurales [85].

Las variantes de FrodoKEM, AES y SHAKE, hacen referencia al modo de generación de números aleatorios y hashing usado en el esquema:

- FrodoKEM-AES usa AES⁵-128 internamente.
- FrodoKEM-SHAKE usa funciones SHAKE⁶ (SHA-3).

En la Figura 32 se grafican los tiempos totales de ejecución para las distintas variantes de FrodoKEM:

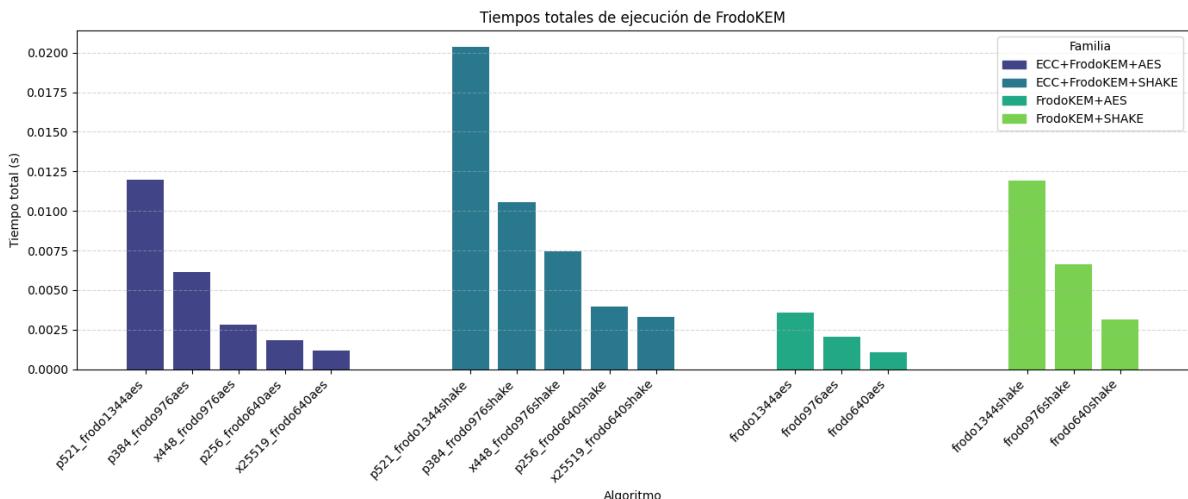


Figura 34. Tiempos totales de ejecución de las variantes FrodoKEM. Elaboración propia.

Se ha comprobado que la implementación de AES (Advanced Encryption Standard) para operaciones simétricas dentro del esquema resulta ser considerablemente más rápida que

⁵ Advanced Encryption Standard. Estándar de cifrado simétrico por bloques, usado como generador de números aleatorios en FrodoKEM.

⁶ Secure Hash Algorithm KECCAK Extendable Output Function. Función hash extendible de la familia SHA-3, empleada como generador determinista en FrodoKEM.

las operaciones de clave pública. Esto se debe a que AES, es un algoritmo simétrico ampliamente optimizado y soportado por hardware especializado.

En la Figura siguiente se presenta un mapa de calor con los tiempos totales (encapsulación + decapsulación) medidos para cada combinación de curva elíptica clásica + FrodoKEM, con y sin funciones SHAKE o AES.

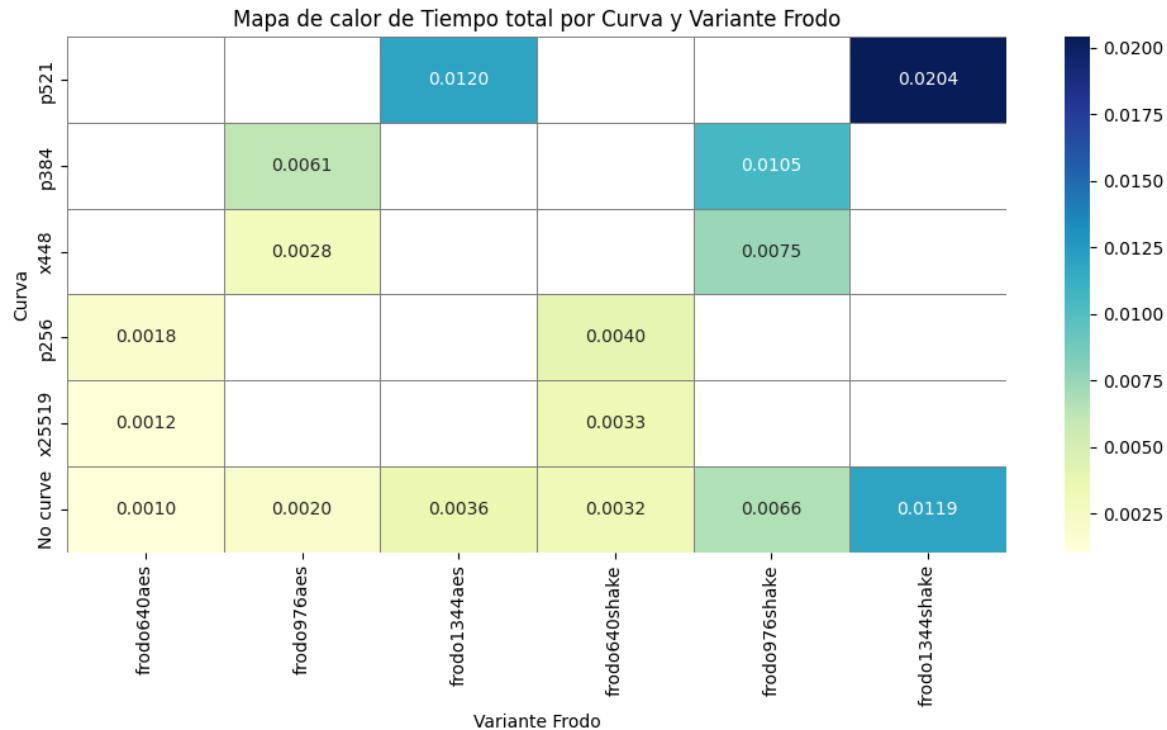


Figura 35. Mapa de calor de tiempo total de ejecución para FrodoKEM. Elaboración propia.

Los resultados confirman que FrodoKEM es significativamente más costoso en términos computacionales que ML-KEM. Incluso las variantes más ligeras, como frodo640aes, muestran tiempos superiores al milisegundo en la mayoría de los casos, y combinaciones como p521 + frodo1344shake alcanzan tiempos de hasta 20 ms, lo que limita su aplicabilidad en entornos sensibles a la latencia (IoT, aplicaciones móviles).

5.1.5. Bit Flipping Key Encapsulation (BIKE) y Hamming Quasi-Cyclic (HQC)

BIKE (Bit Flipping Key Encapsulation) y HQC (Hamming Quasi-Cyclic) pertenecen a la familia de encapsulación de clave basado en códigos correctores de errores, específicamente en códigos quasi-cíclicos de paridad impar. Su diseño parte del sistema McEliece, pero emplea un método de decodificación más eficiente conocido como bit flipping.

BIKE se fundamenta en la dificultad computacional del síndrome decoding problem, que consiste en recuperar el vector de error más probable dado un síndrome y una matriz generadora [86]. Sus niveles de seguridad son los siguientes:

Variante BIKE	Bits de Seguridad Aproximados	Nivel de Seguridad NIST (PQC)
BIKEI1	128 bits	Nivel 1 (AES-128)
BIKEI3	192 bits	Nivel 3 (AES-192)
BIKEI5	256 bits	Nivel 5 (AES-256)

Tabla 32. Bits de seguridad de BIKE [87].

Por otro lado, HQC (Hamming Quasi-Cyclic) también pertenece a la familia de algoritmos basados en códigos, pero utiliza códigos lineales con estructuras algebraicas distintas. Se basa en la dureza del problema de decodificación de códigos aleatorios (Random Code Decoding), junto con la dificultad adicional de preservar la indistinguibilidad entre claves públicas y códigos aleatorios [88]. Sus niveles de seguridad son los siguientes:

Variante HQC	Bits de Seguridad Aproximados	Nivel de Seguridad NIST (PQC)
HQC-128	128 bits	Nivel 1 (AES-128)
HQC-192	192 bits	Nivel 3 (AES-192)
HQC-256	256 bits	Nivel 5 (AES-256)

Tabla 33. Bits de seguridad de HQC [89].

Ambos esquemas han sido considerados como finalistas o alternativos en el proceso de estandarización del NIST.

La siguiente gráfica muestra las velocidades medias de operación para distintas variantes de los algoritmos BIKE y HQC. Los tiempos están representados en escala logarítmica para facilitar la comparación entre órdenes de magnitud:

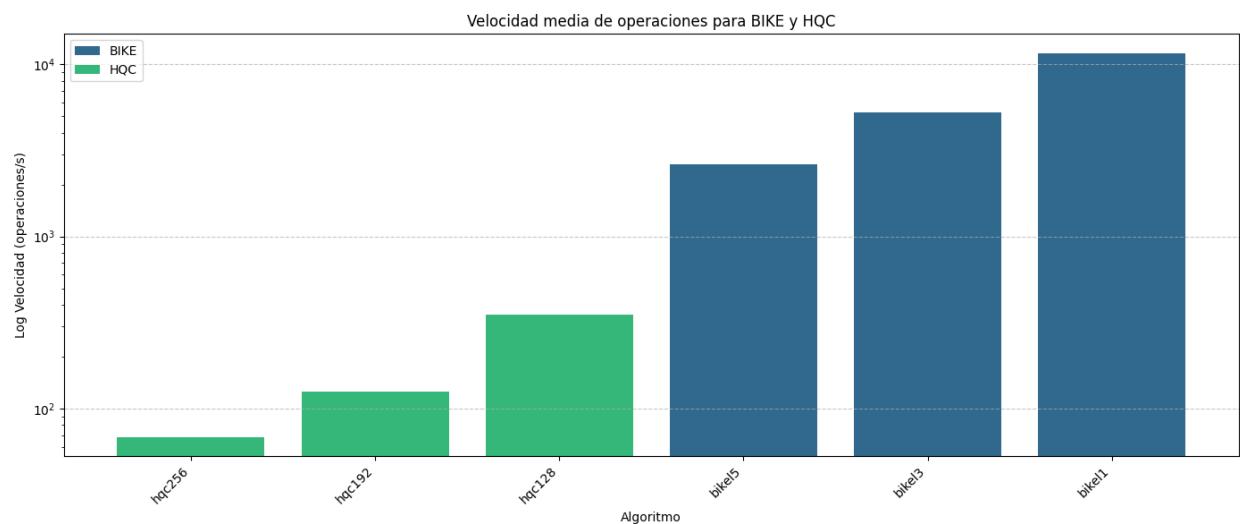


Figura 36. Comparativa de rendimientos de HQC y BIKE.

Se observa que, en general, las variantes de BIKE presentan velocidades significativamente mayores que sus equivalentes de HQC, especialmente en los niveles de seguridad más bajos. Esto refleja la eficiencia del método de decodificación por bit flipping empleado por BIKE.

5.1.6. Comparativa

Para comparar los algoritmos de encapsulación de clave anteriores, se han unido en una tabla clasificados por Familia y Bits de seguridad.

Algoritmo	Familia	Bits de seguridad AES
bikel5	BIKE	256
bikel3	BIKE	192
bikel1	BIKE	128
ecp-384	ECC	192
ecp-521	ECC	256
x448	ECC	224
x25519	ECC	128
ecp-256	ECC	128
p384_bikel3	ECC+BIKE	192
p521_bikel5	ECC+BIKE	256
x448_bikel3	ECC+BIKE	192
p256_bikel1	ECC+BIKE	128
x25519_bikel1	ECC+BIKE	128
p384_frodo976aes	ECC+FrodoKEM+AES	192
x448_frodo976aes	ECC+FrodoKEM+AES	192
p256_frodo640aes	ECC+FrodoKEM+AES	128
x25519_frodo640aes	ECC+FrodoKEM+AES	128
p521_frodo1344aes	ECC+FrodoKEM+AES	256
p521_frodo1344shake	ECC+FrodoKEM+SHAKE	256
x448_frodo976shake	ECC+FrodoKEM+SHAKE	192
p384_frodo976shake	ECC+FrodoKEM+SHAKE	192
p256_frodo640shake	ECC+FrodoKEM+SHAKE	128
x25519_frodo640shake	ECC+FrodoKEM+SHAKE	128
x25519_hqc128	ECC+HQC	128
p384_hqc192	ECC+HQC	192
x448_hqc192	ECC+HQC	192
p521_hqc256	ECC+HQC	256
p256_hqc128	ECC+HQC	128
x25519_ml kem512	ECC+MLKEM	128
x25519_ml kem768	ECC+MLKEM	192
p384_ml kem1024	ECC+MLKEM	256
p256_ml kem512	ECC+MLKEM	128
p384_ml kem768	ECC+MLKEM	192
x448_ml kem768	ECC+MLKEM	192
p521_ml kem1024	ECC+MLKEM	256

p256_ml kem768	ECC+MLKEM	192
frodo976aes	FrodoKEM+AES	192
frodo640aes	FrodoKEM+AES	128
frodo1344aes	FrodoKEM+AES	256
frodo640shake	FrodoKEM+SHAKE	128
frodo1344shake	FrodoKEM+SHAKE	256
frodo976shake	FrodoKEM+SHAKE	192
hqc256	HQC	256
hqc192	HQC	192
hqc128	HQC	128
mlkem1024	MLKEM	256
mlkem768	MLKEM	192
mlkem512	MLKEM	128
rsa2048	RSA	112
rsa3072	RSA	128
rsa4096	RSA	140
rsa7680	RSA	192
rsa1024	RSA	80
rsa15360	RSA	256

Tabla 34. Tabla comparativa de los algoritmos KEM. Elaboración propia.

Mediante esta organización, se ha podido generar una gráfica de rendimientos distribuida por niveles de seguridad:

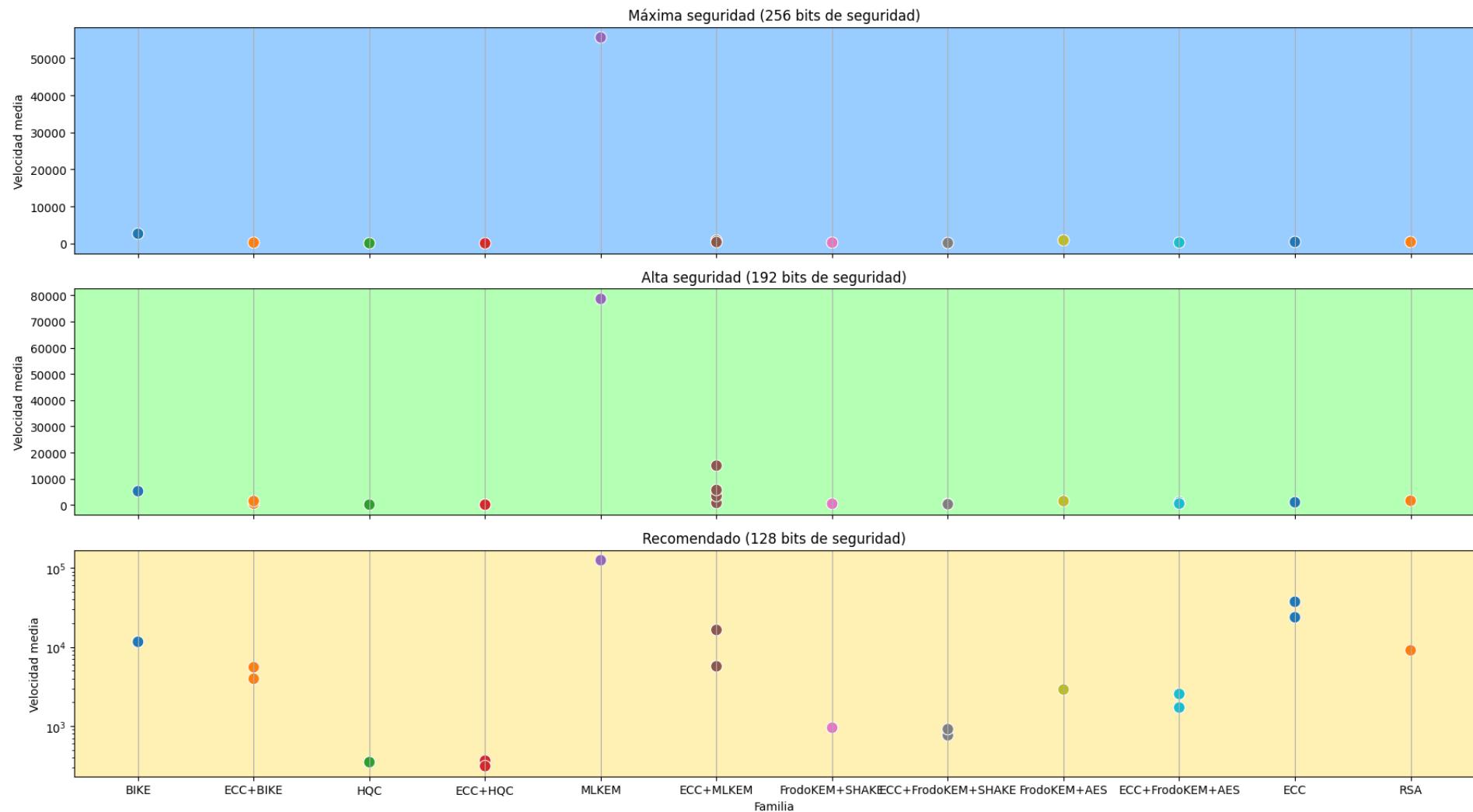


Figura 37. Comparativa de rendimientos por Familia de Algoritmo y nivel de seguridad. Elaboración propia.

La gráfica presenta una comparación entre diferentes familias de algoritmos criptográficos agrupados según tres niveles de seguridad: máxima seguridad (256 bits), alta seguridad (192 bits) y recomendado (128 bits). En el contexto de TLS, estas observaciones permiten identificar qué algoritmos son más adecuados según la relación entre velocidad y nivel de seguridad. Los niveles de seguridad representados son:

1. **Máxima seguridad (256 bits):**
La gráfica muestra una gran disparidad en la velocidad media. Por ejemplo, MLKEM destaca por su altísima velocidad en comparación con otros como HQC o ECC+BIKE que son mucho más lentos. No obstante, la velocidad media de su versión híbrida, MLKEM+ECC, no ofrece ventajas de velocidad tan significativas.
2. **Alta seguridad (192 bits):**
En este rango, la mayoría de los algoritmos presentan velocidades medias bajas en comparación con la máxima seguridad, salvo MLKEM que sigue manteniendo un desempeño superior. En comparación con las velocidades de la franja 256 bits de seguridad, esta franja puede ofrecer una mejora significativa en la velocidad. Esto es un punto clave para aplicaciones empresariales que requieren un equilibrio entre seguridad y rapidez, como servicios web con alta concurrencia.
3. **Para 128 bits de seguridad**
Aquí, los algoritmos muestran una mayor velocidad media, siendo ECC y ECC+FrodoKEM+AES opciones destacadas. Este nivel es común en implementaciones TLS para la mayoría de las empresas que buscan una protección adecuada sin sacrificar la eficiencia. Además, los algoritmos híbridos ECC+BIKE y ECC+MLKEM presentan velocidades similares a RSA, por lo que la migración no ocasionaría desajustes operativos.

Estas observaciones pueden traducirse al ámbito empresarial de la siguiente manera:

- **Empresas con alta sensibilidad en la información (banca, salud, gobierno):**
Se recomienda optar por variantes de algoritmos que ofrecen máxima seguridad (256 bits), priorizando aquellos como ECC+MLKEM que presentan un equilibrio entre alta protección y velocidad.
- **Empresas con necesidades mixtas (tecnología, comercio electrónico):**
Los algoritmos de alta seguridad, especialmente aquellos híbridos como ECC+MLKEM o FrodoKEM+SHAKE, ofrecen un buen compromiso para mantener la seguridad frente a amenazas cuánticas y al mismo tiempo una velocidad aceptable para conexiones TLS.
- **Empresas con alta demanda de rendimiento y menor riesgo percibido (pymes, startups):**
En estos casos, los algoritmos recomendados (128 bits) como ECC+BIKE, o RSA siguen siendo apropiados, proporcionando velocidades elevadas y una seguridad suficiente para aplicaciones que no contengan datos críticos.

La elección del algoritmo criptográfico en TLS debe considerar no solo el nivel de seguridad sino también la velocidad de procesamiento. La gráfica evidencia que no siempre los algoritmos de mayor seguridad son los más lentos, y que existen opciones eficientes para cada nivel. Las empresas deben evaluar sus riesgos y capacidades para seleccionar el algoritmo que mejor se adapte a sus necesidades, considerando la evolución hacia la criptografía postcuántica.

La Figura 36 muestra un mapa de calor que representa la media del tiempo total de ejecución para diferentes combinaciones de curvas elípticas y familias de algoritmos postcuánticos. En cada celda se indica el tiempo promedio (en segundos) requerido para ejecutar la operación criptográfica híbrida correspondiente.

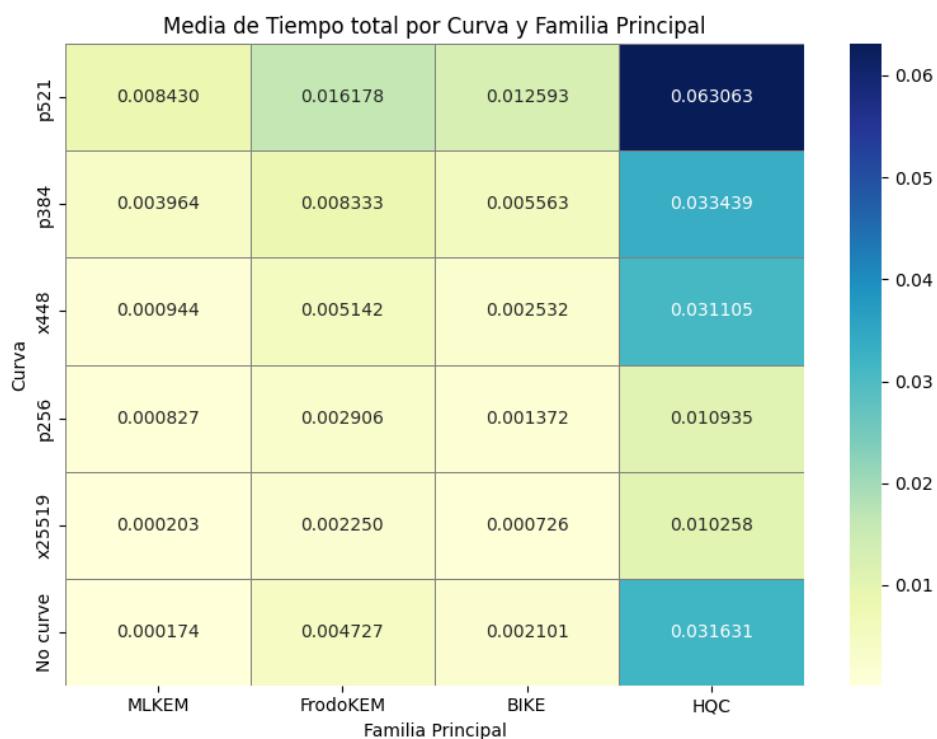


Figura 38. Mapa de calor de rendimiento según la curva elíptica aplicada en los algoritmos KEM. Elaboración propia.

Se observa que, en general, las combinaciones que emplean curvas elípticas de mayor tamaño, como p521, tienden a presentar un mayor coste computacional. Por ejemplo, la combinación de p521 con HQC alcanza el tiempo más elevado de la tabla (0.063063 s), mientras que opciones como x25519 con MLKEM (0.000203 s) ofrecen una ejecución mucho más eficiente. Este comportamiento pone de manifiesto la importancia del diseño eficiente de algoritmos híbridos, donde se debe equilibrar el nivel de seguridad con los recursos computacionales disponibles.

Las diferencias entre familias postcuánticas también son notables: MLKEM y BIKE suelen tener un rendimiento más favorable en la mayoría de las combinaciones, mientras que HQC y FrodoKEM son opciones más costosas en tiempo. Estos resultados indican que todavía existen desafíos importantes en términos de rendimiento para algunos algoritmos postcuánticos.

Asimismo, se presentan los resultados de ejecución con algoritmos exclusivamente postcuánticos (sin curva). Para la mayoría de los casos, resultan más eficientes, y por ello, se plantea la pregunta: **¿Es viable un enfoque exclusivamente postcuántico?**

Aunque el uso exclusivo de criptografía post-cuántica es factible y será el objetivo a largo plazo, en el contexto actual presenta ciertos inconvenientes derivados de la madurez aún limitada de muchas bibliotecas postcuánticas. Por tanto, en el contexto de una transición progresiva, el uso de esquemas híbridos se considera la opción más segura y práctica.

En la Figura 37, se representa un gráfico de barras apiladas, donde cada sección corresponde a una operación de KEM:

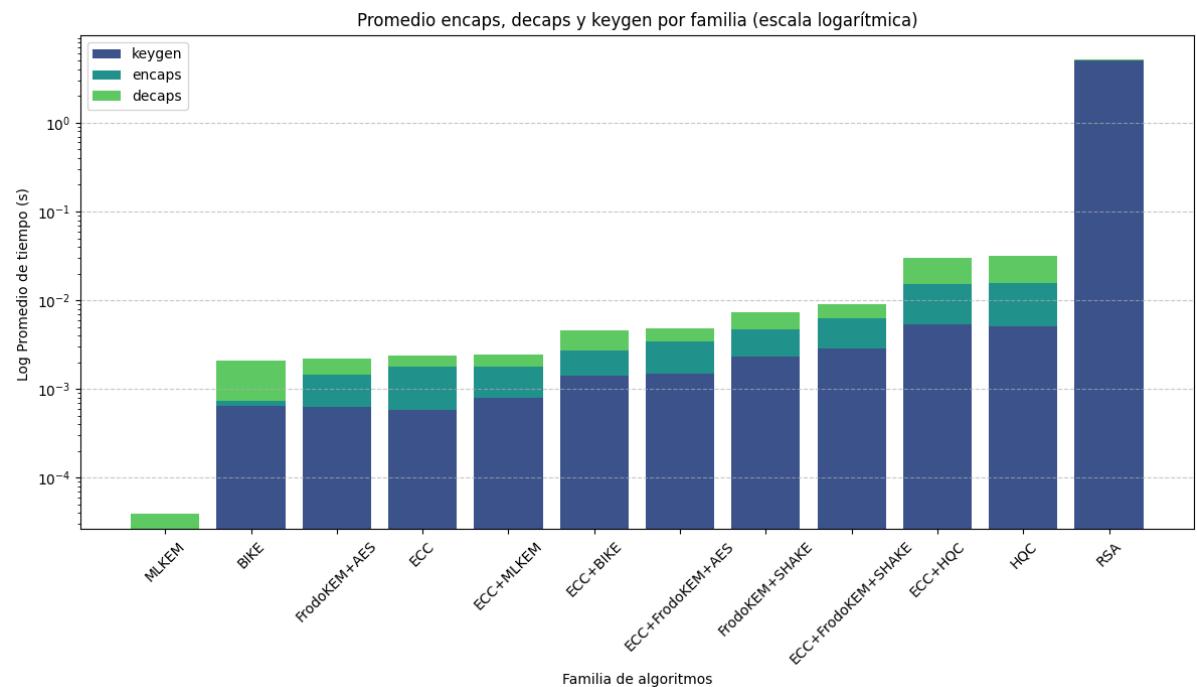


Figura 39. Promedio de velocidad de operaciones para Algoritmos KEM. Elaboración propia.

Se observa que, en general, la operación más lenta en un KEM es la generación de claves (keygen), seguida por decapsulación (decaps), y luego la encapsulación (encaps). Esto ocurre porque la generación de claves suele involucrar operaciones más costosas, especialmente para algoritmos como RSA que requieren claves largas para garantizar su seguridad.

Las otras 2 operaciones de KEM son:

- Encapsulación (encaps): El emisor encapsula la clave secreta usando la clave pública. Esta operación tiende a ser relativamente rápida porque no involucra cálculos tan complejos ni acceso a claves privadas.
- Desencapsulación (decaps): El receptor debe usar su clave privada para decapsular la clave secreta. Este proceso suele ser más costoso computacionalmente respecto a la encapsulación porque implica operaciones de descifrado o cálculos intensivos.

En la Figura 38 se grafican el promedio de tiempo de las operaciones de desencapsulación y encapsulación para los algoritmos KEM:

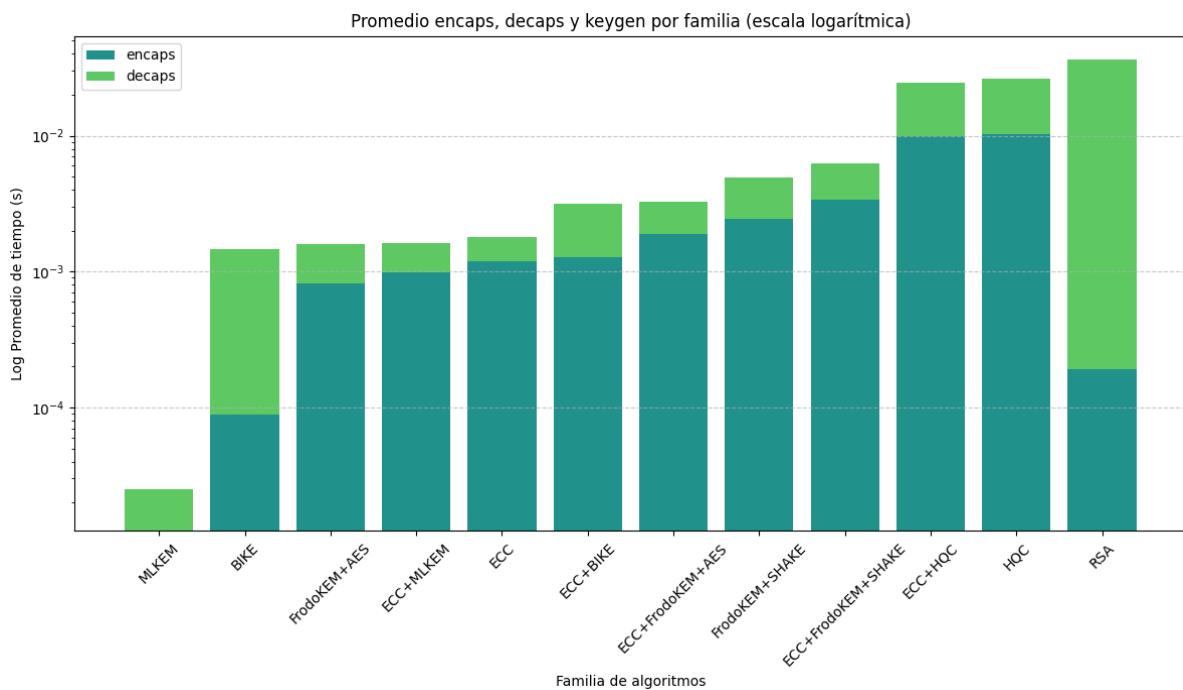


Figura 40. Promedio de encapsulación y decapsulación por algoritmo KEM.

Los resultados de encapsulación y des encapsulación muestran que RSA sigue presentando el mayor tiempo promedio de operación entre los algoritmos analizados, seguido por HQC y su combinación híbrida (ECC+HQC).

También se destaca el aumento en ML-KEM híbrido respecto a su contraparte postcuántica. Cuando la criptografía postcuántica se desarrolle totalmente y no sean necesarias las variantes híbridas, ML-KEM podría aportar grandes beneficios.

Por otra parte, para que RSA alcance niveles aceptables de seguridad frente a amenazas actuales, se requieren claves de al menos 2048 bits o superiores. Para futuras amenazas, podrían necesitarse aún mayores tamaños de clave, o directamente no ser viable. Este aumento en el tamaño de clave implica un incremento significativo en el tiempo de ejecución, particularmente en la operación de generación de claves.

Incluso los algoritmos híbridos con peores rendimientos siguen siendo más eficientes que RSA. Por otra parte, ECC se encuentra a la par con algoritmos híbridos como ECC+MLKEM y ECC+BIKE. En conclusión, migrar a esquemas híbridos, no solo aporta mayor seguridad, sino que también es una medida eficiente desde el punto de vista computacional.

6. IMPLICACIONES ÉTICAS E IMPACTO SOCIAL

En el contexto actual de transformación digital, la informática no solo constituye una herramienta técnica, sino también un campo con profundas implicaciones éticas y sociales. En este sentido, organizaciones como la Association for Computing Machinery (ACM) juegan un papel clave en la promoción de principios que guían el ejercicio profesional responsable.

“La ACM es una organización internacional y la sociedad científica y educativa más grande del mundo en el campo de la informática. Fundada en 1947, la ACM tiene como objetivo promover el avance de la ciencia y la práctica de la informática, así como fomentar la colaboración y el intercambio de conocimientos entre profesionales de la informática y académicos de todo el mundo. La ACM establece estándares éticos y profesionales para los practicantes de la informática a través de su Código de Ética y Conducta Profesional. Este código proporciona directrices fundamentales para el comportamiento ético en el ámbito de la informática y aborda diversos aspectos, como la contribución a la sociedad, la evitación de daños, la honestidad, la no discriminación, el respeto a la privacidad y la confidencialidad, entre otros.” [90]

ACM reafirma su compromiso en su misión y valores fundamentales en todas las actividades científicas y educativas, y también patrocina eventos a nivel mundial relacionados con las ciencias de la computación, como el **ACM International Collegiate Programming Contest**. También ha estado asociada a hitos históricos en el ámbito tecnológico, como la emblemática partida de ajedrez entre Garry Kasparov y la supercomputadora de IBM, Deep Blue [91].



Figura 41. Imagen de Deep Blue jugando contra Garry Kasparov [91].

Para la elaboración de este proyecto se han tenido en cuenta los principios del Código de Ética y Conducta Profesional de la ACM, especialmente en lo referente a la protección de la privacidad, la seguridad de los usuarios y la responsabilidad profesional en el desarrollo de soluciones tecnológicas.

Durante el desarrollo de todo el proyecto, se ha mantenido un enfoque honesto y transparente en la presentación de los resultados, evitando manipulaciones y omisiones; teniendo en cuenta que el impacto del uso de algoritmos postcuánticos puede variar en función de las características técnicas de cada sistema, respetando así el principio de fiabilidad recogido en el artículo 1.3: "Sé honesto y fiable" [92].

Asimismo, se establece que este proyecto supone una prueba de concepto en un entorno controlado, previo a una posible implementación en entornos reales a gran escala. La escalabilidad y adaptación de esta solución requerirán futuras pruebas específicas por parte de profesionales, teniendo en cuenta variables como el número de usuarios, el flujo de información, las capacidades de los servidores y la compatibilidad con sistemas existentes.

6.1. VALOR DEL PROYECTO

Este proyecto responde a una necesidad técnica actual y refleja un compromiso ético con la protección futura de los datos personales ante amenazas emergentes. En particular, se enfoca en un problema crítico: la preparación de las infraestructuras digitales frente a los desafíos que plantea la computación cuántica.

La propuesta contribuye a la evaluación práctica del rendimiento de algoritmos criptográficos postcuánticos, anticipando medidas de seguridad que podrían ser necesarias en un futuro cercano. Con ello, el proyecto tiene un claro componente académico y divulgativo ya que puede servir como referencia o punto de partida para otros investigadores, instituciones educativas o centros tecnológicos interesados en explorar, probar o desarrollar estas soluciones criptográficas. Fomentando así la colaboración y el avance en un campo de creciente relevancia.

6.2. ALCANCE

Este proyecto se enmarca en el ámbito de la ciberseguridad aplicada a la protección de datos en redes corporativas e institucionales, especialmente en el contexto de la transición hacia algoritmos resistentes a ataques cuánticos.

Está dirigido a administradores de sistemas, desarrolladores de software, ingenieros de redes y responsables de infraestructura TI interesados en integrar soluciones criptográficas seguras en sus entornos. De manera indirecta también puede beneficiar a usuarios finales, clientes y entidades públicas o privadas cuyos datos personales o confidenciales se transmiten a través de redes.

Desde el punto de vista técnico, el proyecto analiza rigurosamente el rendimiento y viabilidad de los algoritmos criptográficos en protocolos de comunicación TLS.

El proyecto adopta los siguientes compromisos:

- Garantizar que las pruebas se han realizado en condiciones representativas, como el uso de un portátil de gama media (ThinkPad procesador i5), frecuentemente utilizado en muchas redes internas empresariales.
- Utilizar un software legítimo y de código abierto, respetando las licencias correspondientes y evitando prácticas no éticas como el uso de software pirata.
- Documentar de los resultados de forma clara y reproducible, fomentando una ciencia abierta y responsable.

6.3. IMPLICACIONES DESDE EL PUNTO SOCIAL

En relación con el artículo 2.7. *Fomentar la conciencia y comprensión pública sobre la informática, las tecnologías relacionadas y sus consecuencias* [92], este proyecto fomenta la concienciación en ciberseguridad tanto en el entorno profesional como en la ciudadanía general. En particular, se informa sobre los riesgos que los ordenadores cuánticos pueden representar para la privacidad de los datos, promoviendo la adopción temprana de tecnologías más seguras.

Actualmente, la criptografía postcuántica se encuentran en fase experimental y solo está implementada de manera híbrida en algunos servidores. Una de las principales limitaciones es que, para establecer este tipo de conexión segura, tanto el cliente como el servidor deben tener habilitado el protocolo TLS Quantum Safe, el cual se explora y detalla en este proyecto. A medida que aumente la conciencia sobre la necesidad de migrar hacia estas tecnologías, las conexiones resistentes a ataques de ordenadores cuánticos serán cada vez más accesibles.

Otra implicación social consiste en reforzar la confianza digital al demostrar que es posible implementar tecnologías de seguridad avanzadas sin grandes infraestructuras. Contribuyendo positivamente a la sociedad según lo indica el principio ético 1.1. *Contribuir a la sociedad y al bienestar humano, reconociendo que todas las personas son partes interesadas en la informática* [92].

Este proyecto busca prevenir uno de los principales riesgos asociados con la computación cuántica: la posible ruptura de los actuales sistemas de encriptación y, con ello, la pérdida de privacidad de los datos. Para ello, se han realizado pruebas de conexiones que emplean criptografía moderna, postcuántica e híbrida y se ha analizado su impacto en términos de tiempos de ejecución y seguridad.

Puesto que todas las personas son partes interesadas en la informática, se persigue que las conexiones seguras puedan ser realizadas por cualquier usuario, sin necesidad de equipamiento especializado. Todo el proceso se ha llevado a cabo en un portátil de gama media, comúnmente utilizado en entornos corporativos (ThinkPad con procesador i5). Asimismo, todas las herramientas de software utilizadas para desplegar la infraestructura son gratuitas y de acceso público, lo que garantiza la reproducibilidad del experimento y su aplicabilidad práctica para los usuarios.

6.4. IMPLICACIONES DESDE EL PUNTO DE VISTA ECONÓMICO

El uso de herramientas gratuitas y de código abierto reduce las barreras de entrada para las empresas y usuarios interesados en iniciar la transición hacia tecnologías *Quantum Safe*. No obstante, la formación del personal técnico y la validación de la solución representan inversiones clave. Estas inversiones son estratégicas, ya que la adopción temprana puede suponer un ahorro económico a medio y largo plazo, al evitar los costes asociados a posibles brechas de seguridad causadas por criptografía obsoleta.

La transición también puede generar nuevas oportunidades de negocio, especialmente para empresas especializadas en ciberseguridad, consultoría y servicios de integración. Aquellas organizaciones que se posicionen como pioneras en la adopción de tecnologías *Quantum Safe* podrían beneficiarse de ventajas competitivas tanto en términos de reputación como de cumplimiento normativo anticipado.

6.5. IMPLICACIONES MEDIOAMBIENTALES

El proyecto ha sido desarrollado en un entorno de bajo consumo energético, utilizando un ordenador portátil de gama media. Esta elección no solo reduce el impacto energético directo, sino que también refleja un compromiso con prácticas sostenibles en las primeras etapas del ciclo de vida tecnológico.

Además, se ha demostrado que los algoritmos post cuánticos son más eficientes y requieren menos capacidad computacional en comparación con algoritmos actuales como RSA. No obstante, en escenarios de implementación a gran escala, un flujo elevado de conexiones conllevaría una mayor capacidad de cálculo, lo que implica el uso de servidores de alto rendimiento con un consumo energético significativamente superior. Este aumento de demanda energética, especialmente en centros de datos, puede tener un impacto considerable en términos de emisiones de carbono.

Por ello, este aspecto debe ser considerado cuidadosamente en futuras fases del despliegue, buscando siempre el equilibrio entre seguridad, rendimiento y sostenibilidad. Será esencial evaluar alternativas como el uso de energías renovables en los centros de datos, y la implementación de arquitecturas eficientes desde el punto de vista energético.

En este sentido, soluciones como la plataforma *IBM LinuxONE*, diseñada para consolidar cargas de trabajo intensivas en procesamiento con un consumo energético optimizado, representan una opción viable. Según IBM, el uso de *LinuxONE* puede reducir el consumo energético en hasta un 75% y disminuir la huella de carbono en más de 850 toneladas métricas anuales al reemplazar servidores x86 equivalentes [93].



Figura 42. Métricas de consumo energético por partición en servidores IBM LinuxONE [93].

Incorporar este tipo de infraestructuras sostenibles, junto con métricas de eficiencia energética, permitirá alinear la ciberseguridad con los objetivos de responsabilidad ambiental de las organizaciones.

6.6. MARCO LEGAL

Este proyecto se acoge a los siguientes reglamentos relativos a la protección de datos:

- Reglamento General de Protección de Datos (RGPD) de la Unión Europea [94].
- Ley Orgánica 3/2018 en España [95].

Ambas normativas establecen la necesidad de aplicar medidas técnicas adecuadas para garantizar la seguridad de los datos personales. En particular, el artículo 32 del RGPD exige que el responsable y el encargado del tratamiento apliquen medidas técnicas y organizativas apropiadas para garantizar un nivel de seguridad adecuado al riesgo. Entre las medidas sugeridas, menciona explícitamente: "la seudonimización y el cifrado de los datos personales" (art. 32.1.a). La implementación de un protocolo TLS Quantum Safe responde directamente a este requerimiento, al utilizar cifrados que permiten la protección de los datos frente a amenazas emergentes.

Se han tenido en cuenta las siguientes normativas españolas relacionadas con la ciberseguridad:

- Ley 36/2015 de Seguridad Nacional, que reconoce la ciberseguridad como una prioridad estratégica del Estado [96].
- Real Decreto 43/2021, que regula la seguridad de las redes y sistemas de información, y que obliga a los operadores de servicios esenciales y proveedores de servicios digitales a implementar medidas técnicas y organizativas adecuadas [97].

Además, se contempla el Código de Derecho de la Ciberseguridad, que engloba principios y normas orientadas a proteger los sistemas informáticos y las infraestructuras digitales [98].

En este contexto, la adopción de protocolos con capacidades Quantum Safe se considera una acción proactiva y alineada con el principio de “seguridad por diseño y por defecto”, promovido tanto por el RGPD como por las directrices del NIS. Este enfoque no solo permite cumplir con los estándares actuales, sino que anticipa futuros requerimientos técnicos y legales, promoviendo una cultura de responsabilidad digital y protección de la privacidad.

El desarrollo del proyecto también observa la normativa relativa a la propiedad intelectual:

- Real Decreto Legislativo 1/1996, que regula la propiedad intelectual en España [99].
- Ley 16/1993, que incorpora la Directiva europea sobre la protección jurídica de programas de ordenador [100].

Siguiendo el principio ético general 1.5 Respeta el trabajo necesario para producir nuevas ideas, inventos, obras creativas y artefactos informáticos [92], el proyecto:

- Acredita el trabajo de la organización Open Quantum Safe y la contribución del Instituto Nacional de Estándares y Tecnología (NIST) en la estandarización de algoritmos de criptografía postcuántica, fundamentales para este desarrollo.
- Utiliza herramientas de software libre y de código abierto como Ubuntu Linux, Podman y liboqs, cada una bajo sus respectivas licencias (GPL/MIT, Apache 2.0 y MIT). Se ha respetado íntegramente el contenido de dichas licencias, citando a sus autores y evitando cualquier uso indebido de marca o código. Se adjuntan enlaces a las licencias en el anexo A.

El uso de estas tecnologías no solo cumple con los requisitos legales vigentes en materia de propiedad intelectual, sino que también promueve valores éticos como la transparencia, la colaboración abierta y el respeto por el trabajo de terceros. Asimismo, se rechaza cualquier forma de piratería o uso no autorizado de software, alineando el proyecto con una cultura de responsabilidad digital.

6.7. RIESGOS DEL PROYECTO

El desarrollo e implementación de este proyecto conlleva ciertos riesgos que, de no gestionarse adecuadamente, podrían comprometer la seguridad de los datos. Entre ellos se incluyen posibles fallos en la configuración, vulnerabilidades no detectadas o la exposición a ataques a gran escala que exploten debilidades en los sistemas de cifrado.

Un caso publicado por el Instituto Nacional de Ciberseguridad (INCIBE) expone a una empresa que fue víctima de un ataque de ransomware, en el que sus servidores y sistemas de almacenamiento en red fueron cifrados maliciosamente, comprometiendo información personal sensible como los DNI de sus clientes [101].

Este tipo de incidentes demuestra la importancia de adoptar mecanismos de cifrado robustos y actualizados. Migrar hacia protocolos como TLS Quantum Safe y llevar a cabo una evaluación exhaustiva de los riesgos asociados es fundamental para evitar la interceptación o manipulación de información, especialmente frente a posibles avances en tecnologías como la computación cuántica.

Riesgo	Descripción	Prob.	Acción	Imp
R1	Dependencia de bibliotecas externas desactualizadas con vulnerabilidades.	Alta	Realizar seguimiento activo de actualizaciones y vulnerabilidades. Usar versiones estables y ampliamente probadas.	Alto
R2	Incompatibilidad con clientes o sistemas existentes que no soporten TLS Quantum Safe.	Alta	Utilizar <i>Quantum Safe</i> en redes internas donde se tenga control sobre los servidores.	Medio
R3	Latencia demasiado elevada debido a la complejidad de los algoritmos ejecutados.	Media	Analizar el rendimiento para cada caso y aplicar optimizaciones.	Medio
R4	Filtración de datos por configuración incorrecta del servidor.	Media	Verificar configuración de seguridad (headers, certificados, protocolos activos). Usar herramientas de análisis y escaneo.	Alto
R5	Fallo en la implementación del cifrado, comprometiendo la seguridad.	Medio	Validar con pruebas de penetración (penetration testing) y realizar simulacros de ataques (Red Team) bajo la supervisión de profesionales.	Alto
R6	Cambio en los estándares del NIST.	Medio	Priorizar algoritmos aprobados por NIST. Documentar claramente qué algoritmos se usan y su estado.	Bajo
R7	Descubrimiento de nuevos algoritmos de cifrado más resistentes no incluidos en este proyecto.	Medio	Mantenerse actualizado sobre las nuevas investigaciones en respecto a los algoritmos <i>postcuánticos</i> .	Bajo
R8	Fallo en el sistema operativo utilizado (Ubuntu)	Bajo	Uso de contenedores para aislar la implementación del sistema operativo.	Alto
R10	Alto consumo energético derivado del aumento de capas de seguridad.	Bajo	Uso de contenedores y entornos ligeros, que reduzcan el impacto en el sistema base.	Medio
R11	Desarrollo de nuevas tecnologías	Bajo	Revisar periódicamente la solución según los avances	Alto

	comprometan los algoritmos Quantum Safe utilizados.		tecnológicos. Seguir las recomendaciones oficiales del NIST y otras entidades de estandarización.	
R12	Aumento de consumo energético, impactando negativamente a la sostenibilidad de la organización.	Alta	Utilizar servidores como LinuxONE de IBM, que optimizan el consumo energético. Evaluar periódicamente el impacto ambiental y priorizar el uso de centros de datos alimentados por energías renovables.	Medio

	Alta	R3	R2, R12	R1
Probabilidad	Media	R6, R7	R3, R6	R4, R5
	Baja		R10	R8, R11
		Bajo	Medio	Alto

Impacto

Para mitigar los riesgos más críticos identificados durante el desarrollo del proyecto, se propone implementar una estrategia de monitorización activa de las bibliotecas y herramientas empleadas, así como un seguimiento continuo de las actualizaciones publicadas por los organismos de estandarización, como el NIST, especialmente en lo relativo a algoritmos postcuánticos.

Adicionalmente, se recomienda la realización periódica de pruebas de penetración (Pen Testing), que son realizadas por hackers éticos, con el objetivo de detectar posibles vulnerabilidades en la infraestructura implementada.

En la gestión de los riesgos derivados de utilizar un protocolo TLS modificado con algoritmos postcuánticos, se tuvo en cuenta el principio de liderazgo profesional 3.6. *cuidado al modificar o retirar sistemas* [92]. Este principio resalta la importancia de realizar cambios en los sistemas de manera controlada y fundamentada, minimizando cualquier impacto negativo en los usuarios finales y en la operatividad de las organizaciones.

El protocolo TLS modificado a TLS *Quantum Safe* añade una capa adicional de seguridad para proteger los datos transmitidos entre servidores y clientes. Sin embargo, esta modificación no afecta la interfaz de usuario ni el funcionamiento del servidor, garantizando que los

cambios sean transparentes para el usuario final. La única alteración es el reemplazo de los algoritmos de criptografía asimétricos actuales por algoritmos postcuánticos e híbridos, que aseguran una mayor resistencia frente a posibles amenazas derivadas de la computación cuántica.

6.8. CONCLUSIONES

En relación con todo lo expuesto, dado el valor que presenta el proyecto y que los riesgos detectados relacionados con las brechas de seguridad pueden evitarse, podemos concluir que el proyecto, es viable desde el punto de vista ético, y además es recomendable implementarlo siguiendo los pilares fundamentales recogidos en códigos como el de la ACM y en legislaciones como el RGPD garantizando la confidencialidad, integridad y privacidad de la información.

La adopción de las tecnologías *Quantum Safe* por parte de empresas y organizaciones puede elevar el estándar general de seguridad para toda la sociedad, ya que permiten asegurar los datos sensibles ante ataques de ordenadores cuánticos. Además, el uso exclusivo de herramientas libres, legales y de acceso abierto refleja un compromiso con la propiedad intelectual, la ética profesional y el fomento de una ciberseguridad responsable y accesible.

7. CONCLUSIONES Y TRABAJOS FUTUROS

Este proyecto de investigación ha evidenciado la necesidad de migrar el protocolo TLS hacia esquemas criptográficos resistentes a ataques cuánticos (Quantum Safe). Se ha demostrado la viabilidad técnica y práctica de establecer conexiones TLS seguras frente a las nuevas amenazas que presentan los ordenadores cuánticos. Además, los tiempos de ejecución obtenidos han sido mejores de lo esperado, lo que respalda la factibilidad de aplicar estos algoritmos en escenarios del mundo real.

Durante la realización del proyecto se ha podido entender tanto los mecanismos tradicionales de criptografía como los postcuánticos, y los esfuerzos de implementación necesarios para la transición. El desarrollo de problemas matemáticos complejos, como los basados en lattices, constituye solo la primera etapa. La investigación matemática da paso a un extenso proceso de estandarización, implementación y validación. Esto es especialmente crítico en el ámbito de la ciberseguridad, donde cualquier error puede comprometer información sensible y acarrear consecuencias graves a la empresa.

Los resultados obtenidos indican que el protocolo TLS Quantum Safe no solo incrementa la seguridad al utilizar algoritmos resistentes a la computación cuántica, sino que también mejora significativamente los tiempos de ejecución frente a algoritmos convencionales. Gracias al análisis de datos experimentales se ha podido generar recomendaciones valiosas para la empresa.

RSA, pese a tener un largo recorrido en el mercado y contar con un respaldo ampliamente reconocido en cuanto a fiabilidad, resulta cada vez menos viable. Se aconseja a las empresas valorar la migración de los protocolos TLS que aún lo utilizan, ya que mantener su seguridad requiere un aumento considerable en el tamaño de las claves. Este crecimiento implica un mayor coste computacional, lo que convierte a RSA en una opción cada vez menos eficiente.

Por otra parte, se ha verificado la superioridad de ML-KEM frente a su predecesor Kyber, destacándose como el algoritmo con mejor rendimiento en los experimentos realizados. Con tiempos de ejecución notablemente inferiores a los del resto de algoritmos evaluados, ML-KEM se posiciona como una de las opciones más prometedoras, especialmente en su implementación híbrida, tal y como ya lo han adoptado actores industriales como Cloudflare o WolfSSL.

En este contexto, también se concluye que la estrategia óptima es una transición gradual y controlada hacia algoritmos híbridos que combinen criptografía moderna con postcuántica. Para ello, las curvas elípticas han demostrado ser los candidatos más eficientes y compatibles para este tipo de integración en algoritmos de encapsulación de clave postcuánticos. Concretamente la combinación de Curvas elípticas con ML-KEM se sitúa como la mejor opción, ya que, en un futuro podría migrarse completamente a ML-KEM sin combinación híbrida. Esto resultaría en tiempos de ejecución sorprendentemente reducidos, manteniendo buenos niveles de seguridad.

No obstante, FrodoKEM sigue siendo una alternativa sólida que no debe ser descartada. Depender exclusivamente de un solo esquema criptográfico puede ser arriesgado; si se descubre una vulnerabilidad o backdoor en dicho esquema, las consecuencias podrían ser catastróficas para la seguridad. Por ello, mantener diversidad en los algoritmos postcuánticos contribuye a la resiliencia del sistema criptográfico.

Por último, se destaca la evolución constante de las nuevas tecnologías. El proyecto utilizó la versión 0.12.0 de LibOQS. Sin embargo, durante el desarrollo de este, se publicó una nueva versión de LibOQS. Asimismo, han surgido novedades relevantes en la criptografía postcuántica, como la reciente publicación de versiones Quantum Safe de OpenSSL (The OpenSSL Project, 2024).

Es por ello, que se debe continuar la investigación a medida que surjan nuevas versiones de algoritmos, se estandaricen nuevos esquemas o se descubran mejoras que reduzcan la complejidad computacional. Dado que los algoritmos postcuánticos están en constante evolución, es fundamental mantener una actitud abierta al cambio. Se recomienda a las empresas adoptar un modelo criptoágil⁷, que permita una transición flexible y eficiente entre algoritmos a medida que el panorama criptográfico evoluciona.

Como trabajos futuros, se propone la integración de estas soluciones en navegadores como Chromium, así como ampliar el análisis para otras aplicaciones criptográficas y protocolos.

⁷ La criptoagilidad, o agilidad criptográfica, se refiere a la capacidad de un sistema para conservar su nivel de seguridad al modificar de forma ágil sus algoritmos criptográficos, métodos de gestión de claves y otros mecanismos asociados, todo ello sin afectar ni interrumpir el funcionamiento general de la infraestructura.

8. OTROS MÉRITOS DEL PROYECTO

En este proyecto, se han sentado las bases para un modelo de estudio de tecnologías Quantum Safe aplicadas al protocolo TLS. Todas las herramientas empleadas son de software libre, y los programas desarrollados para el procesamiento y visualización de datos se han puesto a disposición pública en el repositorio de Github: https://github.com/anruki/Quantum-Safe-Analysis/blob/main/Procesamiento_Visualizacion.ipynb. Cualquier usuario sin necesidad de tener un equipo especializado, puede replicar el experimento y analizar los resultados. Esto es especialmente valioso en un contexto donde los algoritmos están sometidos a actualizaciones constantes y se quiera realizar evaluaciones rutinarias.

Durante el desarrollo, también se obtuvieron conclusiones valiosas sobre el comportamiento de los algoritmos analizados. Ofreciendo una visión clara para que las empresas puedan migrar a TLS Quantum Safe de manera informada, conociendo las fortalezas y debilidades de cada algoritmo disponible.

Además, se han utilizado imágenes del proyecto Open Quantum Safe (OQS), originalmente diseñadas para ejecutarse con Docker. Sin embargo, todas las pruebas se realizaron con Podman como motor de contenedores, lo que requirió una adaptación manual de las guías oficiales debido a diferencias en comandos e instrucciones entre ambos sistemas. Esta adaptación no solo permitió validar la compatibilidad de Podman con herramientas postcuánticas, sino que también constituye una contribución práctica a la documentación no oficial de OQS, facilitando su uso en entornos donde Docker no está disponible o no es la opción preferida.

Por último, cabe destacar el carácter interdisciplinar del proyecto, que abarca desde los fundamentos matemáticos hasta la implementación práctica de protocolos actuales como TLS, incluyendo toda la infraestructura necesaria y la incorporación de criptografía postcuántica. De esta manera, se ha creado un documento unificado que explora todo el proceso de migración a Quantum Safe y sus efectos en términos de rendimiento y seguridad.

9. BIBLIOGRAFÍA

- [1] Vicens, J. (2025). *Informe sobre delitos ciberneticos en Estados Unidos*. Editorial Seguridad Digital.
- [2] Instituto Nacional de Ciberseguridad (INCIBE). (2025). *Balance de ciberseguridad en España 2024*. [En línea]. Available: <https://www.incibe.es/incibe/sala-de-prensa/incibe-presenta-su-balance-de-ciberseguridad-2024-con-mas-de-97000-incidentes>
- [3] P. W. Shor, “Algorithms for quantum computation: discrete logarithms and factoring,” *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124–134.
- [4] National Institute of Standards and Technology (NIST), “Post-Quantum Cryptography Standardization Project,” 2024. [Online]. Available: <https://csrc.nist.gov/Projects/post-quantum-cryptography>
- [5] Dierks, T., & Rescorla, E. (2008). *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246. Internet Engineering Task Force (IETF). <https://doi.org/10.17487/RFC5246>
- [6] Open Quantum Safe Project, “liboqs – C library for quantum-resistant cryptographic algorithms,” 2024. [Online]. Available: <https://openquantumsafe.org/liboqs/>
- [7] Stallings, W. (2017). *Cryptography and Network Security: Principles and Practice* (7th ed.). Pearson.
- [8] Katz, J., & Lindell, Y. (2020). *Introduction to Modern Cryptography* (3rd ed.). CRC Press.
- [9] Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). *Handbook of Applied Cryptography*. CRC Press. Disponible en: <http://cacr.uwaterloo.ca/hac/>
- [10] Diffie, W., & Hellman, M. (1976). *New directions in cryptography*. IEEE Transactions on Information Theory, 22(6), 644–654.
- [11] Shor, P. W. (1997). *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*. SIAM Journal on Computing, 26(5), 1484–1509.
- [12] Rivest, R. L., Shamir, A., & Adleman, L. (1978). *A method for obtaining digital signatures and public-key cryptosystems*. Communications of the ACM, 21(2), 120–126.
- [13] Nielsen, M. A., & Chuang, I. L. (2010). *Quantum Computation and Quantum Information* (10th Anniversary Ed.). Cambridge University Press.
- [14] IBM Quantum. (2023). *What is quantum computing?* IBM Research. [Online] Available: <https://www.ibm.com/quantum-computing/what-is-quantum-computing>
- [15] IBM Quantum. (2023). *IBM Quantum systems*. Disponible en: <https://www.ibm.com/quantum>

- [16] IBM Newsroom. (2023). *IBM Debuts Next-Generation Quantum Processor IBM Quantum System Two, Extends Roadmap to Advance Era of Quantum Utility*. Disponible en: <https://newsroom.ibm.com/2023-12-04-IBM-Debuts-Next-Generation-Quantum-Processor-IBM-Quantum-System-Two,-Extends-Roadmap-to-Advance-Era-of-Quantum-Utility>
- [17] World Economic Forum, “Quantum Computing in the Next Decade: Transformative Applications,” Nov. 2022. [En línea]. Disponible en: [Quantum technology | World Economic Forum](#)
- [18] Qiskit Community, “*Shor’s Algorithm*,” Qiskit Textbook, GitHub repository, [En línea]. Disponible: <https://github.com/qiskit-community/qiskit-textbook/blob/main/content/ch-algorithms/shor.ipynb>
- [19] IBM Quantum, “*Asymmetric Key Cryptography*,” Practical Introduction to Quantum-Safe Cryptography, IBM Quantum Learning, [En línea]. Disponible: <https://learning.quantum.ibm.com/course/practical-introduction-to-quantum-safe-cryptography/asymmetric-key-cryptography>
- [20] Uttam Ghosh, Debasish Das, Pushpita Chatterjee. “A Comprehensive Tutorial on Cybersecurity in Quantum Computing Paradigm”. ResearchGate, 2023. Disponible en: https://www.researchgate.net/publication/369353681_A_Comprehensive_Tutorial_on_Cybersecurity_in_Quantum_Computing_Paradigm
- [21] National Institute of Standards and Technology (NIST), “*Post-Quantum Cryptography*,” NIST, [En línea]. Disponible: <https://csrc.nist.gov/Projects/post-quantum-cryptography>
- [22] D. J. Bernstein and T. Lange, “Post-quantum cryptography,” Nature, vol. 549, no. 7671, pp. 188–194, Sep. 2017. [En línea]. Disponible en: <https://www.nature.com/articles/nature23461>
- [23] A. Menezes, “Lecture 4. Lattices (The Mathematics of Lattice-Based Cryptography),” YouTube, 2025. Disponible en: https://www.youtube.com/watch?v=zPa1h-gj8_A
- [24] Chalk Talk, “Lattice-based cryptography: The tricky math of dots,” YouTube, 2022. [En línea]. Disponible: <https://www.youtube.com/watch?v=QDdOoYdb748&t=355s>
- [25] Y. Aono, Y. Hayashi, and H. Yamamoto, “Quantum Speedup in Shortest Vector Problem via Sieve Algorithm,” in Proc. IEEE ISIT, 2018. [En línea]. Disponible: <https://arxiv.org/abs/1804.05044>
- [26] Y. Bai, L. Ducas, and W. van Woerden, “Improved Classical and Quantum Algorithms for the Shortest Vector Problem via Tensor Locality-Sensitive Hashing,” *arXiv preprint*, arXiv:2012.04649, 2020. [En línea]. Disponible: <https://arxiv.org/abs/2012.04649>
- [27] O. Goldreich, S. Goldwasser, y S. Halevi, “On the Security of the Goldreich–Goldwasser–Halevi Cryptosystem,” *Journal of Cryptology*, vol. 12, no. 1, pp. 1–30, 1999.
- [28] D. Micciancio y S. Goldwasser, *Complexity of Lattice Problems: A Cryptographic Perspective*, Springer, 2002.

- [29] J. Hoffstein, J. Pipher, y J.H. Silverman, *An Introduction to Mathematical Cryptography*, Springer, 2008.
- [30] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography,” *Journal of the ACM*, vol. 56, no. 6, 2009.
- [31] C. Peikert, “A Decade of Lattice Cryptography,” *Foundations and Trends® in Theoretical Computer Science*, vol. 10, no. 4, pp. 283–424, 2016.
- [32] A. Menezes, “Lecture 3. Learning With Errors (LWE) Problem (The Mathematics of Lattice-Based Cryptography),” YouTube, 2025. [En línea]. Disponible: <https://www.youtube.com/watch?v=M1cq7cuonbl&t=1322s>
- [33] Chalk Talk, “Learning with errors: Encrypting with unsolvable equations,” YouTube, 2023. [En línea]. Disponible: <https://www.youtube.com/watch?v=K026C5YaB3A>
- [34] National Institute of Standards and Technology, “NIST announces first four quantum-resistant cryptographic algorithms,” NIST, Jul. 5, 2022. [Online]. Available: <https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms>
- [35] National Institute of Standards and Technology, “CRYSTALS-Kyber Algorithm Specifications and Supporting Documentation,” NIST PQC Project, 2022. [Online]. Available: <https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022>
- [36] A&D Solutions, “Entendiendo CRYSTALS-Kyber,” A&D Solutions, [Online]. Available: <https://aandds.com/blog/kyber.html#8a760809>
- [37] L. E. Locascio, “Statement on quantum-resistant algorithms,” NIST, 2022. [Online]. Available: <https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms>
- [38] Internet Engineering Task Force (IETF), “The Transport Layer Security (TLS) Protocol Version 1.3,” RFC 8446, Aug. 2018. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc8446>
- [39] I. Grigorik, “Transport Layer Security (TLS),” *High Performance Browser Networking*, 2013. [Online]. Available: <https://hpbn.co/transport-layer-security-tls/>
- [40] A. Langley, “Overclocking SSL,” *ImperialViolet Blog*, Jun. 25, 2010. [Online]. Available: <https://www.imperialviolet.org/2010/06/25/overclocking-ssl.html>
- [41] D. Beaver, “Scaling the Facebook HTTPS infrastructure,” *IETF HTTP Working Group Archives*, Sep. 20, 2012. [Online]. Available: <https://lists.w3.org/Archives/Public/ietf-http-wg/2012JulSep/0251.html>
- [42] X Developer Documentation, “TLS Requirements,” *X Platform*, 2024. [Online]. Available: <https://docs.x.com/resources/fundamentals/authentication/guides/tls>
- [43] IBM. *Transport Layer Security (TLS) concepts*. IBM Knowledge Center. Disponible en: <https://www.ibm.com/docs/en/ibm-mq/9.3.x?topic=tls-transport-layer-security-concepts> (consultado el 2 de julio de 2025).

- [44] Microsoft. *Networking concepts: SSL/TLS for Azure Database for PostgreSQL Flexible Server*. Microsoft Learn. Disponible en: <https://learn.microsoft.com/es-es/azure/postgresql/flexible-server/concepts-networking-ssl-tls> (consultado el 2 de julio de 2025).
- [45] OpenSSL Project, "OpenSSL: The Open Source toolkit for SSL/TLS," Disponible en: <https://www.openssl.org/>
- [46] H. Choi, S. Matsuo, Y. Kato, Y. Kubo, T. Iwata, y T. Okamoto, "Post-quantum public key algorithms selected in liboqs," *Proceedings of the SCIS 2018 Symposium*, Kanazawa, Japón, 2018. Disponible en: https://caislab.kaist.ac.kr/publication/paper_files/2018/SCIS%2718_HC_SCA.pdf
- [47] F. Borges, P. R. Reis y D. Pereira, "A Comparison of Security and its Performance for Key Agreements in Post-Quantum Cryptography," *IEEE Access*, vol. 8, pp. 142413–142422, 2020, doi: 10.1109/ACCESS.2020.3013250. Disponible en: <https://ieeexplore.ieee.org/document/9153901>
- [48] B. K. Meles, *Implementation and Performance Evaluation of Quantum-Safe Cryptography in C: A Comparative Study Using LibOQS and OpenSSL*, Bachelor's thesis, Dept. of Information Engineering, University of Padua, 2024. [Online]. Available: <https://thesis.unipd.it/handle/20.500.12608/71298>
- [49] Open Quantum Safe Project, "Benchmarking," *Open Quantum Safe*, <https://openquantumsafe.org/benchmarking/> (accedido el 15 de abril de 2025).
- [50] W. W. Royce, "Managing the development of large software systems," *Proceedings of IEEE WESCON*, vol. 26, no. 8, pp. 1–9, 1970.
- [51] IBM Quantum Learning, "Practical Introduction to Quantum-Safe Cryptography," curso en línea, 2023. [En línea]. Disponible: <https://learning.quantum.ibm.com/course/practical-introduction-to-quantum-safe-cryptography>
- [52] The Linux Foundation. (2024). *What is Linux?* Recuperado de <https://www.linuxfoundation.org/what-is-linux/>
- [53] BasicFundas, "Linux Statistics 2024: Market Share, Usage Trends, and Website Traffic Analysis," BasicFundas, 2024. [En línea]. Disponible en: <https://basicfundas.com/linux-statistics-2024-market-share-usage-trends-and-website-traffic-analysis/>
- [54] IBM, "What Are Containers?," IBM Cloud Learn Hub, 2024. [En línea]. Disponible en: <https://www.ibm.com/think/topics/containers>
- [55] Business Research Insights, "Container Technology Market Size, Share & COVID-19 Impact Analysis," 2024. [Online]. Available: <https://www.businessresearchinsights.com/market-reports/container-technology-market-103580>
- [56] Open Quantum Safe, "openquantumsafe/curl - Docker Image," *Docker Hub*. [En línea]. Disponible en: <https://hub.docker.com/r/openquantumsafe/curl>.

- [57] Open Quantum Safe, “openquantumsafe/nginx - Docker Image,” *Docker Hub*. [En línea]. Disponible en: <https://hub.docker.com/r/openquantumsafe/nginx>.
- [58] Open Quantum Safe, *oqs-demos: Demos using liboqs, oqs-provider, and post-quantum cryptography*, GitHub, 2025. [En línea]. Disponible en: <https://github.com/open-quantum-safe/oqs-demos>.
- [59] J. P. Hayes, “What can be done to prepare for post quantum cryptography?,” *EE World Online*, 2023. [En línea]. Disponible en: <https://www.eeworldonline.com/what-can-be-done-to-prepare-for-post-quantum-cryptography/>.
- [60] NGINX, *What is NGINX?*, F5, 2024. [En línea]. Disponible en: <https://www.nginx.com/resources/glossary/nginx/>
- [61] Curl, *Everything curl*, 2025. [En línea]. Disponible en: <https://everything.curl.dev/>
- [62] Built in C, NGINX es uno de los servidores web más utilizados en la actualidad, siendo utilizado por sitios como Facebook, Netflix, Dropbox y Spotify, entre otros
- [63] GeeksforGeeks, “Client-Server Architecture,” *GeeksforGeeks*, 2022. [En línea]. Disponible en: <https://media.geeksforgeeks.org/wp-content/uploads/20220721163106/ClientComputer22.png>
- [64] IBM, "Practical Introduction to Quantum-Safe Cryptography," IBM Quantum Learning, [En línea]. Disponible en: <https://learning.quantum.ibm.com/course/practical-introduction-to-quantum-safe-cryptography/quantum-safe-cryptography>.
- [65] NIST, *Transitioning the Use of Cryptographic Algorithms and Key Lengths*, NIST Special Publication 800-131A Rev. 2, Mar. 2019. [Online]. Available: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar2.pdf>
- [66] W. Diffie and M. Hellman, “New Directions in Cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.
- [67] M. D. Baushke, “Key Exchange (KEX) Method Updates and Recommendations for Secure Shell (SSH),” *Internet-Draft draft-ietf-curdle-ssh-kex-sha2-13*, IETF, Jan. 14 2021, expires Jul. 18 2021. [Online]. Available: <https://www.ietf.org/archive/id/draft-ietf-curdle-ssh-kex-sha2-13.html>
- [68] NIST, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)*, NIST Special Publication 800-56A Rev. 3, Apr. 2018. [Online]. Available: <https://doi.org/10.6028/NIST.SP.800-56Ar3>
- [69] E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3,” *RFC 8446*, Aug. 2018. [Online]. Available: <https://doi.org/10.17487/RFC8446>
- [70] D. J. Bernstein, M. Hamburg, A. Krasnova y T. Lange, “Elliptic Curve Diffie-Hellman (ECDH) Key Agreement Scheme”, RFC 7748, febrero 2016. <https://datatracker.ietf.org/doc/html/rfc7748>
- [71] E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3,” *RFC 8446*, Aug. 2018. [Online]. Available: <https://doi.org/10.17487/RFC8446>

- [72] secunet Security Networks AG, "Elliptic Curve Cryptography Made in Germany," secunet.com, 2015. [Online]. Available: <https://www.secunet.com/en/about-us/news-events/article/elliptic-curve-cryptography-made-in-germany-1>
- [73] J. Merkle y M. Lochter, "Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation", RFC 5639, marzo 2010. [Enlace]datatracker.ietf.org
- [74] E. Rescorla, "Hybrid Public Key Exchange in TLS 1.3," IETF Internet-Draft [draft-ietf-tls-hybrid-design-13](https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design-13), 2023. [En línea]. Disponible en: <https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/>
- [75] E. Barker, "Recommendation for Key Management: Part 1 – General," *NIST Special Publication 800-57 Revision 5*, mayo 2020. Disponible en: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf>
- [76] Spiceworks, "Microsoft deprecates 1024-bit RSA keys in Windows," *Spiceworks*, 2016. Disponible en: <https://www.spiceworks.com/it-security/endpoint-security/news/microsoft-deprecates-1024-bit-rsa-keys-windows/>
- [77] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [78] D. Boneh, "Twenty years of attacks on the RSA cryptosystem," *Notices of the AMS*, vol. 46, no. 2, pp. 203–213, 1999.
- [79] Cloudflare, "Post-Quantum for All: TLS Now Supports Kyber," 2022. [Online]. Available: <https://blog.cloudflare.com/post-quantum-for-all/>
- [80] Open Quantum Safe Project, "Kyber is being deprecated in favour of ML-KEM," GitHub Discussion, May 2024. [Online]. Available: <https://github.com/open-quantum-safe/liboqs/releases/tag/0.12.0>
- [81] National Institute of Standards and Technology (NIST), "FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard," Aug. 13, 2024. [Online]. Available: NIST FIPS 203 [nist.gov+15csrc.nist.gov+15csrc.nist.gov+15](https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.pdf)
- [82] S. Fluhrer, "ML-KEM Security Considerations," *Internet Engineering Task Force (IETF)*, draft-sfluhrer-cfrg-ml-kem-security-considerations-01, Work in Progress, Jan. 2024. [Online]. Available: <https://www.ietf.org/archive/id/draft-sfluhrer-cfrg-ml-kem-security-considerations-01.html>
- [83] wolfSSL, "Post-Quantum Cryptography Support," 2024. [Online]. Available: <https://www.wolfssl.com/docs/post-quantum/>
- [84] IBM / Taurus SA, "The Post-Quantum Technology Landscape," 2023. [Online]. Available: <https://www.taurushq.com/blog/quantum-doomsday-planning-2-2-the-post-quantum-technology-landscape/>
- [85] FrodoKEM Project, "FrodoKEM homepage," *FrodoKEM.org*. [Online]. Available: <https://frodkem.org/>

- [86] A. Misoczki et al., "BIKE: Bit Flipping Key Encapsulation," NIST PQC Submission, 2022. [Online]. Available: <https://bikesuite.org/>
- [87] BIKE Suite Project, "*BIKE: Bit Flipping Key Encapsulation (Version 5.2)*," Oct. 10, 2024. [Online]. Available: https://bikesuite.org/files/v5.2/BIKE_Spec.2024.10.10.1.pdf
- [88] P. Gaborit, G. Zémor, et al., "HQC: Hamming Quasi-Cyclic," NIST PQC Submission, 2022. [Online]. Available: <https://pqc-hqc.org/>
- [89] C. Aguilar-Melchor et al., "Hamming Quasi-Cyclic (HQC)," NIST PQC Seminar, Sept. 20, 2024, National Institute of Standards and Technology, Gaithersburg, MD. [Online]. Available: <https://csrc.nist.gov/projects/post-quantum-cryptography>
- [90] Association for Computing Machinery, "ACM – Association for Computing Machinery," [En línea]. Disponible en: <https://www.acm.org/>. [Accedido: 13-abr-2025].
- [91] DataScientest, "Deep Blue: todo sobre el ordenador que venció a Kasparov", <https://datascientest.com/es/deep-blue-todo-sobre> [Accedido: 13 de abril de 2025].
- [92] Association for Computing Machinery, "ACM Code of Ethics and Professional Conduct," 2018. [En línea]. Disponible en: <https://www.acm.org/code-of-ethics>. [Accedido: 13-abr-2025].
- [93] IBM, "Sostenibilidad con LinuxONE," IBM, [En línea]. Disponible en: <https://www.ibm.com/es-es/products/linuxone-4/sustainability>. [Accedido: 9-abr-2025].
- [94] Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos (Reglamento General de Protección de Datos - RGPD), 2016. Disponible en: <https://eur-lex.europa.eu/ES/legal-content/summary/general-data-protection-regulation-gdpr.html> [Accedido: 10-abr-2025]
- [95] Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales, Boletín Oficial del Estado (BOE), 6 de diciembre de 2018. Disponible en: <https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673> . [Accedido: 10-abr-2025]
- [96] Ley 36/2015, "Ley de Seguridad Nacional," Boletín Oficial del Estado, 2015.
- [97] Real Decreto 43/2021, "Real Decreto sobre la seguridad de las redes y sistemas de información," Boletín Oficial del Estado, 2021.
- [98] Código de Derecho de la Ciberseguridad, "Código de Derecho de la Ciberseguridad," Disponible en: https://www.boe.es/biblioteca_juridica/codigos/codigo.php?id=173&modo=2¬a=0&tab=2 . [Accedido: 10-abr-2025].
- [99] Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia. Disponible en: <https://www.boe.es/buscar/pdf/1996/BOE-A-1996-8930-consolidado.pdf> . [Accedido: 10-abr-2025].

[100] Ley 16/1993, de 23 de diciembre, de incorporación al Derecho español de la Directiva 91/250/CEE, de 14 de mayo de 1991, sobre la protección jurídica de programas de ordenador. Disponible en: <https://www.boe.es/buscar/doc.php?id=BOE-A-1993-30621> . [Accedido: 10-abr-2025].

[101] INCIBE, “Una empresa se infecta de ransomware que se propaga al conectar dispositivos”, Instituto Nacional de Ciberseguridad (INCIBE), [en línea]. Disponible en: <https://www.incibe.es/linea-de-ayuda-en-ciberseguridad/casos-reales/una-empresa-se-infecta-de-ransomware-que-se-propaga-al-conectar-dispositivos> . [Accedido: 13-abr-2025].

ANEXO A: LICENCIAS DE SOFTWARE

- Ubuntu: <https://ubuntu.com/legal>
- Podman (Apache 2.0): <https://www.apache.org/licenses/LICENSE-2.0>
- LibOQS (MIT): <https://opensource.org/licenses/MIT>

ANEXO B: DESPLIEGUE INFRAESTRUCTURA

Guía de instalación de las tecnologías utilizadas:

Instalación y configuración de entorno Linux mediante WSL2 y Podman en Windows

Este apartado describe el proceso de instalación y configuración de un entorno Linux funcional dentro de Windows mediante el uso de WSL 2 (Subsistema de Windows para Linux) y la herramienta Podman para gestión de contenedores. Esta solución permite trabajar con herramientas propias de Linux sin necesidad de máquinas virtuales completas, optimizando recursos y mejorando la integración con el sistema operativo anfitrión.

1. Instalación de WSL2 en Windows



WSL 2 es una máquina virtual ligera integrada en Windows que permite hostear otras distribuciones Linux con un rendimiento notablemente superior al de su versión anterior (WSL 1). Su arquitectura basada en un kernel Linux real ofrece mayor compatibilidad con herramientas nativas de Linux.

Se ha descargado WSL 2 ejecutando comandos desde la terminal PowerShell de Windows (abierta con permisos de administrador).

Comando de instalación:

```
wsl --install
```

Ejecutar este comando en PowerShell de Windows habilitará las características necesarias para ejecutar WSL e instala:

- El subsistema de Windows para Linux (WSL)
- WSL 2 como versión predeterminada
- La distribución de Ubuntu (última versión estable)



2. Despliegue de Ubuntu

Durante la instalación de WSL 2, se ha desplegado automáticamente una distribución de Linux, concretamente Ubuntu 22.04 LTS, ofreciendo así un entorno Linux completo sin recurrir a soluciones como Hyper-V, VirtualBox o VMware.

En caso de que la distribución no se instale automáticamente, puede descargarse e instalarse desde la Microsoft Store mediante el siguiente enlace:

[Get Ubuntu | Download | Ubuntu](#)

3. Instalar Podman



Podman es una herramienta para la gestión de contenedores compatible con Docker, pero que no requiere un servicio (daemon) en segundo plano, lo que la convierte en una opción ligera e ideal para entornos WSL 2.

La instalación se realiza desde la terminal de Ubuntu con el siguiente comando:

```
sudo apt-get update  
sudo apt-get -y install podman
```

Requerimientos de Podman en WSL 2

Sistema Operativo:

- ✓ Windows 10, versión 1903 o superior, con WSL 2 activado.

Dependencias:

- ✓ WSL 2 requiere que estén habilitadas las características Hyper-V y Virtual Machine Platform.
- ✓ Es necesario contar con una distribución de Linux (en este caso, Ubuntu 22.04) dentro de WSL.

Requisitos de Hardware:

- ✓ Una máquina con al menos 2 GB de RAM y suficiente espacio en disco.

Estos requerimientos ya se han completado en pasos anteriores, y se cumplen para un Thinkpad Intel core i5.

4. Descargar contenedores libOQS

En el marco de este trabajo se han utilizado contenedores que integran bibliotecas criptográficas resistentes a la computación cuántica, proporcionadas por el proyecto Open

Quantum Safe (OQS). Estos contenedores están basados en imágenes de NGINX y curl, modificadas específicamente para soportar algoritmos de cifrado post-cuántico mediante la biblioteca liboqs.

Las imágenes oficiales se encuentran disponibles en Docker Hub:

- [openquantumsafe/curl](#)
- [openquantumsafe/nginx](#)

Estas imágenes permiten experimentar con conexiones TLS que emplean algoritmos híbridos y cuántico-resistentes definidos por el estándar TLS 1.3 y adaptados mediante OpenSSL + liboqs.

Para obtener estas imágenes y usarlas en el entorno WSL2 con Podman, se deben seguir los siguientes pasos desde la terminal de Ubuntu:

```
> podman pull docker.io/openquantumsafe/curl  
> podman pull docker.io/openquantumsafe/nginx
```

Podman, a diferencia de Docker, no asume automáticamente el registro predeterminado docker.io por lo que es importante usar el nombre completo del repositorio, incluyendo el prefijo docker.io/ para garantizar que Podman busque la imagen directamente en Docker Hub.

5. Levantar el servidor NGINX post-cuántico

Ejecutar el siguiente comando para iniciar un contenedor con la imagen modificada de NGINX, exponiendo el puerto 4433:

```
podman run -d -p 4433:4433 docker.io/openquantumsafe/nginx
```

Esto inicia un servidor HTTPS accesible en: <https://localhost:4433>

6. Abrir la terminal interactiva de curl post-cuántico

Para ejecutar comandos dentro del contenedor curl con soporte para algoritmos cuánticos, se utiliza:

```
podman run -it docker.io/openquantumsafe/curl
```

Realizar la conexión TLS usando un algoritmo post-cuántico

Dentro del contenedor curl, se ejecuta:

```
curl --curves kyber768 https://localhost:4433
```

En este comando, curl utiliza el algoritmo de intercambio de claves Kyber768, que forma parte de la criptografía post-cuántica, para negociar la conexión TLS. La URL <https://localhost:4433> corresponde al servidor NGINX post-cuántico previamente iniciado.

ANEXO C: TABLAS DE RENDIMIENTOS

Las tablas de rendimiento obtenidas tras la ejecución de los algoritmos se encuentran disponibles en:

<https://github.com/anruki/Quantum-Safe-Analysis/blob/main/rendimientos.xlsx>

ANEXO D: CÓDIGO PARA EL PROCESAMIENTO DE DATOS Y VISUALIZACIÓN

El código creado para el procesamiento de los datos obtenidos y su posterior visualización se encuentra disponible en:

https://github.com/anruki/Quantum-Safe-Analysis/blob/main/Procesamiento_Visualizacion.ipynb

ANEXO E: TABLAS DE RENDIMIENTO PROCESADAS

Las tablas de rendimiento obtenidas tras el procesamiento se encuentran disponibles en:

https://github.com/anruki/Quantum-Safe-Analysis/blob/main/rendimientos_KEM_KEX_processed.xlsx

ANEXO F: PODMAN

Podman es una herramienta para gestionar contenedores y pods en sistemas Linux. A diferencia de Docker, Podman no utiliza un daemon en segundo plano, sino que se ejecuta como un proceso sin privilegios, lo que mejora la seguridad y facilita su integración con entornos de usuario sin acceso root.

Aunque Podman es compatible con el formato de contenedores de Docker y comparte muchas órdenes comunes, existen diferencias importantes en su funcionamiento. A diferencia de Docker, Podman no asume un registro predeterminado, lo que puede requerir ajustes al especificar imágenes. Además, algunos comandos y scripts diseñados para Docker pueden necesitar pequeñas modificaciones. Al no depender de un daemon, Podman permite la ejecución sin privilegios, lo que lo hace ideal para entornos más seguros. También ofrece compatibilidad con la CLI de Docker para facilitar la transición.

En este proyecto se utilizaron imágenes del proyecto Open Quantum Safe (OQS), originalmente diseñadas para ser ejecutadas con Docker. Sin embargo, todas las pruebas se llevaron a cabo utilizando Podman como motor de contenedores. Esto requirió una adaptación manual de las guías oficiales, ya que algunas instrucciones y comandos no funcionan de manera idéntica en Podman.

Esta adaptación no solo permitió validar la compatibilidad de Podman con herramientas post-cuánticas, sino que también aporta una contribución práctica a la documentación no oficial de OQS, facilitando su uso en entornos donde Docker no está disponible o no es deseado.

Como conclusión, se comprobó que Podman puede utilizarse satisfactoriamente como sustituto de Docker para ejecutar herramientas criptográficas post-cuánticas, ofreciendo además ventajas adicionales como la ejecución sin privilegios de root y un modelo de seguridad más estricto.