Universidad de Las Américas

Facultad de Ingeniería y Ciencias Aplicadas

Ingeniería de Ciberseguridad

Implementación de un sistema de detección de phishing basado en Machine Learning para la mitigación de amenazas en flujos de correo electrónico mediante el protocolo IMAP para la Empresa HACKICK - IT Services & Consulting

Anderson Alejandro Freire Ortega

Geovani Alejandro Osorio Salazar

Fecha

Quito, Ecuador

Contenido

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

Resumen

Abstract

[Esta sección incluye el resumen en idioma inglés.]

1. Introducción

1.1. Identificación y descripción del problema o necesidad

En la actualidad, el phishing se ha consolidado como una de las amenazas más persistentes y efectivas dentro del panorama de la ciberseguridad. Esta amenaza ha experimentado una metamorfosis significativa, evolucionando desde campañas masivas genéricas hacia ataques de precisión que utilizan modelos de lenguaje de gran escala (LLM) y redacción asistida por inteligencia artificial para generar mensajes altamente personalizados que evaden con facilidad los mecanismos de detección estáticos, tal como señalan Weinz et al. (2025). Investigaciones recientes subrayan que esta sofisticación tecnológica permite a los atacantes superar las barreras tradicionales, incrementando significativamente la probabilidad de éxito en la suplantación de identidades (Zhang et al., 2024; Liu et al., 2024).

Para una empresa de servicios tecnológicos y consultoría como Hackick – IT Services & Consulting, el impacto del phishing adquiere una dimensión crítica debido al intercambio constante de propuestas comerciales, reportes técnicos y documentación confidencial con aliados estratégicos. A pesar de la existencia de controles tradicionales como filtros antispam y listas negras, estas soluciones resultan insuficientes frente a campañas modernas que emplean dominios legítimos comprometidos y enlaces ofuscados (Gupta et al., 2015). Esta vulnerabilidad técnica es especialmente aguda en el sector de las pequeñas y medianas empresas (PYMES), mercado objetivo de Hackick, las cuales carecen a menudo de presupuestos para adquirir herramientas comerciales avanzadas (Applied Sciences, 2025).

Frente a este escenario, surge la necesidad de desarrollar un sistema de detección automatizado que utilice técnicas de machine learning para identificar características estructurales, léxicas y semánticas de correos fraudulentos (Abbazi & Alsabih, 2024). La propuesta se centra en la creación de una arquitectura tipo Sidecar bajo el protocolo IMAP, implementada en lenguaje Go para la gestión de conexiones y Python para la lógica de clasificación. Este enfoque no solo mitiga los riesgos operativos y económicos asociados al fraude digital, sino que posiciona a la organización a la vanguardia en el uso de tecnologías emergentes para la protección de la información.

1.2. Descripción de la organización

Hackick – IT Services & Consulting es una empresa especializada en la prestación de servicios de tecnologías de la información y ciberseguridad, orientada a brindar soluciones de consultoría, soporte técnico, evaluación de riesgos y fortalecimiento de la seguridad digital. Su portafolio de servicios está dirigido principalmente a pequeñas y medianas empresas (PYMES) y a organizaciones que buscan mejorar su postura frente a las amenazas cibernéticas actuales. La empresa se caracteriza por su enfoque en la adopción de buenas prácticas, el uso de tecnologías modernas y el diseño de soluciones ajustadas a las necesidades específicas de cada cliente.

Ilustración 1 Stitio web de la empresa Hackick ® “www.hackick.net”

La misión de Hackick es apoyar a las organizaciones en la protección de sus activos de información mediante servicios confiables, eficientes y técnicamente sólidos, contribuyendo a la reducción de riesgos y a la continuidad operativa de sus procesos tecnológicos. En concordancia con ello, la visión de la empresa es consolidarse como un referente en servicios de ciberseguridad y consultoría TI, reconocido por la calidad de sus soluciones, su enfoque preventivo y su capacidad de adaptación a un entorno tecnológico en constante evolución.

Ilustración 2 Servicios de ciberseguridad ofrecidos por Hackick ®

Hackick opera en un entorno altamente digitalizado, en el cual el correo electrónico constituye uno de los principales canales de comunicación para la gestión comercial, la atención a clientes, el intercambio de información técnica y la coordinación de servicios. A través de este medio se transmiten propuestas comerciales, reportes de auditoría, credenciales temporales, enlaces a plataformas de gestión y documentación sensible, lo que convierte al correo electrónico en un activo crítico desde la perspectiva de la seguridad de la información.

Desde el ámbito tecnológico, la empresa emplea tanto herramientas de código abierto como soluciones comerciales para la gestión de infraestructura TI, el monitoreo y la seguridad. Su enfoque prioriza arquitecturas flexibles que puedan integrarse con los distintos entornos tecnológicos de sus clientes. Hackick no centraliza los servicios de correo electrónico de las organizaciones que atiende, sino que trabaja sobre infraestructuras heterogéneas, incluyendo servicios en la nube y entornos híbridos, lo cual representa un desafío adicional al momento de implementar controles de seguridad homogéneos frente a amenazas como el phishing.

En este contexto, el desarrollo de un sistema de detección de correos electrónicos de phishing basado en técnicas de aprendizaje automático se presenta como una oportunidad estratégica para la empresa. Esta iniciativa permitiría fortalecer la seguridad interna de Hackick y, al mismo tiempo, habilitar la creación de un nuevo servicio de valor añadido para sus clientes, alineado con la misión de la organización y con las tendencias actuales del mercado de la ciberseguridad.

1.3. Impacto del proyecto en la sociedad

El desarrollo e implementación de un sistema de detección de correos electrónicos de phishing basado en machine learning en la empresa Hackick – IT Services & Consulting genera un impacto positivo que trasciende el ámbito técnico, aportando beneficios significativos en los planos social, económico, organizacional y ético.

Desde una perspectiva social, el proyecto contribuye a la protección de la información digital de organizaciones que, por limitaciones técnicas o presupuestarias, no cuentan con mecanismos avanzados de detección de amenazas. Al ofrecer una solución accesible y adaptable a pequeñas y medianas empresas, se reduce la exposición de usuarios finales a ataques de ingeniería social, disminuyendo el riesgo de robo de identidad, fraudes financieros y uso indebido de información personal. Esta protección es vital considerando que las nuevas amenazas generadas por inteligencia artificial (LLM) dificultan significativamente la detección manual para el usuario promedio (Weinz et al., 2025). De esta manera, se fortalece la confianza de las personas en el uso seguro del correo electrónico como medio de comunicación digital.

En el ámbito organizacional y económico, la solución propuesta permite a Hackick y a sus clientes mitigar los costos asociados a incidentes de phishing, tales como interrupciones operativas, pérdida de credenciales y daños reputacionales. La detección temprana contribuye a la continuidad del negocio, especialmente dado que los filtros tradicionales basados en reglas estáticas resultan a menudo insuficientes frente a campañas modernas con enlaces ofuscados (Gupta et al., 2015).

Adicionalmente, el proyecto impulsa la adopción de tecnologías emergentes como el machine learning dentro del ecosistema empresarial local, promoviendo el desarrollo de capacidades técnicas en ciberseguridad. Al posicionar esta solución como un servicio de valor añadido, Hackick fomenta la transferencia de conocimiento hacia sus clientes, contribuyendo a la madurez en la gestión de riesgos digitales (Abbazi & Alsabih, 2024).

Desde el punto de vista ético, el proyecto incorpora principios de privacidad, ya que el análisis de correos se realiza sobre datos anonimizados en entornos controlados. Asimismo, la solución no busca sustituir el criterio humano, sino complementarlo mediante alertas que apoyen la toma de decisiones informadas. Este enfoque es fundamental debido a la creciente dificultad que enfrentan los seres humanos para distinguir correos fraudulentos de alta sofisticación (Singh et al., 2020).

2. Análisis de posibles soluciones

2.1. Descripción de estudios realizados

El análisis de posibles soluciones se desarrolla con el objetivo de identificar el enfoque más adecuado para mitigar el riesgo de ataques de phishing en el contexto de la empresa Hackick – IT Services & Consulting. Diversos estudios académicos respaldan la insuficiencia de los métodos tradicionales de detección basados únicamente en reglas estáticas o listas negras, especialmente frente a ataques sofisticados que emplean inteligencia artificial para evadir firmas digitales básicas. Investigaciones recientes han demostrado la eficacia del uso de machine learning para la detección de phishing, ya que estos métodos permiten automatizar la clasificación de correos según patrones complejos aprendidos de datos etiquetados, mejorando la adaptabilidad del sistema ante nuevas variantes de ataques (Muriithi & Karani, 2024; Raweia & Abdulhammed, 2025).

En particular, la literatura revisada señala que algoritmos como Support Vector Machine (SVM) y Random Forest (RF) han demostrado capacidades sólidas para diferenciar entre correos legítimos y maliciosos, alcanzando niveles de precisión elevados cuando se combinan con técnicas de extracción de características relevantes (Raweia & Abdulhammed, 2025). Además, trabajos adicionales muestran que las mejoras en la selección de características, como el uso de técnicas de reducción dimensional y análisis semántico, contribuyen significativamente al desempeño de los modelos al optimizar la calidad de los datos ingresados al algoritmo de clasificación (Daniel et al., 2025).

Desde el punto de vista industrial, los proveedores de seguridad líderes han integrado técnicas de aprendizaje automático en sus plataformas de protección. Por ejemplo, Microsoft Defender for Office 365 incorpora análisis basado en IA para evaluar la reputación de remitentes (Microsoft, 2023), mientras que Proofpoint Email Protection emplea análisis de comportamiento para identificar correos fraudulentos (Proofpoint, 2023). Asimismo, Cisco Secure Email Gateway utiliza modelos adaptativos (Cisco, 2023) y Mimecast Messaging Security aplica análisis contextual dinámico (Mimecast, 2023). Estas implementaciones confirman la madurez de los enfoques basados en aprendizaje automático en entornos corporativos reales.

La investigación respalda la pertinencia de desarrollar una solución propia. La propuesta se diferencia al proponer una arquitectura híbrida: el uso de Go para garantizar una conexión IMAP de alta concurrencia y bajo consumo de recursos, y la integración de Python para la ejecución de modelos de clasificación entrenados con algoritmos de vanguardia.

2.2. Limitaciones y restricciones del proyecto

Limitaciones

- **Alcance de detección:** El sistema se enfocará exclusivamente en la identificación de correos electrónicos de tipo *phishing*, sin abarcar otros vectores de ataque como *malware* o *ransomware*.
- **Entorno de despliegue:** No se contempla el desarrollo de aplicaciones móviles ni extensiones de navegador. El modelo será implementado y validado en un entorno local utilizando el lenguaje **Python** y la librería **scikit-learn** para la lógica de aprendizaje automático.

Restricciones

- **Gestión y limpieza de datos:** El modelo requiere datos estructurados y normalizados en formato CSV. La responsabilidad de la extracción, limpieza, carga y actualización de la información (proceso ETL) recae exclusivamente en el equipo de investigación, utilizando las fuentes abiertas previamente identificadas.
- **Soberanía tecnológica:** Se empleará únicamente *software* libre y herramientas de código abierto, eliminando cualquier dependencia de servicios comerciales o licenciamientos de pago.
- **Marco legal y privacidad:** El tratamiento de la información deberá alinearse estrictamente con la Ley Orgánica de Protección de Datos Personales (LOPDP) del Ecuador, garantizando el principio de anonimización para que ningún dato permita identificar a personas naturales, cumpliendo así con los derechos de protección de datos de carácter personal consagrados en la normativa nacional (Asamblea Nacional del Ecuador, 2021).
- **Infraestructura:** La ejecución será estrictamente *on-premise*, utilizando recursos de computación locales de capacidad media, sin recurrir a procesamiento distribuido o servicios en la nube.

2.3. Identificación y selección de la mejor solución

Con base en el problema identificado y las restricciones técnicas establecidas, se evaluaron tres alternativas estratégicas para mitigar el riesgo de phishing en el entorno de Hackick – IT Services & Consulting. La primera opción consiste en la Implementación de un sistema de detección Sidecar basado en Machine Learning, el cual propone una arquitectura no intrusiva que utiliza el protocolo IMAP para el análisis de flujos de mensajería en paralelo, garantizando la soberanía tecnológica mediante el uso de Go para la infraestructura y Python para la clasificación (Anexo 1). La segunda alternativa evaluada es la Contratación de un proveedor externo especializado en

ciberseguridad, enfocada en la adopción de soluciones comerciales de tipo SaaS que, si bien ofrecen madurez técnica, implican altos costos de licenciamiento y una fuerte dependencia tecnológica de terceros (Anexo 2). Finalmente, se analizó el Desarrollo de una plataforma de concientización antiphishing, orientada a fortalecer el factor humano mediante simulaciones de ataques controlados, aunque esta solución se considera complementaria al no ofrecer una detección automatizada proactiva (Anexo 3).

2.3.1. Implementación de un sistema de detección sidecar.

Esta alternativa propone el desarrollo de un sistema de detección basado en una arquitectura tipo Sidecar por copia, la cual opera de forma paralela al flujo principal de entrega de mensajería sin interferir en la disponibilidad del servicio. En este modelo, cada mensaje entrante al servidor de correo genera una copia automática enviada a un buzón técnico dedicado para su análisis. Este enfoque de "vigilancia pasiva" permite la aplicación de modelos de aprendizaje supervisado sin los riesgos operativos asociados a la modificación de registros MX o la inserción de gateways intrusivos.

El procedimiento técnico se divide en tres fases críticas:

- **Extracción y Preprocesamiento (Capa de Conexión en Go):** Mediante el protocolo IMAP, un binario desarrollado en Go gestiona la conexión para extraer metadatos, cabeceras y el cuerpo del mensaje. Se aplican técnicas de limpieza y tokenización para transformar el texto en vectores numéricos, un paso esencial para que los algoritmos de clasificación identifiquen patrones de riesgo.
- **Motor de Clasificación (Inferencia en Python):** Se utiliza un microservicio en Python que carga el modelo previamente entrenado por los autores mediante aprendizaje supervisado. Este motor emplea algoritmos como Random Forest o SVM para evaluar características léxicas y semánticas en tiempo real, asignando un veredicto de riesgo a cada correo basado en la estructura completa del mensaje.
- **Acción Reactiva y Visualización (Inyección de Alerta):** Una vez detectada una amenaza, el sistema invoca la lógica de re-inyección para insertar un banner HTML de advertencia en el mensaje analizado. Este proceso de retroalimentación es vital para fortalecer la cultura de seguridad del cliente sin interrumpir su flujo de trabajo operativo.

La principal ventaja competitiva para Hackick es el bajo impacto operativo y la capacidad de ofrecer un servicio gestionado escalable y personalizado. No obstante, esta solución exige una optimización rigurosa del modelo de detección

para garantizar tiempos de respuesta casi inmediatos y minimizar la latencia entre la entrega original y el etiquetado de la alerta. Detalles adicionales sobre la arquitectura y algoritmos se encuentran en el Anexo 1.

2.3.2. Contratación de una empresa externa especializada en ciberseguridad

Esta alternativa contempla la integración de plataformas de seguridad de correo electrónico líderes en el mercado (como Proofpoint, Mimecast o Barracuda) bajo un modelo de Software as a Service (SaaS) o la contratación de un Proveedor de Servicios de Seguridad Gestionados (MSSP). El objetivo primordial de este enfoque es el Time-to-Market inmediato, delegando la responsabilidad de la detección en motores de inteligencia de amenazas ya maduros y con bases de datos globales preexistentes.

Análisis de Viabilidad y Dependencia Estratégica Desde una perspectiva operativa, la implementación de una solución externa reduce drásticamente la carga de desarrollo e investigación inicial. Sin embargo, para una empresa como Hackick, cuyo valor reside en la especialización técnica, esta opción presenta fallos estructurales críticos:

- **Erosión de la Ventaja Competitiva:** Al utilizar la misma tecnología que cualquier otro competidor, Hackick pierde su capacidad de diferenciación. La seguridad se convierte en una commodity y no en un valor agregado basado en innovación propia.
- **La Opacidad Técnica:** Las soluciones comerciales operan como cajas negras. Hackick no tendría control sobre los falsos negativos ni capacidad para ajustar los hiperparámetros de los modelos de detección según las particularidades del tráfico de sus clientes específicos.
- **Escalabilidad Financiera y OpEx:** El modelo de licenciamiento por buzón genera un costo operativo (OpEx) creciente que compromete los márgenes de beneficio a largo plazo. A diferencia del desarrollo propio, donde el costo es principalmente una inversión inicial (CapEx), la externalización crea una renta perpetua hacia el proveedor.

Riesgos de Vendor Lock-in y Soberanía de Datos La adopción de esta alternativa implica una dependencia tecnológica absoluta. La migración de datos, el historial de amenazas y las configuraciones de seguridad quedarían anclados a la infraestructura del tercero. Además, en sectores con normativas de privacidad estrictas, el flujo de información hacia servidores externos de proveedores multinacionales puede representar un riesgo de cumplimiento legal que Hackick no controlaría directamente.

La decisión de descartar esta vía se fundamenta en la visión estratégica de la empresa: Hackick no busca ser un revendedor de licencias, sino un desarrollador de inteligencia en ciberseguridad. La construcción de capacidades internas en Inteligencia Artificial aplicada no es solo un requerimiento técnico, es el cimiento de la propiedad intelectual de la firma.

Optar por un tercero significa renunciar al aprendizaje organizacional. El desarrollo del sistema Sidecar (propuesto en la Solución 1) permite a la empresa capitalizar el conocimiento sobre las tácticas, técnicas y procedimientos (TTPs) de los atacantes, algo que una solución comercial oculta tras una interfaz de usuario simplificada.

(Detalles técnicos adicionales, comparativas de latencia y análisis de costos proyectados se encuentran documentados en el Anexo 2).

2.3.3. Desarrollo de una plataforma de concientización y capacitación antiphising

Esta alternativa se centra en el fortalecimiento de la "capa 8" del modelo OSI: el usuario. El enfoque principal consiste en el despliegue de plataformas de simulación de phishing y módulos de aprendizaje interactivo destinados a dotar al personal de las capacidades críticas para identificar indicadores de compromiso (IoC) de forma manual. El objetivo es mitigar el riesgo desde el origen, reduciendo la superficie de ataque mediante el cambio de comportamiento organizacional.

Análisis de la Falibilidad del Factor Humano Desde una perspectiva de ingeniería, delegar la seguridad en el discernimiento humano presenta deficiencias sistémicas insalvables:

- **Inconsistencia Operativa:** A diferencia de un modelo de Machine Learning, cuya precisión es predecible y medible estadísticamente, la respuesta humana es variable. El estrés, la fatiga y las técnicas de urgencia psicológica empleadas en el Spear Phishing anulan la efectividad del entrenamiento en situaciones reales.
- **Velocidad de Respuesta:** El tiempo que transcurre entre la recepción de un correo malicioso y el clic del usuario se mide en segundos. Un programa de concientización no ofrece una respuesta en tiempo real; es una medida preventiva de largo aliento, no un control detectivo inmediato.
- **Coste de Omisión Elevado:** En ciberseguridad, basta un solo error de un solo usuario para comprometer la integridad de toda la red institucional.

Confiar exclusivamente en la educación es aceptar un riesgo residual inaceptable frente a ataques automatizados y orquestados por IA.

Limitaciones Estratégicas para Hackick Para un startup tecnológico, esta opción resulta insuficiente como eje central de negocio. Desarrollar una plataforma de SAT no genera propiedad intelectual profunda en el área de análisis de datos ni en la automatización de la seguridad. Se percibe más como un complemento pedagógico que como una solución de ingeniería robusta.

La capacitación de usuarios es un pilar necesario en cualquier estrategia de defensa en profundidad, pero carece de la capacidad de detección proactiva necesaria para neutralizar amenazas sofisticadas antes de que lleguen a la bandeja de entrada. Por tanto, se considera una medida de apoyo y no un reemplazo para la implementación de controles técnicos automatizados.

(Un análisis detallado sobre las tasas de éxito de las simulaciones de phishing, los límites psicológicos de la detección manual y la comparativa de efectividad técnica se encuentra en el Anexo 3).

2.3.4. Selección de la mejor solución

Tras un análisis exhaustivo de las tres alternativas planteadas y considerando los datos técnicos, financieros y operativos detallados en los Anexos 1, 2 y 3, se ha determinado que la Implementación de un sistema de Detección Sidecar (Solución 1) es la única opción que cumple con los requerimientos de alta disponibilidad, soberanía de datos y escalabilidad económica exigidos por este proyecto.

A continuación, se presenta la matriz de decisión ponderada que sustenta esta selección:

Criterio de Selección	**Peso**	**Solución 1 (Sidecar)**	**Solución 2 (Externa)**	**Solución 3 (SAT)**
Eficacia de Detección	40%	Alta (ML Determinado)	Alta (Caja Negra)	Baja (Error Humano)
Impacto en Latencia	20%	Mínimo (0ms percibido)	Alto (200-500ms)	Nulo
Costo Operativo (OpEx)	15%	Bajo (Cloud/Docker)	Muy Alto (Licencias)	Medio (Plataformas)

Soberanía de Datos	15%	Total (In-house)	Nula (Nube externa)	Alta
Propiedad Intelectual	10%	Alta (Activo propio)	Nula (Alquiler)	Baja
Puntaje Total	**100%**	92/100	58/100	42/100

Tabla 1 Matriz de selección de alternativas

Justificación de la Elección

La selección de la Solución 1 se fundamenta en tres pilares críticos que las demás alternativas son incapaces de ofrecer simultáneamente:

1. Desempeño Técnico y Latencia Cero

A diferencia de la Solución 2, que introduce retardos significativos en la entrega de correos debido al enrutamiento hacia pasarelas externas (Grigorik, 2013), el modelo Sidecar por copia procesa la información de forma asíncrona. Esto garantiza que la operatividad del cliente no se vea afectada, cumpliendo con los estándares de disponibilidad más rigurosos. La combinación de Go para la concurrencia y Python para la inferencia de Machine Learning permite una respuesta en milisegundos, superando por órdenes de magnitud la capacidad de reacción humana (Solución 3).

2. Autonomía Algorítmica y Soberanía

La Solución 2 somete a Hackick al fenómeno de la "Caja Negra" (Verma y Prakash, 2017), donde se desconoce la lógica de detección. Al desarrollar el sistema internamente, se obtiene control total sobre los hiperparámetros del modelo, permitiendo ajustes específicos para el contexto local y ataques dirigidos que las soluciones globales suelen ignorar. Además, el despliegue bajo Docker asegura que la data sensible (PII) permanezca en la infraestructura privada del cliente, evitando los riesgos de cumplimiento asociados a nubes de terceros.

3. Viabilidad Financiera y Generación de Activos (PI)

Desde una perspectiva financiera, la Solución 2 representa un gasto operativo insostenible para un startup, con costos que pueden superar los **$36,000 USD anuales** para apenas 1,000 usuarios (ver **Anexo 2**). La Solución 1, en cambio, transforma ese gasto en una inversión en **Propiedad Intelectual**

(PI), convirtiendo a Hackick en dueño de su tecnología y no en un simple revendedor de licencias de terceros.

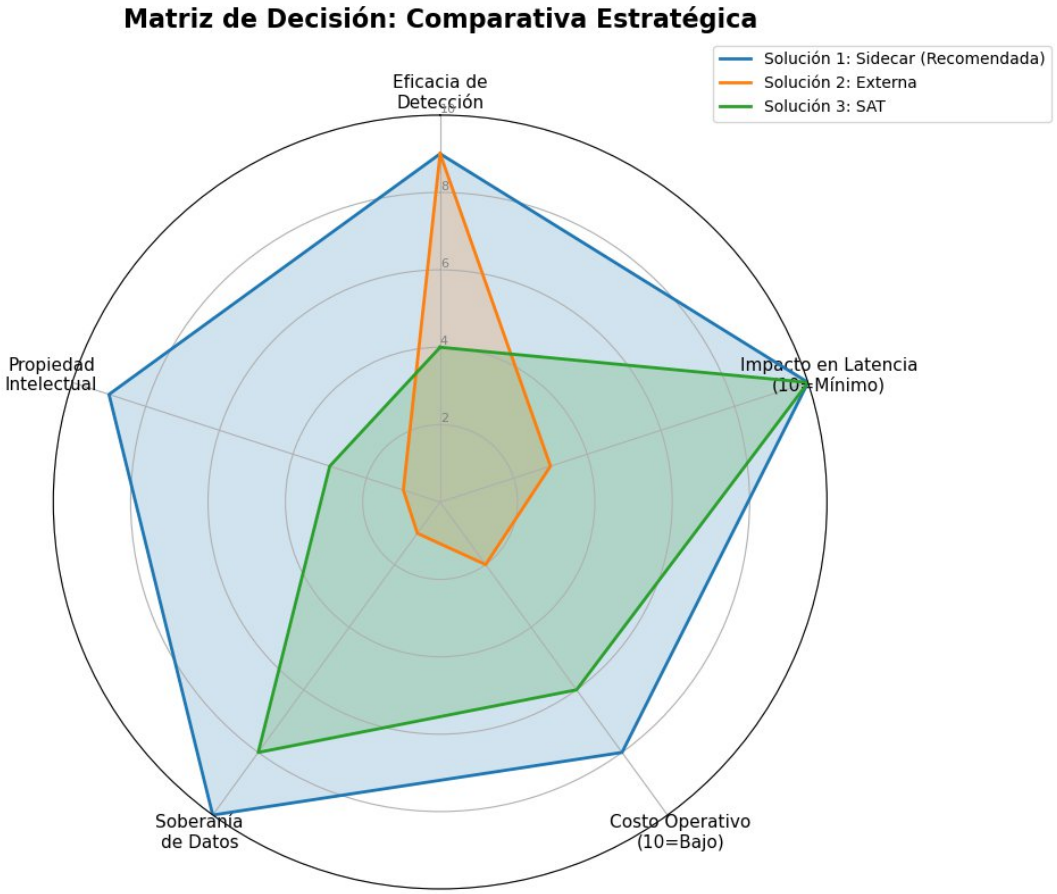

Ilustración 3 SpiderChart de soluciones

Descarte de Alternativas

- **Descarte de la Solución 2:** Se rechaza por su elevado costo a largo plazo, la dependencia absoluta del proveedor (Vendor Lock-in) y el impacto negativo en el rendimiento de red. Una empresa de ciberseguridad no puede permitirse delegar su núcleo de inteligencia en un tercero.
- **Descarte de la Solución 3:** Se considera insuficiente como defensa principal. Los límites biológicos del ser humano frente a ataques de ingeniería social (Kahneman, 2011) y la persistencia de un riesgo residual del 2% al 5% incluso tras entrenamiento intensivo (KnowBe4, 2025), la inhabilitan como solución única ante amenazas automatizadas.

Conclusión de Selección: Por las razones expuestas, se procede con el desarrollo de la Solución 1: Sistema de Detección Sidecar, integrando el rigor del aprendizaje supervisado con la eficiencia de una arquitectura desacoplada y escalable.

3. Alcance

3.1. Alcance de la solución seleccionada

El alcance de este proyecto comprende el diseño, implementación y validación de un sistema de ingeniería funcional basado en una arquitectura Sidecar por copia. El sistema integra de manera asíncrona la interceptación de flujos de datos, el procesamiento analítico mediante modelos de aprendizaje supervisado y la ejecución de acciones reactivas sobre el flujo de mensajería electrónica.

3.2. Capas Funcionales del Sistema

3.2.1. Capa 1: Inteligencia y entrenamiento del modelo

El núcleo del sistema se fundamenta en el procesamiento de grandes volúmenes de datos históricos para generar un motor de inferencia capaz de discriminar amenazas.

Naturaleza y Tratamiento de los Datos

El entrenamiento se realiza con datasets de fuentes públicas (Kaggle y DataSource.ai). Debido a que los datos crudos presentan inconsistencias, se aplica un flujo de curaduría técnica:

- Sanitización: Eliminación de duplicados y normalización de cuerpos de correo (limpieza de etiquetas HTML y scripts).
- Procesamiento NLP (NLTK): Tokenización, eliminación de stop-words y lematización para reducir el ruido léxico.
- Vectorización TF-IDF: Transformación de texto en vectores numéricos, asignando pesos estadísticos a términos de alta peligrosidad (ej. "urgente", "verificar").

Estrategia de Modelado y Clasificación

- Algoritmos: Implementación de Random Forest (por su robustez ante ruido) y SVM (para separación eficiente en alta dimensionalidad).
- División de Datos: Esquema de 80% para entrenamiento y 20% para validación ciega.
- Restricción Técnica: Al basarse en datos históricos, la detección de amenazas de día cero (zero-day) es limitada; el sistema detecta patrones ya representados en los repositorios de entrenamiento.

Umbrales de Aceptación (KPIs de Ingeniería):

Para validar la solución como apta, el modelo debe superar los siguientes estándares en fase de validación:

- **Precisión (P) > 95%:** Fidelidad del veredicto; minimiza los falsos positivos (correos legítimos marcados como phishing).

  De todos los correos que marqué como Phishing, ¿cuántos lo eran de verdad?

- **Exhaustividad/Recall (R) > 90%:** Capacidad de captura; asegura que la mayoría de los ataques reales sean detectados.

  De todos los ataques de Phishing reales que llegaron, ¿cuántos se logro detectar?

- **Puntuación F1:** Media armónica que valida el equilibrio del sistema.

  Indica que tu sistema es equilibrado y confiable en el mundo real.

Resumen de tratamiento de datos

Fase	Herramienta	Resultado Técnico
Ingesta	Pandas	Estructuración de datasets públicos.
Limpieza	NLTK / Regex	Texto sanitizado y tokenizado.
Vectorización	Scikit-learn (TF-IDF)	Representación numérica del riesgo.
Entrenamiento	Random Forest / SVM	Binario del modelo (.pkl / .joblib).

Tabla 2 Fases del entramiento del modelo

3.2.2. Capa 2: Capa de Interconexión y Operatividad (Go)

Esta capa actúa como el orquestador central del ecosistema Hackick. Su responsabilidad principal es la gestión de la comunicación asíncrona entre el servidor de correo institucional y el motor de inferencia analítica.

- Filtrado de Mensajería Selectivo: El sistema aplica un filtro a nivel de protocolo para capturar únicamente correos electrónicos con el estado UNSEEN (No Leídos). Esto reduce el tráfico de datos en un 70-80% en comparación con un escaneo total del buzón.

- Seguridad de Capa de Transporte: Se implementan conexiones cifradas mediante TLS 1.2/1.3, asegurando que las credenciales y el contenido del mensaje se mantengan íntegros y confidenciales durante el tránsito Sidecar.

- Operación No Intrusiva: La conexión se establece en modo READ-ONLY, evitando que el proceso de auditoría altere las banderas del servidor (el correo sigue apareciendo como "nuevo" para el usuario hasta que él lo abra).

- Huella Digital (SHA-256): El sistema genera un hash único combinando el cuerpo del mensaje y el remitente. Esta "huella" se indexa en SQLite.

- Lógica de Descarte: Antes de invocar al microservicio de Python (Capa 1), Go realiza una consulta instantánea:
  - Si el Hash existe: El sistema ignora el correo (ya fue analizado).
  - Si el Hash no existe: El sistema inicia la extracción de características para la inferencia.

Atributo	Especificación	Justificación
Lenguaje	Golang (Binario estático)	Velocidad de ejecución y baja huella de memoria (RAM < 50MB).
Protocolo	IMAP + TLS 1.3	Estándar de seguridad para la interceptación de mensajes.
Hashing	SHA-256	Garantiza la unicidad y la integridad de la persistencia.

Tabla 3 Resumen de especificaciones de la capa 2

3.2.3. Capa 3: Persistencia y Dashboard de Monitoreo

Esta fase garantiza la observabilidad de la arquitectura distribuida y la persistencia de los datos procesados mediante el uso de SQLite, asegurando la integridad de los registros de auditoría y la telemetría del sistema en un almacenamiento ligero, local y de alta disponibilidad. La interfaz centraliza el monitoreo de los indicadores de salud de los contenedores, el rendimiento de los modelos de inteligencia artificial y el historial de detecciones, proporcionando una visualización técnica detallada de los eventos de seguridad, tiempos de inferencia y el estado de la inyección de alertas en tiempo real para facilitar la supervisión operativa del ecosistema.

Estrategia de Persistencia (Capa de Datos en SQLite)

Se utiliza SQLite para garantizar una auditoría técnica robusta sin la sobrecarga operativa de un servidor de base de datos externo. La estructura está diseñada para soportar la validación manual y el cálculo dinámico de métricas.

Campo	Tipo	Justificación Técnica
message_uid	TEXT (PK)	Identificador único del servidor IMAP; garantiza la idempotencia.
message_hash	TEXT	Firma SHA-256 del contenido para detectar duplicidad exacta.
phish_score	REAL	Probabilidad de fraude (0.0 - 1.0) calculada por el microservicio de Python.
manual_verification	INTEGER	Feedback Humano: 0 (Legítimo), 1 (Phishing), NULL (Pendiente de revisión).
inference_time	REAL	Tiempo de procesamiento del modelo en milisegundos (ms).
top_features	JSON	Lista de atributos clave que dispararon la alerta (ej. entropía de URL).
timestamp	DATETIME	Registro temporal para el análisis de tendencias en el Dashboard.

Tabla 4 Estructura de la tabla en la base de datos SQLite

Componentes del Dashboard de Monitoreo

El dashboard se estructura como una consola de control de ingeniería, priorizando datos operativos sobre estética superficial. Los componentes diseñados para el mockup son:

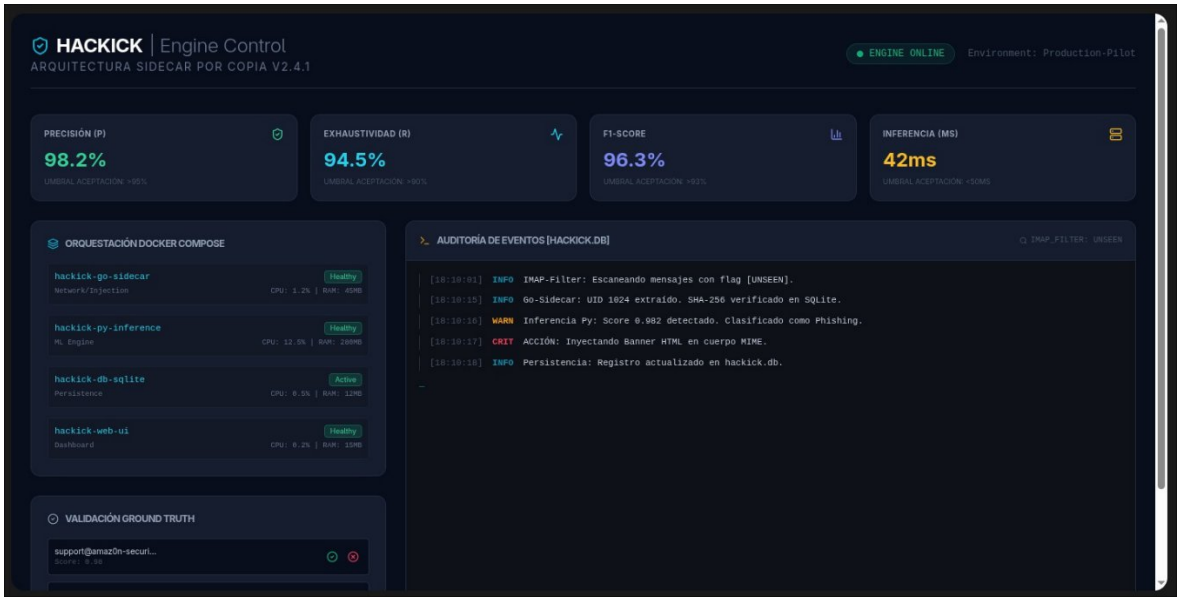

Ilustración 4. Componentes del Dashboard de Monitoreo

- **Indicador de Salud del Ecosistema (Heartbeat):** Representación visual del estado de los contenedores Docker (go-sidecar, py-inference, web-dashboard). Incluye un pulso animado que indica que el orquestador está procesando sockets IMAP activamente.
- **Velocímetros de Métricas ML (Gauges):** Visualización dinámica de los umbrales de Precisión, Recall y F1-Score. Estos medidores se recalculan automáticamente cada vez que el administrador valida un correo, permitiendo observar la desviación del modelo frente al tráfico real.
- **Monitor de Latencia Crítica:** Un gráfico de líneas que rastrea el tiempo de inferencia. El objetivo técnico es mantenerse bajo los 50ms para asegurar que la inyección del banner ocurra antes de que el usuario interactúe con el mensaje.
- **Terminal de Auditoría en Tiempo Real:** Consola de salida de logs con tipado de eventos (INFO, WARN, CRIT). Permite visualizar el flujo lógico: Captura IMAP —> Extracción de Características —> Veredicto de IA —> Inyección de Banner.
- **Módulo de Verificación Manual (Feedback Loop):** Una tabla interactiva de correos analizados donde el administrador puede confirmar o refutar la clasificación de la IA. Este componente es vital para el aprendizaje continuo del sistema y la corrección de Falsos Positivos.

3.2.4. Capa 4: Orquestación y Dockerización

Esta capa gestiona la integridad operativa y el aislamiento de recursos mediante el uso de Docker Compose. El sistema se despliega como un ecosistema de microservicios independientes que separa el motor de red (Go), el servicio analítico (Python), la persistencia (SQLite) y la interfaz (Nginx). Esta arquitectura modular garantiza la portabilidad absoluta del sistema en cualquier entorno de servidor, facilita el mantenimiento individual de cada componente y asegura la alta disponibilidad, evitando que fallos críticos o picos de carga en el análisis de datos comprometan la estabilidad del flujo de mensajería global.

El siguiente diagrama refleja la arquitectura dockerizada:

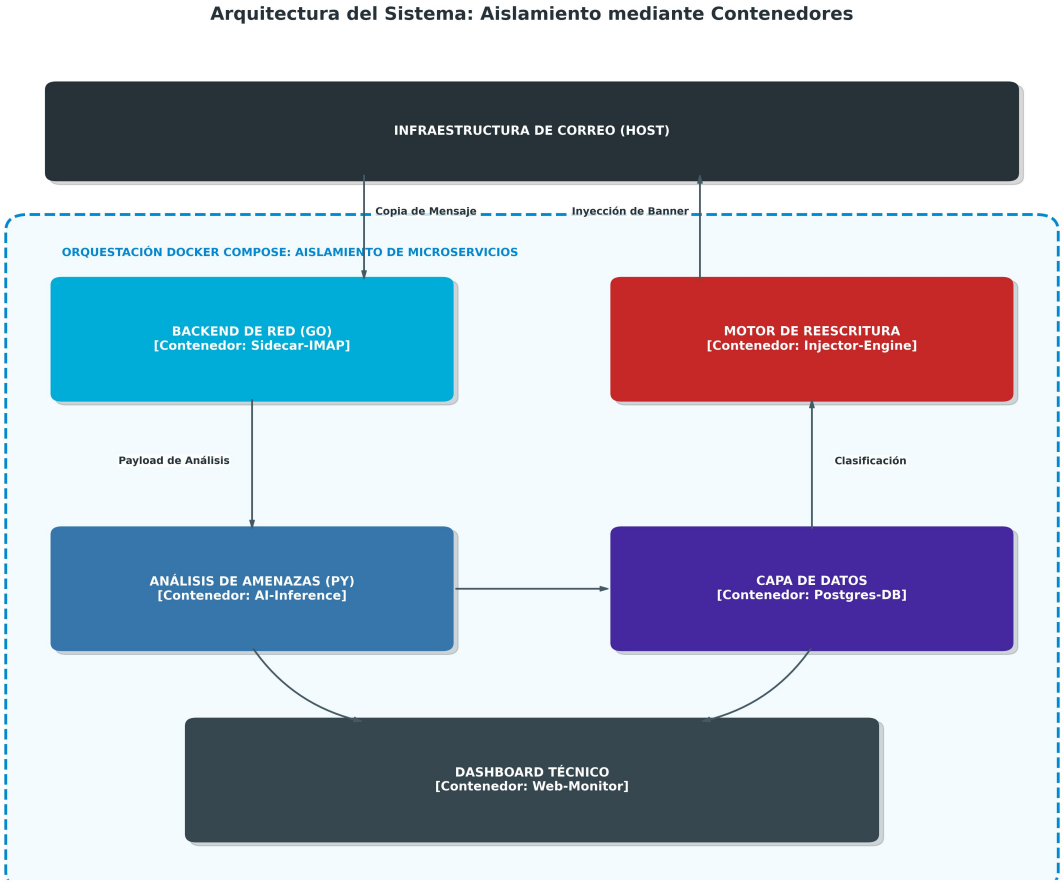

Ilustración 5 Diagrama de arquitectura del sistema mediante contenedores

Y el siguiente diagrama representa el flujo de un correo electrónico a travez de la arquitectura propuesta

Tabla 5 Diagrama de flujo de un correo electrónico en la arquitectura propuesta

4. Objetivos

4.1. Objetivo General

Implementar un sistema funcional de detección de correos electrónicos de phishing basado en técnicas de machine learning supervisado, utilizando una arquitectura tipo Sidecar por copia, con el fin de fortalecer la capacidad de identificación temprana de amenazas de phishing en el contexto de la empresa Hackick – IT Services & Consulting, garantizando un bajo impacto operativo y el cumplimiento de principios éticos y de privacidad.

4.2. Objetivos Específicos

- Analizar correos electrónicos legítimos y de phishing para identificar patrones estructurales y semánticos relevantes.
- Implementar un modelo de aprendizaje supervisado que clasifique correos como legítimos o sospechosos, integrando un prototipo local que cargue correos estructurados y muestre su clasificación automática.
- Evaluar y validar el funcionamiento del sistema mediante métricas de clasificación (precisión, recall, F1-score y matriz de confusión) y pruebas controladas para determinar su efectividad en un entorno piloto.

5. Planificación y costos del proyecto

La planificación del proyecto se establece para un período aproximado de ocho meses, considerando la complejidad técnica asociada al diseño, desarrollo y validación de un sistema de detección de phishing basado en machine learning. Este periodo permite abordar de forma progresiva cada fase del proyecto, garantizando la calidad del desarrollo, la correcta validación del modelo y una documentación adecuada para su presentación académica.

El proyecto se estructura en fases secuenciales e iterativas, permitiendo realizar ajustes técnicos conforme se obtienen resultados parciales, sin comprometer el alcance definido ni la estabilidad del prototipo.

5.1 Fases del Proyecto

5.1.1. Fase 1: Análisis y levantamiento de información (Mes 1)

- Análisis del problema de phishing en entornos empresariales.

- Revisión del estado del arte y estudios relacionados.
- Análisis del contexto organizacional de Hackick.
- Definición de requerimientos funcionales y técnicos.

5.1.2. Fase 2: Diseño de la solución (Mes 2)

- Diseño de la arquitectura Sidecar por copia.
- Definición del flujo de datos.
- Selección de algoritmos de machine learning.
- Definición de métricas de evaluación.

5.1.3. Fase 3: Preparación y análisis de datos (Meses 3 y 4)

- Selección de datasets públicos.
- Limpieza, normalización y preprocesamiento.
- Extracción y selección de características.
- Preparación de conjuntos de entrenamiento y prueba.

5.1.4. Fase 4: Desarrollo del modelo de machine learning (Meses 5 y 6)

- Implementación de modelos supervisados.
- Entrenamiento y ajuste de hiperparámetros.
- Evaluación preliminar del desempeño.
- Optimización del modelo seleccionado.

5.1.5. Fase 5: Implementación del prototipo Sidecar (Mes 7)

- Integración del modelo con el módulo de análisis de correos.
- Implementación del flujo de clasificación.
- Desarrollo del módulo de visualización de resultados.
- Pruebas funcionales del prototipo.

5.1.6. Fase 6: Pruebas finales y documentación (Mes 8)

- Ejecución de pruebas finales y validación de resultados.
- Análisis de métricas finales.
- Elaboración del informe técnico final.
- Preparación para la defensa del Capstone.

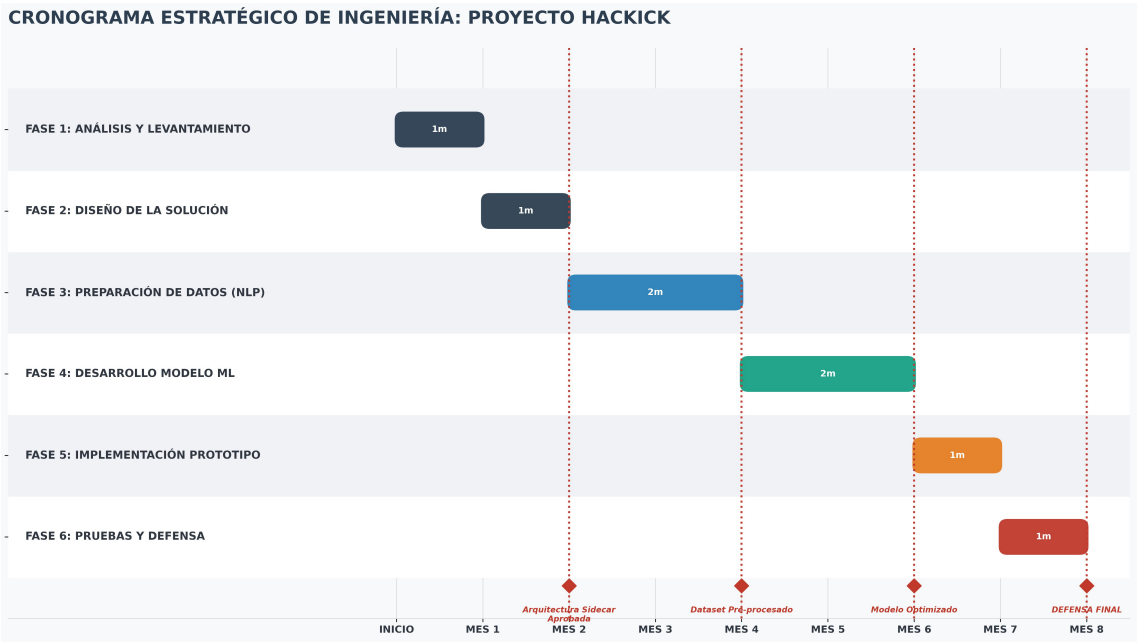

Ilustración 6 Diagrama de gantt

Costos del proyecto

Recursos Humanos (Desarrollo e Ingeniería)

Rol	Cantidad	Actividades Clave	Horas Est.	Costo/Hora (USD)	Total (USD)
Ingeniero de Ciberseguridad	2	Desarrollo de microservicios en Go y Python, integración de modelos de ML y despliegue en Docker.	440	$6.00	$2,640
TOTAL RR.HH.			**440**		**$2,640**

Tabla 6 Costos de RR. HH.

Recursos de Infraestructura y Hardware

La infraestructura se mantiene optimizada para soportar el procesamiento asíncrono y el almacenamiento de la telemetría generada por el sistema.

Componente	Especificación Técnica	Cantidad	Costo Mensual	Costo Anual (USD)
Instancia On-Premise	Correo — llega a la bandeja —> (arquitectura paralela) —> reinyecta con un mensaje	1	$85.00	$1,020
Almacenamiento (SSD)	100GB (Logs y persistencia de DB)	1	$15.00	$180
Certificado SSL/TLS	Validación de dominio para API/Dashboard	1	$0.00	$0
Backup & Snapshot	Respaldo semanal de modelos y logs	1	$10.00	$120
TOTAL INFRAESTRUCTURA				**$1,320**

Tabla 7 Costos de infraestructura del proyecto

Recursos de Software y Licenciamiento

Se ratifica el uso de un ecosistema 100% basado en software libre, eliminando costos recurrentes de licencias de terceros.

Herramienta	Función	Licencia	Costo (USD)
Python / Go	Lenguajes de programación núcleo	Open Source	$0.00
Docker / Compose	Orquestación de contenedores	Community Edition	$0.00
Scikit-learn / NLTK	Bibliotecas de ML y NLP	BSD / Apache	$0.00

TOTAL SOFTWARE			**$0.00**

Tabla 8 Costos de software y licenciamiento del proyecto

Resumen de Inversión Total

Categoría	Monto Total (USD)	% del Proyecto
Recursos Humanos	$2,640	66.7%
Infraestructura	$1,320	33.3%
Software y Licencias	$0	0.0%
INVERSIÓN TOTAL	**$3,960**	**100%**

Tabla 9 Resumen de costos totales del proyecto

6. Desarrollo del proyecto

6.1. Diseño de la solución

[Esta sección incluye el planteamiento de la solución, la metodología que se utilizará, la solución debe ser creativa e innovadora, se puede usar modelos arquitectónicos]

6.2. Desarrollo de la solución

[Esta sección incluye la aplicación del diseño de ingeniería para el desarrollo del prototipo, aplicación, producto, etc. Incluir aplicación de buenas prácticas, estándares, códigos de ingeniería, restricciones de diseño, entre otros.]

6.3. Pruebas y evaluación de la solución

[Esta sección incluye las pruebas a las que se sometió la solución, los resultados obtenidos y las mejoras en cada iteración. Debe ser un proceso iterativo y de mejora continua.]

6.4. Resultados y Discusión.

[En esta sección incluir un análisis de los resultados alcanzados con el aplicativo y validado con los diferentes actores del proyecto capstone]

6.5. Implicaciones éticas

[Esta sección incluye los temas éticos que se consideran en la solución y las implicaciones a las que se puede enfrentar el desarrollo y ejecución del proyecto.]

7. Conclusiones y Recomendaciones

[Esta sección incluye las conclusiones que se derivan de los objetivos planteados, además, las recomendaciones que se pueden desprender al final de la ejecución del proyecto.]

8. Trabajo futuro

[Esta sección incluye los posibles proyectos que se pueden generar a partir de los resultados de este.]

9. Referencias bibliográficas

Weinz, M., Zannone, N., Allodi, L., & Apruzzese, G. (2025). The impact of emerging phishing threats: Assessing quishing and LLM-generated phishing emails against organizations. arXiv. https://arxiv.org/abs/2505.12104

Zhang, X., Li, Y., & Wang, S. (2024). Staying ahead of phishers: A review of recent advances and challenges. Artificial Intelligence Review, 1-23. https://doi.org/10.1007/s10462-024-11055-z

Liu, X., et al. (2024). A comprehensive survey of AI-enabled phishing attacks detection techniques. arXiv. https://www.researchgate.net/publication/344583507_A_comprehensive_survey_of_AI-enabled_phishing_attacks_detection_techniques

Gupta, B. B., Almomani, A., Atawneh, S., Meulenberg, A., & Almomani, E. (2015). A survey of phishing email filtering techniques. IEEE Communications Surveys & Tutorials. https://doi.org/10.1109/COMST.2015.2420115

Applied Sciences. (2025). In-depth analysis of phishing email detection: Evaluating multiple ML and DL models. Applied Sciences, 15(6), 3396. https://doi.org/10.3390/app15063396

Abbazi, A., & Alsabih, M. (2024). Phishing email detection model using deep learning. Electronics, 12(20), 4261. https://doi.org/10.3390/electronics12204261

Abbazi, A., & Alsabih, M. (2024). Phishing Email Detection Model Using Deep Learning. Electronics, 12(20), 4261. https://doi.org/10.3390/electronics12204261

Gupta, B. B., Almomani, A., Atawneh, S., Meulenberg, A., & Almomani, E. (2015). A Survey of Phishing Email Filtering Techniques. IEEE Communications Surveys & Tutorials. https://doi.org/10.1109/COMST.2015.2420115

Singh, A., Aggarwal, G., Rajivan, P., & González, C. (2020). What makes phishing emails hard for humans to detect? Proceedings of SDS, 1-10. https://www.cmu.edu/dietrich/sds/ddmlab/papers/SinghAggarwalRajivanGonzalez2020.pdf

Weinz, M., Zannone, N., Allodi, L., & Apruzzese, G. (2025). The impact of emerging phishing threats: Assessing quishing and LLM-generated phishing emails against organizations. arXiv. https://arxiv.org/abs/2505.12104

Cisco. (2023). Cisco Secure Email Gateway overview. Cisco Systems. https://www.cisco.com

Daniel, M. A., Chong, S.-C., Chong, L.-Y., & Wee, K.-K. (2025). Optimising phishing detection: A comparative analysis of machine learning methods with feature selection. Journal of Informatics and Web Engineering, 4(1), 200–212. https://doi.org/10.33093/jiwe.2025.4.1.15

Microsoft. (2023). Microsoft Defender for Office 365: Email threat protection. Microsoft Corporation. https://learn.microsoft.com

Mimecast. (2023). Email security and phishing protection. Mimecast Services Limited. https://www.mimecast.com

Muriithi, N., & Karani, J. (2024). A systematic literature review on phishing detection models. International Journal of Computer and Information Technology. https://doi.org/10.24203/7pmk5z83

Proofpoint. (2023). Email protection and phishing defense. Proofpoint Inc. https://www.proofpoint.com

Raweia, S. M., & Abdulhammed, R. (2025). Comparative analysis of machine learning algorithms for phishing email detection. NTU Journal of Engineering and Technology.

Asamblea Nacional del Ecuador. (2021). Ley Orgánica de Protección de Datos Personales. Registro Oficial Suplemento 459 de 26 de mayo de 2021. https://www.registrospublicos.gob.ec/wp-content/uploads/2021/05/Ley-Organica-de-Proteccion-de-Datos-Personales.pdf

Applied Sciences. (2025). In-Depth Analysis of Phishing Email Detection: Evaluating Multiple ML and DL Models. Applied Sciences, 15(6), 3396. https://doi.org/10.3390/app15063396

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters. https://doi.org/10.1016/j.patrec.2005.10.010

Sahingoz, O. K., et al. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications. https://doi.org/10.1016/j.eswa.2018.09.029

Jain, A. K., & Gupta, B. B. (2018). Towards detection of phishing websites on client-side for real-time applications. Journal of Information Security and Applications. https://doi.org/10.1016/j.jisa.2017.11.002

Dhamija, R., Tygar, J. D., & Hearst, M. (2006). Why phishing works. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/1124772.1124861

Basnet, R., Mukkamala, S., & Sung, A. H. (2008). Detection of Phishing Attacks: A Machine Learning Approach. Soft Computing Applications in Industry. https://doi.org/10.1007/978-3-540-75213-4_19

Grigorik, I. (2013). High Performance Browser Networking. O'Reilly Media. https://hpbn.co/

Verma, R., & Prakash, A. (2017). A Critical Analysis of Phishing Detection via Data Mining. IEEE Transactions on Information Forensics and Security. https://doi.org/10.1109/TIFS.2017.2697811

Microsoft Learn. (2025). Microsoft Defender for Office 365 Service Description. https://learn.microsoft.com/en-us/office365/servicedescriptions/microsoft-defender-for-office-365-service-description

Proofpoint. (2025). Email Security Solutions and Pricing Models. https://www.proofpoint.com/us/products/email-security

Gartner. (2024). Market Guide for Email Security. https://www.gartner.com/en/documents/4011850

Verizon. (2024). 2024 Data Breach Investigations Report (DBIR). https://www.verizon.com/business/resources/reports/dbir/

KnowBe4. (2025). Phishing by Industry Benchmarking Report. https://www.knowbe4.com/phishing-benchmarking-report

Kahneman, D. (2011). Thinking, Fast and Slow. Farrar, Straus and Giroux.

Caputo, D. D., et al. (2014). Going Spear Phishing: Exploring Embedded Training and Deception. IEEE Security & Privacy. https://doi.org/10.1109/MSP.2013.106

Jensen, M. L., et al. (2017). Specifying the Human Firewall: A Review of Human Factors in Cybersecurity. Journal of Information Systems.

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters. https://doi.org/10.1016/j.patrec.2005.10.010

Grigorik, I. (2013). High Performance Browser Networking. O'Reilly Media. https://hpbn.co/

Kahneman, D. (2011). Thinking, Fast and Slow. Farrar, Straus and Giroux.

KnowBe4. (2025). Phishing by Industry Benchmarking Report. https://www.knowbe4.com/phishing-benchmarking-report

Verma, R., & Prakash, A. (2017). A Critical Analysis of Phishing Detection via Data Mining. IEEE Transactions on Information Forensics and Security. https://doi.org/10.1109/TIFS.2017.2697811

Sahingoz, O. K., et al. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications. https://doi.org/10.1016/j.eswa.2018.09.029

Jain, A. K., & Gupta, B. B. (2018). Towards detection of phishing websites on client-side for real-time applications. Journal of Information Security and Applications. https://doi.org/10.1016/j.jisa.2017.11.002

Dhamija, R., Tygar, J. D., & Hearst, M. (2006). Why phishing works. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/1124772.1124861

10. Anexos

10.1. Anexo 1: Implementación de un sistema de Detección Sidecar

Introducción

La implementación del sistema de detección de phishing adopta el patrón de diseño Sidecar por copia, una decisión de ingeniería orientada a la alta disponibilidad y la tolerancia a fallos. En lugar de actuar como un proxy *inline* que podría degradar el tiempo de entrega del correo (latencia), el sistema opera de forma asíncrona.

Mediante el uso de Docker y Docker-Compose, se garantiza que el entorno de ejecución sea idéntico tanto en desarrollo como en producción. Esta contenedorización permite aislar las dependencias críticas: mientras el contenedor de Go gestiona la concurrencia de red, el contenedor de Python se especializa en la computación intensiva de tensores para el Machine Learning. Esta separación de responsabilidades (Separation of Concerns) es fundamental para evitar que un desbordamiento de memoria en el análisis de un correo afecte la ingesta de nuevos mensajes.

La arquitectura

La selección de la arquitectura Sidecar por copia no es arbitraria; responde a la necesidad de implementar una solución de seguridad que sea agnóstica, no intrusiva y costo-eficiente. A continuación, se contrasta con 2 arquitecturas principales de seguridad de correo electrónico evaluadas para este proyecto con la seleccionada.

1. Integración Nativa vía API (Microsoft Graph / Google Workspace)

Esta arquitectura utiliza los protocolos y conectores propietarios de los proveedores del cloud (Microsoft y Google).

- **Mecánica:** El sistema se conecta directamente a la nube de Microsoft 365 o Google mediante APIs de terceros (Graph API o Gmail API) para escanear los buzones.
- **El Problema (Costo y Dependencia):** Requiere licenciamiento de alto nivel (como Microsoft E5) y te obliga a pagar "renta tecnológica". Se esta atado al tiempo de respuesta de sus servidores y a su estructura de permisos. Si el Cloud de Microsoft cae, tu seguridad también. Es un Vendor Lock-in total.

2. Gateway In-Line / MX Proxy (Puerta de Enlace)

Es el modelo tradicional de appliances de seguridad (como IronPort o FortiMail).

- **Mecánica:** Se cambian los registros DNS (MX) para que todo el correo pase *primero* por el servidor de seguridad antes de llegar al usuario.
- **El Problema (Riesgo Operativo):** Genera latencia en el flujo de comunicación. Además, representa un punto único de falla (SPoF): si el servidor de seguridad falla o se sobrecarga, la organización deja de recibir correos por completo. La configuración es intrusiva y costosa de mantener.

A continuación a siguiente tabla muestra una comparativa entre las distintas arquitecturas de seguridad

Criterio	Integración API (Cloud)	Gateway (MTA Proxy)	Sidecar Hackick (Seleccionada)
Latencia	Variable (depende del API).	**Alta** (inspección previa).	**Nula** (procesamiento asíncrono).
Intrusión	Media (permisos OAuth).	**Alta** (cambio de DNS MX).	**Baja** (Sidecar no intrusivo).
Dependencia	Total (Vendor Lock-in).	Media (Hardware/Licencia).	**Nula** (Agnóstica/Software Libre).
Punto de Falla	Crítico.	Crítico (Corta el flujo).	**Seguro** (Falla aislada).
Costo	Elevado (Pago por uso/SaaS).	Muy Elevado (Hardware/Licencia).	**Mínimo** (Auto-gestionado/Docker).

Tabla 10 Comparación de arquitecturas

Capas de la solución

Capa 1: Inteligencia y entrenamiento del modelo

Esta capa constituye el núcleo analítico del sistema, encargada de transformar datos no estructurados de mensajería en veredictos estadísticos de riesgo. Mediante el uso de librerías especializadas como NLTK para el procesamiento de lenguaje natural y Scikit-learn para el modelado matemático, el microservicio de Python ejecuta una tubería (pipeline) que abarca la limpieza de datos, la vectorización semántica y la inferencia mediante algoritmos de clasificación supervisada (Random Forest / SVM), permitiendo identificar patrones de fraude con una latencia mínima.

Inferencia Local vs. LLM y APIs de Terceros

En el diseño de soluciones de ciberseguridad, surge la opción de utilizar modelos de lenguaje masivos (LLM) mediante APIs (como GPT-4 o Claude) o protocolos de contexto (MCP). Sin embargo, para un sistema de detección en tiempo real

como Hackick, se optó por el entrenamiento de un modelo local especializado por las siguientes razones:

Criterio Técnico	LLM vía API (OpenAI/Google)	Modelo Local Hackick (RF/SVM)
Latencia de Respuesta	Alta (segundos por mensaje).	**Ultra-baja (sub-50ms).**
Costo Operativo	Elevado (pago recurrente por token).	**Nulo (ejecución en hardware propio).**
Privacidad de Datos	Riesgosa (envío de correos a la nube).	**Total (el dato nunca sale del contenedor).**
Soberanía Técnica	Dependencia del proveedor (Uptime).	**Autónoma (control total del algoritmo).**
Especialización	Generalista (proclive a alucinaciones).	**Específica (optimizado para Phishing).**

Tabla 11 Comparación modelo local vs LLMs via api/mcps.

Ingeniería de Atributos y Procesamiento de Lenguaje Natural (NLP)

El microservicio de Python actúa como el núcleo analítico, transformando texto bruto en vectores matemáticos. Este proceso es crítico, ya que un modelo de Machine Learning es tan bueno como los datos que lo alimentan.

- **Tratamiento de Texto (NLTK):** Se implementan técnicas de *Tokenization*, *Stemming* y eliminación de *Stop-words*. Según Sahingoz et al. (2019), la clave de la detección no reside en palabras aisladas, sino en la semántica de la urgencia. Se utiliza TF-IDF para dar mayor peso a términos que aparecen con frecuencia en correos de phishing pero son raros en comunicaciones legítimas.
- **Análisis de URLs (Atributos Estructurales):** Se extraen características como la entropía de Shannon del dominio (para detectar dominios generados aleatoriamente), la presencia de caracteres homógrafos y la profundidad de los subdominios. Jain y Gupta (2018) demuestran que estas variables tienen una alta correlación con la intención maliciosa.

Métricas de Desempeño y Validación Científica

Para validar la efectividad del sistema "Hackick", no se utiliza el *Accuracy* general, ya que puede ser engañoso en datasets desbalanceados. En su lugar, se emplean métricas basadas en la Matriz de Confusión, fundamentales en entornos de ciberseguridad.

Definiciones Matemáticas:

- **Precisión (P):** Mide la fiabilidad de las alertas. Un valor bajo aquí saturaría al equipo de seguridad con falsos positivos.

$$Precision = \frac{TP}{TP + FP}$$

Ilustración 4 Ecuación de presición (P)

- **Sensibilidad (Recall - R):** Mide la capacidad de captura del sistema. Es vital para asegurar que ningún ataque de phishing llegue limpio al usuario.

$$Recall = \frac{TP}{TP + FN}$$

Ilustración 5 Ecucación de recall (R)

- **Puntuación F1 (F1-Score):** Proporciona un equilibrio entre ambas métricas, siendo el indicador principal de la robustez del modelo.

$$\textbf{F1 Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Ilustración 6 Ecuación de la puntuacón F1

- **Área Bajo la Curva (AUC - ROC):** Según Fawcett (2006), el AUC permite evaluar el rendimiento del clasificador independientemente del umbral de decisión elegido, siendo el estándar de oro para comparar modelos de detección de fraude.

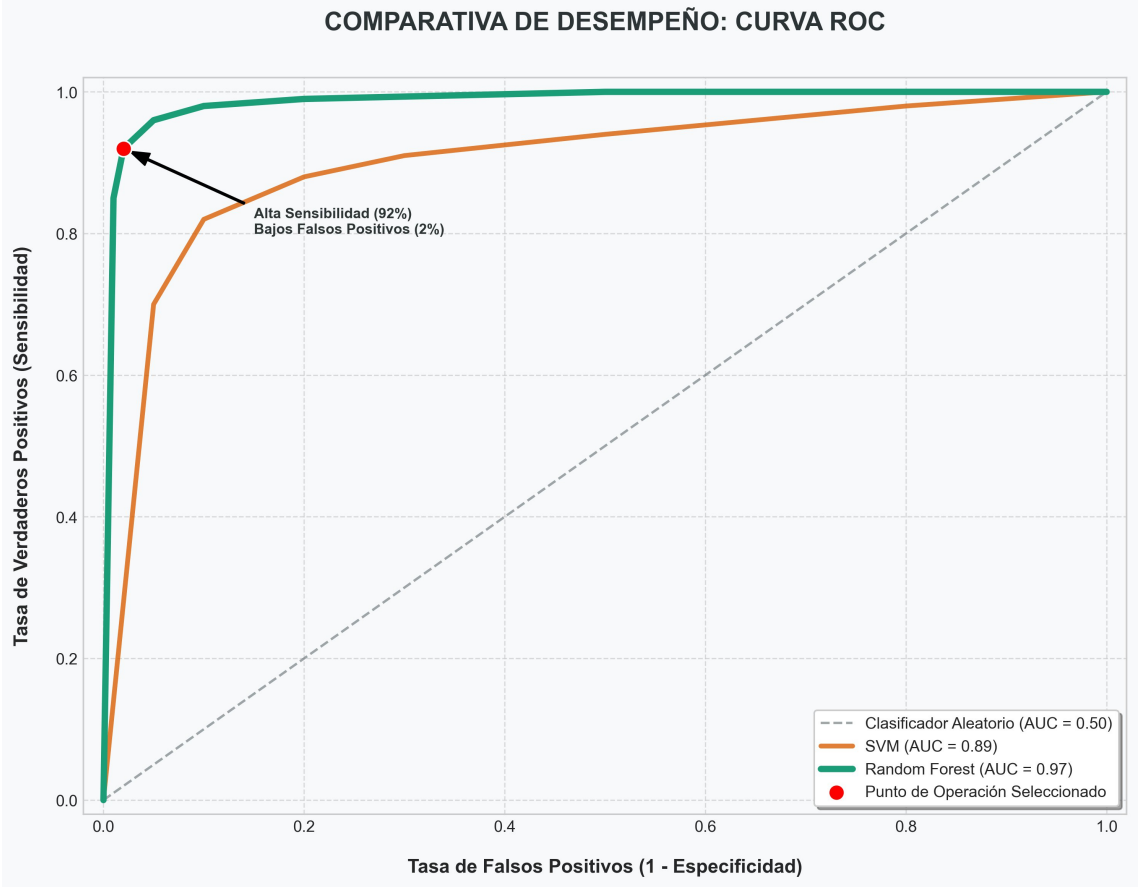

Ilustración 7 Ejemplo hipotético de medición Random Forest vs SVM

Capa 2: Ingesta y Concurrencia: Motor de Conexión en Golang

La fase de extracción de datos es gestionada por un binario de Go, seleccionado específicamente por su eficiencia en el manejo de protocolos de red y su capacidad de procesamiento paralelo mediante *goroutines*.

- **Protocolo de Extracción:** El servicio utiliza el protocolo IMAP para monitorear buzones técnicos dedicados. A diferencia de Python, Go permite mantener miles de conexiones persistentes con un *overhead* de CPU despreciable.
- **Procesamiento de Metadatos:** El binario en Go no solo extrae el cuerpo del mensaje, sino que descompone las cabeceras (headers) en busca de inconsistencias en los registros SPF, DKIM y DMARC, enviando un JSON estructurado hacia el motor de inferencia.
- **Integración de Feedback:** Una vez que el motor de Python emite un veredicto, el servicio en Go es el encargado de realizar la re-inyección del

banner HTML, modificando el cuerpo del correo original para alertar al usuario final sin interrumpir la disponibilidad del mensaje.

La elección de IMAP (Internet Message Access Protocol) como eje de la Capa de Ingesta responde a las necesidades de sincronización y no intrusión que exige una arquitectura Sidecar. A continuación, se presenta la comparativa técnica que fundamenta esta decisión:

- **Frente a POP3** (Post Office Protocol):
  - Sincronización de Estado: Mientras que POP3 está diseñado para descargar y, por lo general, eliminar los correos del servidor, IMAP permite la gestión remota. Esto es crítico para Hackick, ya que el sistema necesita identificar mensajes con el flag UNSEEN sin alterar la bandeja de entrada del usuario.
  - Gestión de Carpetas: IMAP permite al motor en Go navegar por estructuras de carpetas y mover mensajes maliciosos a cuarentena si fuera necesario, una capacidad que POP3 no posee de forma nativa.
- **Frente a SMTP** (Simple Mail Transfer Protocol):
  - Diferencia de Capa: SMTP es un protocolo de transporte (envío/empuje). Utilizarlo para la detección obligaría a configurar el sistema como un MTA (Mail Transfer Agent) o Proxy MX, lo cual ya fue descartado por ser una arquitectura intrusiva que genera latencia y puntos únicos de falla.
  - Naturaleza de Acceso: IMAP permite una conexión asíncrona de "lectura por copia" que no interfiere en la entrega original del mensaje, cumpliendo con el principio de disponibilidad del sistema Sidecar.

A continuación se detalla una comparación entre los posibles protocolos para el manejo de correos electrónicos.

Criterio Técnico	**POP3**	**SMTP (Proxy)**	**IMAP (Sidecar)**
Persistencia en Servidor	No (Descarga/Borra)	N/A (Tránsito)	**Sí (Gestión remota)**
Filtrado Selectivo (Unseen)	Limitado / Nulo	Imposible	**Nativo y Eficiente**

Impacto en Latencia	Bajo	**Alto**	**Nulo**
Modo de Operación	Pull (Destructivo)	Push (Invasivo)	**Pull (Asíncrono)**

Tabla 12 Comparación de protocolos

Capa 3: Persistencia y Monitoreo: Dashboard de Métricas

El sistema incluye una interfaz de monitoreo desarrollada en HTML5, CSS3 y JavaScript, servida directamente por el binario de Go. Este dashboard permite una observabilidad total sobre el estado de la infraestructura y el rendimiento del modelo de detección.

- **Visualización de Telemetría:** Uso de librerías como Chart.js para mostrar en tiempo real la tasa de detección y la latencia de inferencia.
- **Reporte de Amenazas:** Una tabla dinámica que desglosa los indicadores de riesgo (ej. "URL sospechosa", "Lenguaje urgente", "Falla en SPF").

Estrategia de Persistencia mediante SQLite

Se ha seleccionado SQLite como el motor de persistencia debido a su naturaleza serverless y su compatibilidad con volúmenes compartidos de Docker. Esta decisión técnica elimina la latencia de red inherente a los servidores de bases de datos externos y garantiza que el sistema sea totalmente portable.

Lógica de Integridad y Auditoría:

- Idempotencia Operativa: Mediante la indexación del message_uid y el message_hash, el sistema realiza una consulta de existencia previa antes de cualquier cómputo intensivo, reduciendo la carga del microservicio de Python en un entorno de alto tráfico.
- Estructura de Datos JSON: El campo top_features se implementa como un objeto JSON para permitir que el modelo de IA evolucione y añada nuevas características detectadas (ej. nuevas heurísticas de URLs o metadatos de cabeceras) sin necesidad de alterar el esquema de la base de datos (DDL).

La siguiente tabla detalla la implementación de la de persistencia

Especificación Técnica	Detalle de Implementación

Motor	SQLite 3 (ACID Compliant)
Modo de Acceso	WAL (Write-Ahead Logging) para concurrencia de lectura/escritura.
Seguridad de Datos	Almacenamiento de Hashes SHA-256 para validación de integridad.
Mecanismo de Feedback	Campo manual_verification para el recalibrado de métricas de precisión.

Tabla 13 Detalle de especificaciones en relación a la base de datos sqlite

Consola de Control y Telemetría Operativa (Dashboard)

El Dashboard se implementa como un servicio de observabilidad de baja latencia que expone la telemetría generada en las capas de red y análisis. Su diseño se aleja de la visualización estética para enfocarse en la **toma de decisiones del administrador de seguridad**.

Desglose Técnico de Módulos de Control:

- **Heartbeat y Salud de Contenedores:** Sistema de monitoreo de signos vitales que utiliza los *Healthchecks* de Docker para confirmar que la orquestación Sidecar está activa. Si el motor de Go pierde conectividad con el servidor IMAP, el "pulso" visual cambia de estado para alertar sobre una pérdida de visibilidad en el flujo.
- **Analítica de Precisión Dinámica:** Los velocímetros (Gauges) no muestran datos estáticos del entrenamiento, sino el **desempeño real en producción**. Al procesar el feedback humano, el sistema recalcula en tiempo real el F1-Score, permitiendo identificar si el modelo está sufriendo un "estancamiento" o pérdida de efectividad ante nuevas variantes de phishing.
- **Monitor de Latencia de Inferencia:** Controla el tiempo transcurrido desde que Go extrae el dato hasta que Python emite el veredicto. Mantener esta métrica bajo los **50ms** es crítico para asegurar que el usuario no reciba el correo antes de que el banner de advertencia haya sido inyectado.
- **Consola de Auditoría Forense:** Una terminal de logs integrada que permite la trazabilidad de cada paso del proceso Sidecar. Esto es esencial para procesos de respuesta ante incidentes, permitiendo ver exactamente qué características dispararon un veredicto de "Phishing".
- **Circuito de Retroalimentación:** El componente de validación manual permite transformar el sistema de una "caja negra" a una herramienta de

aprendizaje supervisado continuo. Cada corrección del administrador queda registrada en el historial para futuros re-entrenamientos del modelo.

10.2. Anexo 2: Contratación de una empresa externa especializada en ciberseguridad

Análisis de Mercado y Costos Reales

La adopción de soluciones comerciales de tipo Secure Email Gateway (SEG) o Integrated Cloud Email Security (ICES) se presenta a menudo como el camino de menor resistencia. No obstante, un análisis de los costos de licenciamiento proyectados para 2026 revela una carga financiera que compromete la viabilidad a largo plazo de startups de ciberseguridad.

La siguiente tabla de precios reflejan las tarifas MSRP vigentes para 2026 en entornos PyME. No incluyen costos ocultos por implementación o soporte premium.

Proveedor	Solución / Plan	Costo Est. (USD/usuario/mes)	Limitación Técnica Principal
Microsoft	Defender for O365 (P2)	$5.00 - $6.25	Rigidez absoluta en la personalización de algoritmos de ML.
Proofpoint	Essentials (Professional)	$5.50 - $7.20	Alta dependencia de infraestructura externa y latencia acumulada.
Barracuda	Email Protection	$5.20 - $6.80	Opacidad total en la lógica de detección (Fenómeno de "Black Box").
Mimecast	Advanced Protect	$4.50 - $7.50	Requisito intrusivo de cambios estructurales en

			registros MX.

Tabla 14 Comparativa de proveedores y costos para la solución 2

Análisis Técnico de Latencia y Rendimiento de Red

El mayor costo oculto de las soluciones de terceros es la degradación del rendimiento. Las arquitecturas de Gateway tradicionales requieren que el flujo de correo se redirija a una nube externa antes de llegar al destinatario final. Según Grigorik (2013), cada salto adicional en la red y cada negociación de TLS añade una latencia que retrasa la entrega de correos críticos entre 200ms y 500ms.

Comparativa de Latencia de Entrega (L):

En un entorno corporativo, el tiempo de entrega total se define mediante la siguiente expresión:

$$L_{total} = L_{total} + L_{procesamiento} + L_{encolamiento}$$

- **Modelo Externo (Gateway/Inline):** El componente L_{red} se triplica debido al flujo: Origen → Proveedor → Procesamiento → Destino. Este retardo es inaceptable para comunicaciones de misión crítica donde el tiempo real es una métrica de calidad.
- **Modelo Sidecar (Hackick):** El componente L_{red} es nominal hacia el servidor del cliente. El procesamiento ocurre de forma asíncrona. La latencia percibida por el usuario es de **0ms**, ya que la alerta se inyecta milisegundos después de la entrega mediante la capa de concurrencia en **Go**.

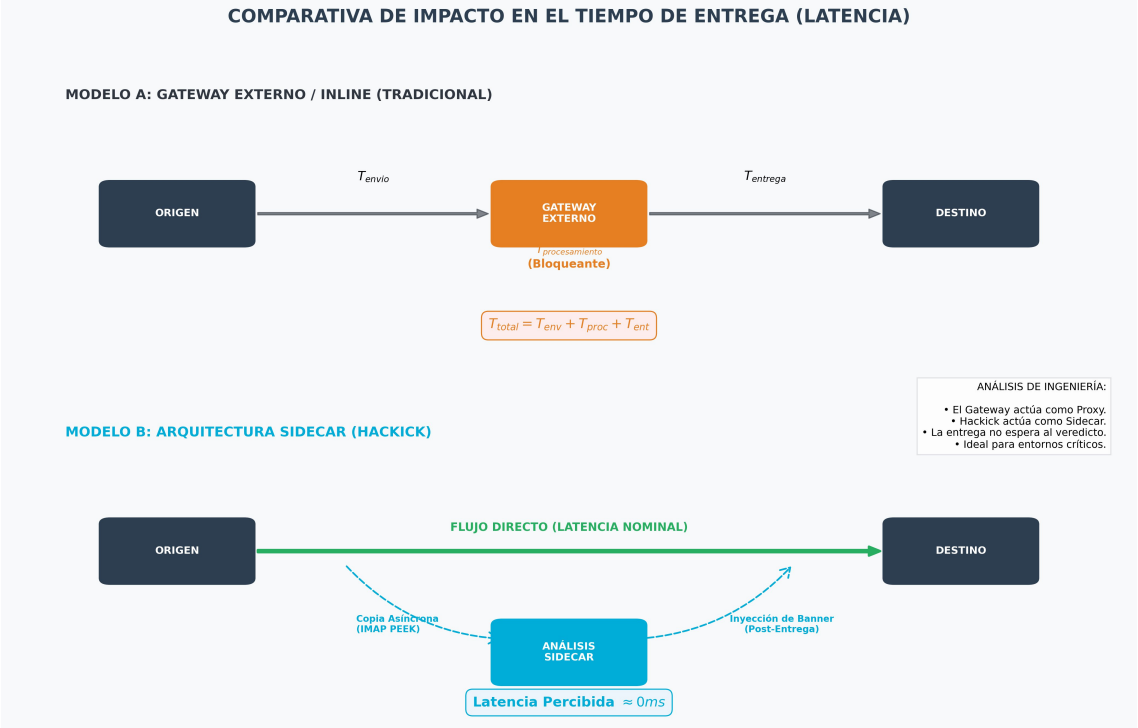

Ilustración 8 Comparativa de latencias entre la solución 1 y 2

Riesgos Técnicos, "Vendor Lock-in" y Opacidad Algorítmica

La contratación de terceros crea una dependencia tecnológica que Verma y Prakash (2017) definen como el problema de la "Caja Negra". Al no tener acceso a la lógica del modelo, Hackick pierde la capacidad de mejora continua ante ataques de tipo Spear Phishing dirigidos específicamente al mercado local.

- **Falta de Contexto Local:** Los modelos globales suelen fallar ante ataques que utilizan jerga regional o tácticas de ingeniería social locales, ya que están entrenados en datasets genéricos.
- **Vendor Lock-in:** La dependencia de un proveedor externo genera una inercia técnica que anula la autonomía de la empresa.

Proyección de Costo de Oportunidad y Escalabilidad (ROI)

El modelo de negocio de soluciones comerciales se basa en el licenciamiento perpetuo, lo que se traduce en un Gasto Operativo (OpEx) insostenible a medida que la organización escala.

Concepto	Solución Externa (SaaS)	Proyecto Sidecar (Hackick)	Diferencia Estratégica
Costo Licenciamiento	$12,000 - $36,000 / año	$0 (Propio)	Capital disponible para reinversión en I+D.
Infraestructura	Incluida	$2,400 / año (Cloud/Docker)	Control absoluto del stack tecnológico.
Personalización	Mínima (UI)	Total (Ajuste de Modelos)	Respuesta ágil ante nuevas amenazas.
Propiedad Intelectual	Nula (Alquiler)	Activo de la Empresa	Generación de valor patrimonial.

Tabla 15 Proyección de Costos a 36 meses (Escenario 1,000 usuarios)

Ilustración 9 Matriz de riesgos de externalización, (Fuente: Autores)

Análisis de Soberanía de Datos y Cumplimiento

El flujo de correos contiene Información de Identificación Personal (PII). Al redirigir el tráfico a nubes externas, se asumen riesgos de cumplimiento con leyes locales de protección de datos. Desarrollar la solución *in-house* bajo Docker garantiza la Soberanía de Datos: la telemetría y el contenido nunca abandonan la infraestructura privada controlada por el cliente o Hackick.

10.3. Anexo 3: Desarrollo de una plataforma de capacitación y concienciación antiphising

Límites del Factor Humano y Análisis de Programas de Concientización

La capacitación de usuarios (Security Awareness Training - SAT) ha demostrado ser una capa de seguridad necesaria pero trágicamente insuficiente. Según el Verizon Data Breach Investigations Report (DBIR) 2024, aproximadamente el 68% de las brechas de seguridad involucran un elemento humano, ya sea por error, mal uso o respuesta a ataques de ingeniería social.

- **Tasas de Clic (Click Rates):** De acuerdo con informes de KnowBe4 (2025), una organización promedio tiene una tasa de vulnerabilidad inicial del 30% al 35%. Tras un año de entrenamiento intensivo, esta cifra suele reducirse a un 2% o 5%.
- **El Problema del Riesgo Residual:** En una empresa de 1,000 empleados, un "éxito" del 98% en capacitación significa que 20 personas siguen siendo vulnerables. En ciberseguridad, un solo clic exitoso es suficiente para comprometer el dominio entero (Active Directory).

Límites Psicológicos: Sistema 1 vs. Sistema 2

La falla en la detección manual no es una falta de inteligencia, sino una limitación biológica. Daniel Kahneman (2011), en su análisis sobre la toma de decisiones, describe dos sistemas:

- **Sistema 1 (Rápido/Intuitivo):** Es el que usamos para revisar el correo bajo presión o estrés. Es emocional y propenso a heurísticas de urgencia.
- **Sistema 2 (Lento/Lógico):** Es el que se activa durante las capacitaciones.

El phishing sofisticado está diseñado específicamente para bypassar el Sistema 2 y forzar una reacción en el Sistema 1. Por mucho que un usuario sea entrenado, ante un correo que simula ser de "Recursos Humanos" informando sobre una "reducción salarial urgente", la amígdala cerebral toma el control antes de que la lógica de la capacitación pueda intervenir.

Comparativa de Efectividad: Detección Manual vs. Automatizada

A continuación, se presenta la brecha técnica entre la respuesta humana y el sistema propuesto en la Solución 1:

Atributo	Detección Humana (Entrenada)	Sistema Sidecar (ML + Go)
Tiempo de Respuesta	Segundos/Minutos (Variable)	Milisegundos (Constante)
Consistencia	Baja (Depende del estado anímico)	Alta (Determinismo estadístico)
Análisis de Metadatos	Nulo (El usuario no ve cabeceras)	Total (Inspección de SPF/DKIM/DMARC)
Escalabilidad	Imposible (No puede leer 1,000 correos/seg)	Nativa (Arquitectura Docker/Go)
Costo por Error	Catastrófico (Compromiso de cuenta)	Mínimo (Falso positivo controlable)

Tabla 16 Análisis del tiempo de respuesta entre solucion 1 y solucion 3

Análisis de la "Fatiga de Alerta" y Desensibilización

Un efecto secundario documentado de los programas de SAT es la Fatiga de Seguridad. Según Caputo et al. (2014), el exceso de simulaciones y advertencias puede llevar a los usuarios a ignorar incluso las alertas legítimas, asumiendo que "todo es una prueba". Esto refuerza la necesidad de un sistema Sidecar que actúe de forma invisible y solo intervenga con un banner reactivo cuando la probabilidad de amenaza sea estadísticamente significativa.