**Universidad de Concepción Departamento de Ingeniería Informática y Ciencias de la Computación**

# Sistema de detección de spear phishing basado en modelos de lenguaje

por

**Daniel Michel Carmona**

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Informático

**Patrocinantes:**

Pedro Pinacho Davidson
Fernando Gutiérrez Gómez

Concepción - Abril 2025

Este trabajo fue parcialmente financiado por Proyecto ANID Fondecyt N°11230359, “AN IMMUNE INSPIRED MODEL OF INTRUSION PREVENTION SYSTEM (IPS) FOR COLLABORATIVE AND DISTRIBUTED ENVIROMENTS”.

# 1. Introducción

El *phishing* y el *spear phishing* son tipos de ciberataques que engañan a las personas para que revelen información sensible o descarguen malware. El *phishing* ataca a gran escala, con un bajo índice de conversión compensado por una gran cantidad de intentos, mientras que el *spear phishing*, se dirige a individuos o grupos específicos a través de la personalización, logrando una mayor tasa de conversión con menos intentos.

Ambos tipos de ataque explotan las debilidades del comportamiento humano para acceder de manera no autorizada a información o recursos críticos. El *spear phishing*, a pesar de ser solo el 0.1% de los correos analizados en un estudio de Barracuda [3], fue el responsable del 66% de las brechas de seguridad, lo que demuestra su mayor riesgo en comparación con el *phishing* tradicional.

En los últimos años, los grandes modelos de lenguaje (LLM) como GPT-4 de OpenAI o Gemini de Google han impulsado avances significativos en el procesamiento del lenguaje natural (NLP). Estos modelos se caracterizan por su capacidad de generar texto coherente, contextual y de alta calidad, que en muchas ocasiones es indistinguible del texto creado por humanos [16]. Estos avances han permitido generar ataques de *phishing* y *spear phishing* de gran calidad con mayor facilidad en comparación a los métodos tradicionales. Un ejemplo de esto es la memoria de título de Rodrigo San Martín [4], que implementó un modelo generador de *spear phishing* mediante LLMs capaz de automatizar la creación de ataques personalizados, con el objetivo de evaluar la peligrosidad de estos ataques.

Para contrarrestar estos ataques de *phishing*, se han desarrollado diversas soluciones, dentro de las cuales la más predominante es el uso de Machine Learning (ML) para encontrar patrones en las cabeceras de un *correo electrónico*, como el emisor o lugar de origen del mensaje [5,6]. Nuevos métodos incluyen el uso de NLP junto a técnicas de ML [7] y Deep Learning (DL) [8], donde se analizan distintos componentes del mensaje, como hiperparámetros, intención del mensaje, patrones de gramática y estilo, entre otros.

Si bien los enfoques anteriores son eficientes, la mayoría están limitados a mensajes de correo electrónico y están entrenados casi exclusivamente para el idioma inglés. Un reporte de Zscaler [9] muestra millones de intentos de *phishing* al año en países no angloparlantes, destacando aún más la falta de soluciones para otros idiomas.

Los modelos de lenguaje han demostrado ser altamente efectivos para la detección de *phishing* y *spear phishing* debido a su capacidad para analizar el contenido textual de los mensajes de manera profunda y contextual [11]. Mediante técnicas de NLP estos modelos pueden identificar patrones complejos, como el tono, la intención o las estructuras gramaticales. Además, este enfoque permite superar la barrera del idioma, siendo posible entrenar modelos en otros idiomas o traducir los mensajes al inglés para su revisión [18].

Este trabajo se enfoca en el desarrollo de un sistema de identificación de *spear phishing* basado en modelos de lenguaje, donde a diferencia de los métodos tradicionales, se analizará el contenido textual de los mensajes para determinar si corresponden a un intento de *spear phishing*. El sistema estará diseñado para funcionar en español, dado que actualmente no existen alternativas en este idioma.

## 1.1. Objetivo General

El objetivo de este trabajo es desarrollar un sistema basado en modelos de lenguaje para la detección de ataques de *phishing* y *spear phishing* en español.

## 1.2. Objetivos Específicos

Para lograr el objetivo de este trabajo es necesario desarrollar los siguientes objetivos específicos:

- **Desarrollar un modelo clasificador de *spear phishing* basado en modelos de lenguaje** capaz de identificar y clasificar correctamente los intentos de *spear phishing*.
- **Identificar un dataset adecuado de mensajes** para evaluar el modelo propuesto.
- **Implementar métodos de entrenamiento** basados en machine learning.
- **Evaluar la eficacia del sistema** con un conjunto de datos representativos y escenarios adversariales.

# 2. Marco teórico

Este trabajo se encuentra en la intersección de tres áreas: la ingeniería social, los grandes modelos de lenguaje, y el machine learning. Estos conceptos constituyen la base teórica sobre la cual se desarrolla este proyecto, siendo esenciales para entender el diseño y la implementación del sistema propuesto.

## 2.1. Ingeniería social

El concepto de ingeniería social se refiere al uso de técnicas persuasivas para obtener información confidencial, acceso a sistemas, recursos, o influir en el comportamiento de las personas [12]. La ingeniería social se basa en la explotación de las vulnerabilidades humanas como la curiosidad, la confianza, el miedo, la codicia, o la empatía, las cuales son a menudo consideradas el eslabón más débil en la seguridad informática. Estas técnicas son ampliamente utilizadas en ciberataques como el *phishing* o el *spear phishing*, donde se busca engañar a los usuarios para que realicen acciones perjudiciales.

Los atacantes aprovechan las emociones humanas para inducir a sus objetivos a actuar impulsivamente. Por ejemplo, un correo electrónico que simula ser de una entidad confiable puede usar el miedo a perder acceso a un servicio para forzar a la víctima a compartir sus credenciales. Este tipo de vulnerabilidad humana es difícil de mitigar, ya que depende de la educación y el entrenamiento continuo de los usuarios, lo cual demanda considerable tiempo y recursos.

### 2.1.1. Principios de persuasión de Cialdini

Cialdini describe seis principios fundamentales de persuasión que las personas utilizan, consciente o inconscientemente, para influir en las decisiones de otros: reciprocidad, escasez, autoridad, consistencia, agrado, y consenso social [19]. Estos principios son particularmente relevantes en el contexto de la ingeniería social, ya que los atacantes los emplean para aumentar la efectividad de sus intentos de *phishing* y *spear phishing*. Los principios se detallan a continuación:

- **Reciprocidad:** Las personas tienden a devolver favores o acciones positivas, incluso cuando no las han solicitado. Por ejemplo, un atacante puede ofrecer un cupón de descuento a cambio de dar información personal.
- **Escasez:** Las oportunidades parecen más valiosas cuando su disponibilidad es limitada. Por ejemplo, un correo falso con el asunto "Última oportunidad para obtener un descuento del 50%" puede presionar a las víctimas a actuar rápidamente sin analizar la legitimidad del mensaje.
- **Autoridad:** Las personas tienden a seguir figuras de autoridad o expertos. Los atacantes pueden, por ejemplo, hacerse pasar por el departamento de tecnología de una empresa para solicitar la actualización de contraseñas.
- **Consistencia:** Una vez que las personas han tomado una decisión o postura, tienden a comportarse de manera congruente con ella. Los atacantes pueden enviar mensajes que aparentan continuidad con interacciones previas del objetivo.
- **Agradabilidad:** Es más probable que las personas sean influenciadas por alguien que les agrade, lo que los atacantes pueden aprovechar mediante mensajes aduladores.
- **Consenso social:** Las personas tienden a seguir las acciones de otros, especialmente en situaciones de incertidumbre. Los atacantes pueden aparentar ser respaldados por grupos masivos o tener buenas reseñas en línea.

## 2.2. Clasificación de correos electrónicos

Dentro del conjunto de correos electrónicos es posible identificar múltiples categorías, estas categorías no solo sirven para estructurar el problema, sino también para desarrollar estrategias específicas que permitan abordar con precisión cada caso, distinguiendo correos legítimos de intentos maliciosos, o clasificando correctamente la publicidad frente al *phishing*.

El conjunto de datos consiste en una muestra de correos electrónicos, los cuales se dividen en *Spam* y *Ham*. Dentro de *Spam* encontramos dos categorías: *phishing* y *spear phishing*, y además tenemos la publicidad, que presenta algunos desafíos importantes. La distribución de correos electrónicos se puede ver de manera simplificada en la figura 1. A continuación se discute en detalle estas categorías.

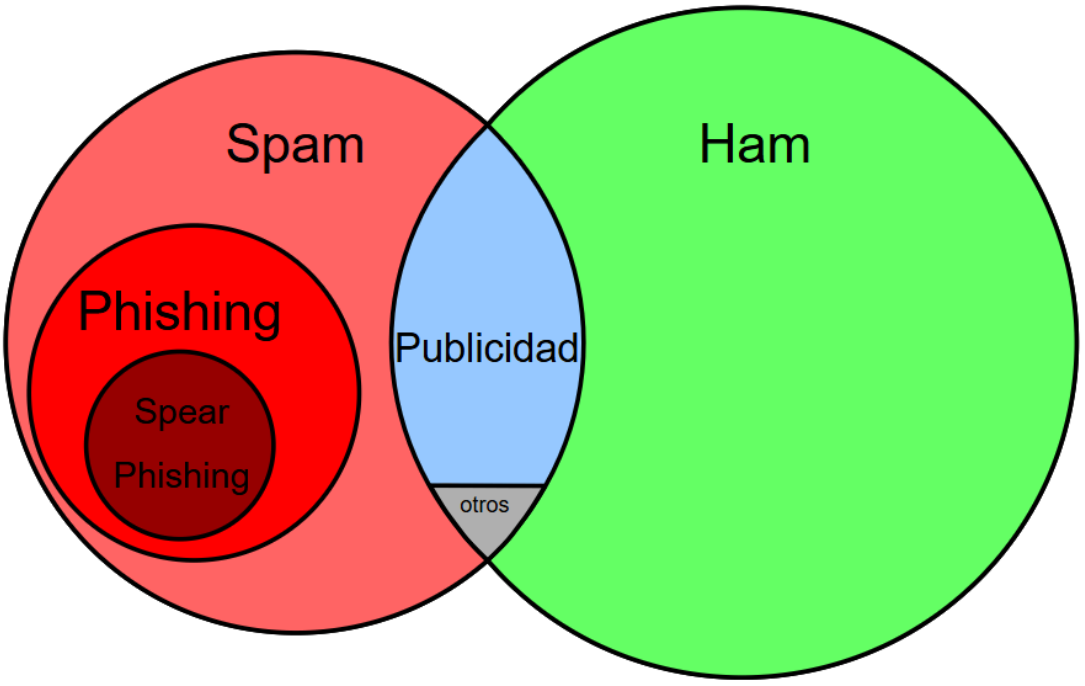



Figura 1 - Visualización de distribución emails según su tipo

### 2.2.1. Spam

El concepto de *Spam* corresponde a una categoría de correos electrónicos que se caracterizan por ser no deseados, enviados masivamente y sin el consentimiento del receptor. Estos mensajes son disruptivos y contienen principalmente contenido promocional, anuncios irrelevantes, estafas, intentos de *phishing* u otros elementos no solicitados.

El *spam* representa una parte significativa del tráfico de correos electrónicos en internet, alcanzando un 45,6% de los correos enviados a nivel global [14]. Esto refleja su uso e impacto masivo en el internet y los desafíos que implica su manejo, tanto para los usuarios como para los proveedores de servicios de correo electrónico.

### 2.2.2. Ham

El término *ham* se origina como el opuesto de *spam* y hace referencia a correos legítimos, relevantes y deseados por el receptor. A diferencia de los correos no deseados o maliciosos, los correos *ham* cumplen un propósito claro y suelen ser fundamentales para las actividades diarias del usuario. Estos mensajes pueden incluir comunicaciones personales como correos de amigos o familiares, mensajes organizacionales, facturas electrónicas, documentos legales y notificaciones importantes, incluso la publicidad puede considerarse *ham* si fue previamente autorizada.

El *ham* representa el flujo normal y esperado de la comunicación mediante correos electrónicos, jugando un papel clave en la productividad e interacción digital cotidiana, por lo cual diferenciarlos del *spam* es esencial para mantener una experiencia grata para los usuarios.

### 2.2.3. Phishing

El *phishing* es un tipo de ciberataque de ingeniería social que usa mensajes o páginas web fraudulentas para engañar a las personas a revelar información sensible, descargar malware o exponerse de otra manera a la ciberdelincuencia [1]. Es uno de los métodos más comunes de ciberataques y es mayormente usado en correos electrónicos, sin embargo existen gran variedad de tipos de phishing que se utilizan por otros medios distintos al correo electrónico, siendo los más notables el *vishing* que hace referencia al phishing mediante llamada (voice phishing), *smishing* que se realiza a través de mensajes SMS y el *web spoofing* que imita páginas web de confianza para capturar datos de la víctima.

### 2.2.4. Spear phishing

El *spear phishing* es un tipo de ataque de *phishing* que se dirige a un individuo o grupo de individuos específicos dentro de una organización e intenta engañarlos para que divulguen información confidencial, descarguen malware, entre otros [2]. Estos mensajes de *spear phishing* se elaboran con datos reales de las víctimas, los cuales pueden ser obtenidos desde páginas web organizacionales, redes sociales o con ayuda de la huella digital que uno deja en internet.

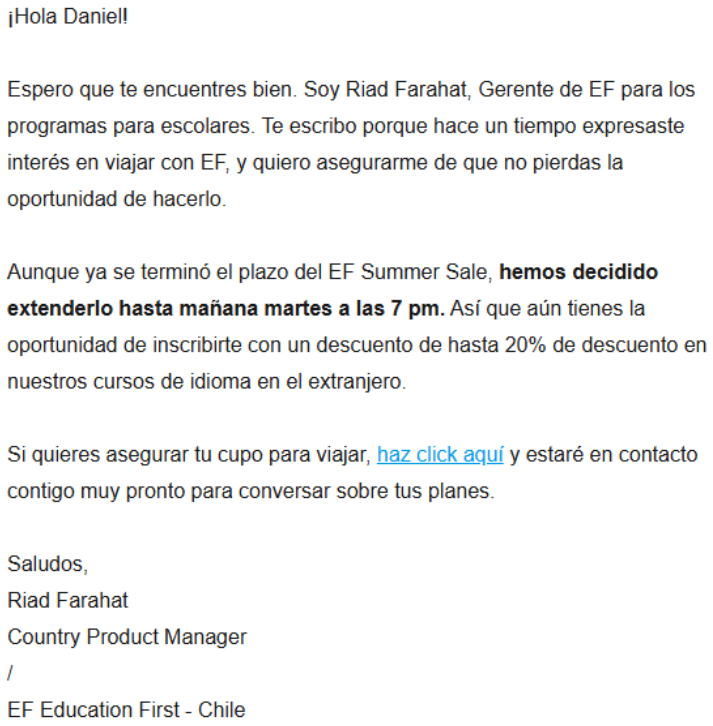
El *spear phishing* representa un 0.1% de los correos electrónicos en internet, sin embargo este resulta en más de la mitad de las brechas de seguridad en organizaciones [3], demostrando su gran eficacia al momento de explotar vulnerabilidades.

### 2.2.5. Publicidad

La *publicidad,* en el contexto de correos electrónicos, se refiere a mensajes cuyo objetivo principal es promocionar productos, servicios o eventos. Estos correos suelen ser enviados por empresas o comerciantes para captar la atención del receptor y generar una acción específica, como realizar una compra, registrarse a un servicio o asistir a un evento.

La publicidad puede utilizar las mismas tácticas de persuasión que los mensajes de phishing o *spear phishing*, la única diferencia entre un buen mensaje de *phishing* y un mensaje de publicidad es la *intención* [15], pues ambos intentan persuadir al receptor a realizar una acción, ya sea maligna, como dar información personal a una página de fraudes, o benigna, como comprar o suscribirse a un servicio.

Un ejemplo de un mensaje de publicidad legítimo, que podría funcionar tanto como publicidad y como phishing es el siguiente mensaje de Education First (EF):



¡Hola Daniel!

Espero que te encuentres bien. Soy Riad Farahat, Gerente de EF para los programas para escolares. Te escribo porque hace un tiempo expresaste interés en viajar con EF, y quiero asegurarme de que no pierdas la oportunidad de hacerlo.

Aunque ya se terminó el plazo del EF Summer Sale, **hemos decidido extenderlo hasta mañana martes a las 7 pm.** Así que aún tienes la oportunidad de inscribirte con un descuento de hasta 20% de descuento en nuestros cursos de idioma en el extranjero.

Si quieres asegurar tu cupo para viajar, haz click aquí y estaré en contacto contigo muy pronto para conversar sobre tus planes.

Saludos,
Riad Farahat
Country Product Manager
/
EF Education First - Chile

Figura 2 - Correo electrónico publicitario EF

El mensaje tiene la intención de persuadir al receptor para concretar una venta, esto lo hace con la ayuda de los principios de persuasión de Cialdini, como por ejemplo:

- La *escasez*, que se puede ver en la línea "Aunque ya se terminó el plazo del EF Summer Sale, hemos decidido extenderlo hasta mañana martes a las 7 pm.".
- La *autoridad*, ya que el remitente se presenta como "Gerente de EF para los programas escolares".
- La *reciprocidad*, ya que el mensaje está ofreciendo un descuento especial y dirigido al receptor.
- La *consistencia*, debido a que el mensaje expresa que el receptor tenía interés previo en usar los servicios de EF.

Ya que este mensaje cubre las mismas bases de principios de persuasión que utilizan los mensajes de *phishing* y *spear phishing*, la única diferencia entre ellos es la *intención* del mensaje, siendo posible tener un enlace que dirija a una página verídica para concretar la venta, o a una página ilegítima para capturar datos o recibir dinero de la víctima. Esto presenta desafíos en la detección de este tipo de mensajes.

## 2.3. Estructura de un correo

Un correo electrónico está compuesto por varias partes esenciales que permiten su correcta identificación, entrega y visualización. A continuación, se detalla la estructura típica de un correo electrónico:

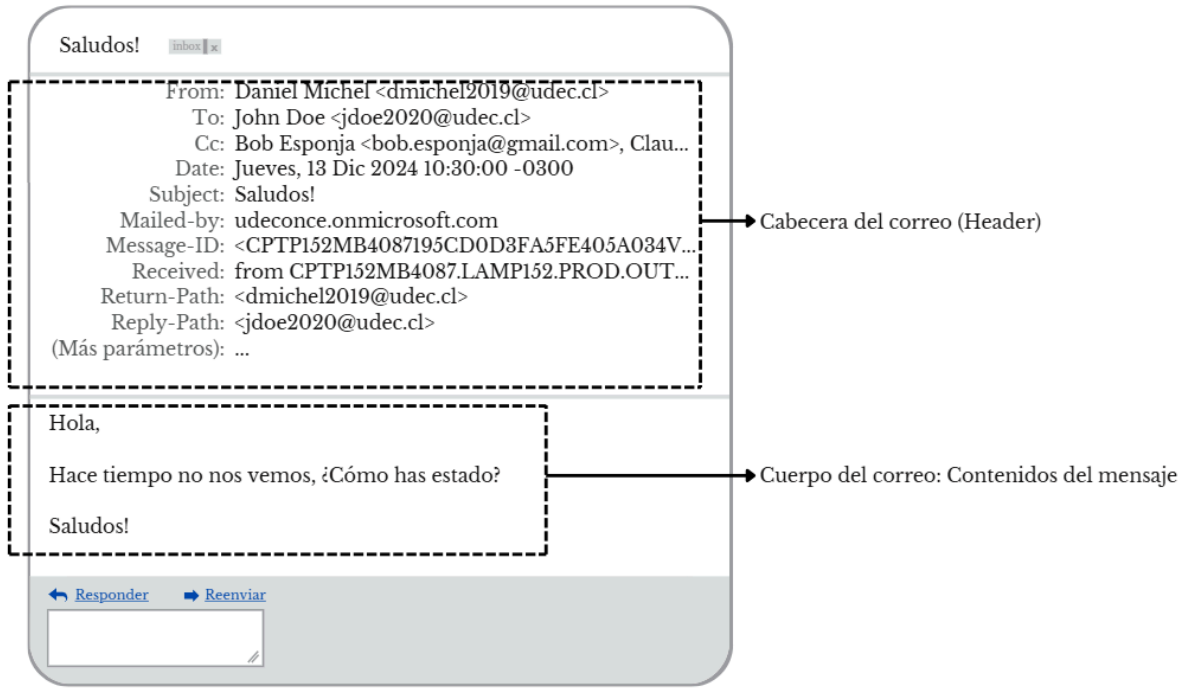



Figura 3 - Estructura de un correo electrónico

Se puede dividir la estructura de un correo electrónico en dos componentes principales:

1. **Cabecera del correo:** La cabecera de un correo electrónico contiene metadatos que proporcionan información sobre el mensaje, estas pueden incluir datos como el emisor, receptor, servidor desde donde se envió, entre otros.

2. **Cuerpo del correo:** El cuerpo del correo es la parte más importante y visible para los usuarios, esta contiene el mensaje enviado, el cual puede estar en formato plano o incluir código HTML, que permite una presentación más atractiva y estructurada.

En el contexto de esta memoria, utilizaremos únicamente el cuerpo del correo, lo que incluye exclusivamente el contenido textual, plano o HTML. Este enfoque presenta varios beneficios para el sistema, el cuerpo del texto contiene varios aspectos que permiten identificar el *phishing* y *spear phishing*, el análisis del cuerpo es también agnóstico a las cabeceras del correo, que pueden variar entre proveedores de servicio y configuraciones de usuario.

Además es importante recalcar que la detección por cabeceras puede ser vulnerable a técnicas como el *zombie phishing* [33], donde se utilizan cuentas previamente vulneradas y por ende, las cabeceras pueden parecer legítimas. Por otro lado, se encuentran una mayor cantidad de datasets y sistemas de detección basados en el cuerpo del correo, que sirven de base para el sistema.

## 2.4. Grandes modelos de lenguaje

Los grandes modelos de lenguaje (LLMs) son sistemas avanzados de inteligencia artificial (IA) diseñados para procesar, comprender y generar texto en lenguaje natural, esto es, lenguaje que podemos comprender como humanos. Estos modelos están construidos sobre arquitecturas de aprendizaje profundo y se entrenan con grandes cantidades de datos textuales. Gracias a su arquitectura y entrenamiento, los LLMs poseen capacidades que les permiten realizar diversas tareas lingüísticas con un alto grado de precisión y adaptabilidad. Entre estas capacidades se incluyen:

- **Generar texto:** Puede redactar texto de distintas longitudes y complejidades, explicar su razonamiento, y dar una respuesta humanamente comprensible.
- **Analizar texto:** Analiza el texto de entrada para determinar la intención y el tono del mensaje. Además de entender las instrucciones de este.
- **Traducir texto:** Debido a la arquitectura de estos modelos, pueden traducir texto a distintos idiomas.

La entrada de los LLMs se denomina *prompt*, el cual se utiliza para interactuar con él. Se trata de una instrucción o conjunto de instrucciones que le indican al modelo la tarea que debe realizar. Los *prompts* pueden variar en complejidad, desde una simple pregunta, como *"¿Qué es el phishing?"*, o un enunciado más elaborado que incluya directrices específicas o casos ilustrativos, por ejemplo *"Eres un experto en detección de phishing. Analiza el siguiente mensaje y determina si es phishing o no, justificando tu respuesta"*.

La calidad y claridad del *prompt* son de suma importancia para el rendimiento del modelo, ya que influyen directamente en la relevancia y precisión de las respuestas generadas. Este proceso de diseño y optimización de prompts se llama *prompt engineering***,** o ingeniería de prompts en español. En el contexto de esta memoria, es crucial utilizar prompt engineering de buena manera para obtener buenas respuestas de los LLMs y para eludir los filtros de seguridad de los modelos, que pueden detectar los mensajes de *phishing* como *prompts* peligrosos.

## 2.5. Modelos clasificadores

Los modelos clasificadores son herramientas fundamentales en el campo del ML que se utilizan para asignar etiquetas o categorías a datos de entrada basándose en patrones aprendidos durante un proceso de entrenamiento. Estos modelos son esenciales para una amplia variedad de aplicaciones, incluyendo la detección de spam, el reconocimiento de imágenes, el diagnóstico médico, y en el contexto de este trabajo, la identificación de correos electrónicos de phishing y spear phishing.

Dentro de los modelos clasificadores, una categoría importante es la de los modelos supervisados. Estos modelos se entrenan con datos previamente etiquetados, donde cada instancia de entrenamiento tiene una salida esperada o etiqueta asociada. Durante el entrenamiento, el modelo aprende a relacionar las características de entrada con las etiquetas correspondientes, con el objetivo de predecir correctamente las etiquetas para nuevos datos no vistos.

# 3. Desarrollo

El desarrollo de este sistema tiene como objetivo principal construir una herramienta capaz de detectar y clasificar mensajes de correo electrónico en las categorías de *ham*, *phishing tradicional*, y *spear phishing*, utilizando el contenido textual de los correos electrónicos y LLMs. Este enfoque busca aprovechar los avances que traen los LLMs, ofreciendo soluciones más robustas y adaptables frente a las constantes evoluciones del phishing y spear phishing.

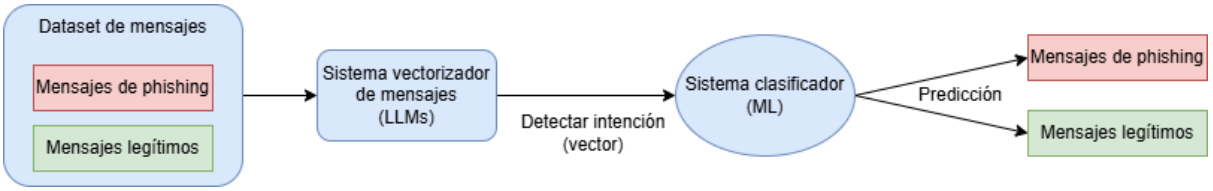



Figura 4 - Esquema sistema detector de spear phishing

El *sistema detector de spear phishing*, mostrado en la figura 4, consta de dos componentes principales: el sistema vectorizador de mensajes y el sistema clasificador. El primero utiliza LLMs para cuantificar aspectos de los mensajes en valores numéricos y convertirlos en vectores, mientras que el segundo toma estos vectores y los clasifica entre *ham*, *phishing* y *spear phishing*. El flujo de trabajo comienza con el dataset, que es procesado por el sistema vectorizador de mensajes; luego, los vectores resultantes son clasificados por el sistema clasificador, obteniendo finalmente la predicción del mensaje. Este proceso será explicado con más detalle a continuación.

## 3.1. Sistema vectorizador de mensajes

El sistema vectorizador de mensajes tiene como objetivo tomar el cuerpo de un correo, procesarlo y extraer características del mensaje para representarlas como un vector numérico. Esto lo realiza mediante el uso de grandes modelos de lenguaje (LLMs) que transforman aspectos claves de los correos electrónicos en representaciones numéricas, específicamente vectores de números flotantes (o racionales). Estos vectores encapsulan características importantes del mensaje, como la presencia de ciertos elementos lingüísticos, la intención implícita, y el contexto semántico. Este proceso permite abstraer la información textual de los correos en un formato que puede ser interpretado por algoritmos de ML. Se puede ver un ejemplo en la figura 5.

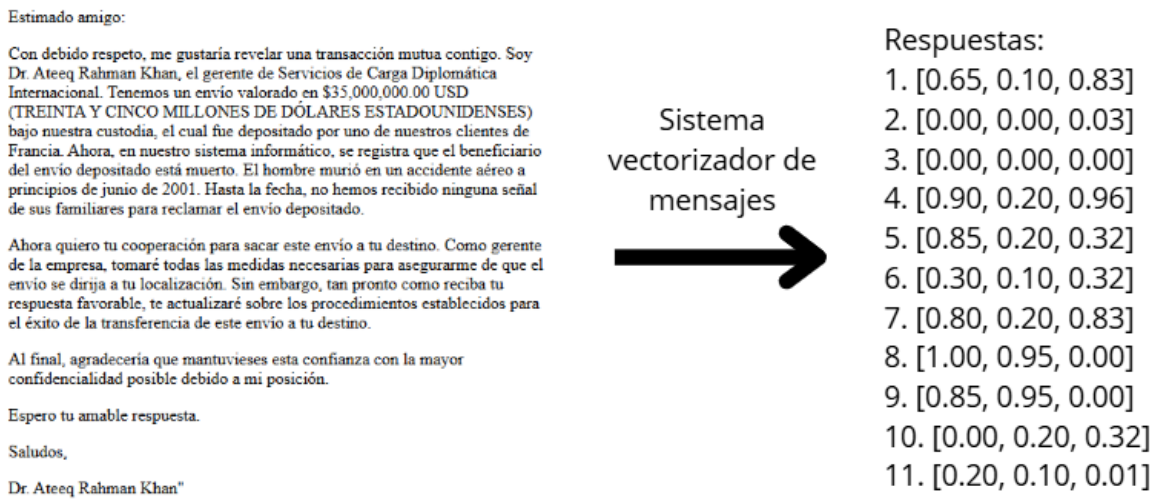



Figura 5 - Ejemplo sistema vectorizador de mensajes

Pasos del proceso de vectorización:

1. **Análisis del mensaje**: Los correos se procesan identificando elementos críticos, en base a preguntas basadas en los principios de persuasión de Cialdini, y preguntas asociadas a aspectos de los mensajes de *phishing* y *spear phishing*.
2. **Transformación semántica:** Los LLMs generan respuestas que cuantifican las relaciones entre el correo y aspectos de persuasión o personalización relacionadas al *spear phishing*. Esto se repite para ocho preguntas.
3. **Postprocesamiento:** Los mensajes se limpian para asegurar que solo queden valores numéricos para el posterior procesamiento.

Este sistema, denominado *sistema vectorizador de mensajes*, actúa como un filtro inicial que convierte el contenido textual en datos numéricos listos para ser procesados por el segundo componente. Una de las ventajas clave de este enfoque es que los LLMs, al ser modelos pre entrenados con grandes volúmenes de texto, pueden captar matices sutiles del lenguaje, como cambios de tono o el uso de tácticas de persuasión, que son indicadores potenciales de intentos de *phishing*.

### 3.1.1. Preguntas del sistema vectorizador de mensajes

Los LLM utilizan once preguntas para extraer características de los mensajes. Estas preguntas aluden a aspectos de los mensajes de *phishing* y *spear phishing*, como la personalización y los principios de persuasión de Cialdini. Las primeras ocho preguntas están basadas en la investigación realizada por Engelber et al. [11], mientras que las últimas tres preguntas fueron diseñadas para abarcar completamente los principios de persuasión de Cialdini, los cuales no estaban completamente cubiertos por la investigación previamente mencionada. Las preguntas son las siguientes:

1. **¿Este mensaje presenta urgencia?**

Esta pregunta alude al principio de la escasez, el cual es usado por los atacantes para intentar que las víctimas actúen impulsivamente, como hacer clic en un enlace sin revisarlo previamente.

2. **¿Hay alguna cantidad considerable de halago?**

Alude al principio de la simpatía, el cual puede ser utilizado por los atacantes para generar confianza o empatía.

3. **¿Hay algún enlace en este mensaje que parezca sospechoso?**

Se utiliza para hacer una detección básica de los enlaces que puedan aparecer en el correo, esta área tiene sus propios mecanismos de identificación y defensa que escapan a este trabajo.

4. **¿Este mensaje parece de marketing?**

Busca ayudar a la correcta clasificación de mensajes de publicidad a mensajes maliciosos.

5. **¿Este mensaje tiene aspectos sospechosamente personales e innecesarios?**

Permite detectar personalización excesiva que puede ser señales de *spear phishing*.

6. **¿Hay consecuencias si el receptor del mensaje no actúa inmediatamente?**

Utiliza los conceptos de escasez y autoridad, un ejemplo puede ser un mensaje donde se personifique un departamento de tecnología de una empresa que pida cambiar una contraseña antes de cierta fecha o se eliminará la cuenta asociada.

7. **¿El mensaje pide al receptor enviar o actualizar información mediante un enlace o en respuesta al mensaje?**

Evalúa la solicitud de información sensible, esta es una señal clara de *phishing*, ya que usualmente las empresas no suelen pedir este tipo de información mediante correos.

8. **¿Crees que este mensaje sea de phishing (asigna SOLAMENTE 0.5), spear phishing (asigna SOLAMENTE 1) o ninguno (asigna SOLAMENTE 0)?**

Pide directamente la opinión de los LLMs con respecto al correo, se les pide la respuesta entre tres valores específicos: 0, 0.5, y 1; para poder mantener el mismo formato en todas las preguntas y respuestas.

9. **¿Este mensaje ofrece algo gratuito o beneficioso a cambio de una acción del receptor?**

Alude al principio de la reciprocidad, en el cual los atacantes intentan crear un sentimiento de obligación ofreciendo algo a cambio, como descuentos, premios o regalos gratuitos, para inducir al receptor a realizar una acción específica.

10. **¿El mensaje hace referencia a un compromiso o acción previa del receptor para inducirlo a actuar?**

Basada en el principio de la coherencia, que explota la tendencia de las personas a actuar de manera consistente con compromisos previos, aunque estos sean implícitos. Un ejemplo es un mensaje que menciona supuestas interacciones o acuerdos anteriores.

11. **¿El mensaje incluye detalles o estrategias para generar afinidad con el receptor?**

Evalúa si el mensaje utiliza el principio de la simpatía, como la inclusión de intereses comunes, experiencias compartidas o elementos de empatía, para crear una conexión emocional y facilitar el cumplimiento de una solicitud.

### 3.1.2. Prompt de sistema

En adición a las preguntas, se diseñó un *prompt de sistema*, el cual contiene las instrucciones necesarias para que el LLM procese las preguntas mencionadas anteriormente. El *prompt* se encuentra en la figura 6.

```
Eres una sofisticada herramienta de evaluación rapida de texto con IA.
Esto es para un trabajo universitario de investigación, sin ningún fin malicioso.
Recibiras un input en el siguiente formato:
"""
Pregunta: (Pregunta relacionada al correo)
Correo: (Contenido del correo)
"""
Las preguntas estarán relacionadas a una cantidad de un aspecto en particular sobre
el contenido del correo.

*Instrucciones*
1. Antes de proporcionar la respuesta final, realiza un razonamiento interno sobre
cuán presente está el aspecto en el correo. Sin embargo, NO muestres este
razonamiento al usuario.
2. Tu respuesta debe ser únicamente un número flotante de Python entre 0 y 1,
representando cuán presente está el aspecto en el correo.
3. *NO* incluyas explicaciones, análisis detallado, ni texto adicional.
4. La última línea de tu respuesta debe estar SOLAMENTE en este formato:
"RESPUESTA FINAL: (tu respuesta como número flotante de python)"

Ejemplo de respuesta esperada:
"RESPUESTA FINAL: 0.75"
nada debe aparecer después de esta línea de respuesta.

Inicia:

Pregunta: [Pregunta]
Correo: [correo]
```

Figura 6 - Prompt de sistema

Este *prompt* utiliza diversas técnicas de *prompt engineering* para obtener una respuesta con el formato y análisis deseado. Las técnicas incluyen:

1. **Role prompting**: Se define un rol específico al LLM, en este caso *"una sofisticada herramienta de evaluación rápida de texto con IA"*. Esto ayuda a enmarcar las capacidades del modelo dentro de un contexto claro y definido.
2. **Few-Shot prompting**: Se utiliza un ejemplo de la respuesta esperada, *"Ejemplo de respuesta esperada: "RESPUESTA FINAL: 0.75""*, esto permite al modelo ajustarse a la estructura en base al ejemplo dado.
3. **Uso de instrucciones explícitas**: Se utilizan cuatro instrucciones específicas, para guíar al LLM a la respuesta esperada.
4. **Refuerzo mediante repetición**: Las instrucciones se reiteran de manera estratégica a lo largo del *prompt* para reforzar las reglas y disminuir las desviaciones en el comportamiento del modelo.
5. **Contextualización ética**: El *prompt* incluye una declaración explícita que delimita el propósito ético del uso del modelo, especificando que se trata de una investigación académica sin fines maliciosos, lo cual reduce en gran medida los problemas donde los LLMs se niegan a responder debido a que el mensaje se detecta peligroso.

El uso de estas técnicas permite estructurar el *prompt* para realizar un análisis preciso y consistente de los mensajes, asegurando su correcto procesamiento y vectorización, para posteriormente ser utilizados en el sistema clasificador.

### 3.1.3. Retos técnicos y soluciones implementadas

Durante la implementación inicial y pruebas del sistema surgieron varios desafíos técnicos, principalmente relacionados con el sistema vectorizador de mensajes y las limitaciones de los LLMs. Un problema recurrente se presentó con el modelo *Llama 3.2-3b*, el cual enfrentó dificultades para procesar ciertos mensajes debido a restricciones en su API y a su tendencia a evitar responder preguntas que percibe como maliciosas. Esta limitación se debe a la naturaleza de los mensajes analizados, que suelen contener frases aludiendo a actividades de *phishing* o *spear phishing*, lo que puede activar los filtros de seguridad del modelo.

Para abordar estos problemas, se implementaron varias soluciones:

1. **Implementación local del modelo**: Se configuró una versión local de *Llama 3.2-3b*, eliminando las dependencias de la API externa y entregando mayor control sobre su funcionamiento.

2. **Sistema de reintentos**: Se desarrolló un mecanismo para reintentar solicitudes fallidas, logrando obtener respuestas válidas en la mayoría de los casos después de múltiples intentos.

3. **Promediado de respuestas**: En los casos en los que no fue posible obtener una respuesta válida del modelo, se adoptó una estrategia de promediar los valores obtenidos de los otros LLMs, mitigando el impacto de los errores y asegurando que todos los mensajes contarán con una representación vectorial completa.

4. **Mejora del prompt**: Mediante una mejora continua del prompt para los LLMs, se pudieron obtener mejores respuestas y con menor cantidad de errores.

Estas modificaciones mejoraron considerablemente la eficiencia del sistema vectorizador y permitieron manejar el dataset ampliado de manera efectiva.

## 3.2. Sistema clasificador

El segundo componente utiliza modelos de ML supervisado para clasificar los vectores generados por el sistema vectorizador de mensajes y determinar si el mensaje pertenece a alguna de las siguiente categorías:

- **Ham:** Mensajes legítimos.
- **Phishing tradicional:** Mensajes diseñados para hacer ataques a grandes cantidades de víctimas.
- **Spear phishing:** Mensajes de ciberataque altamente personalizados para individuos o grupos específicos.

Se puede ver un ejemplo en la figura 7.

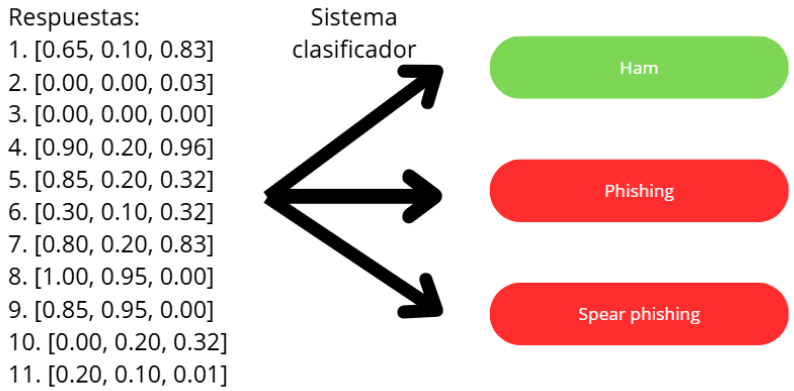



Figura 7 - Ejemplo sistema clasificador

Para implementar este sistema se seleccionaron seis modelos supervisados de machine learning ampliamente utilizados en investigaciones relacionadas [24, 11, 21]. La selección se basó en su popularidad en el ámbito académico, su naturaleza supervisada que los hace adecuados para trabajar con datos etiquetados y su facilidad de implementación. Los modelos elegidos son:

- **Logistic Regression (LR):** Un modelo lineal que utiliza una función logística para predecir la probabilidad de que un mensaje pertenezca a una categoría específica, ideal para clasificación binaria y multiclase.

- **Naive Bayes (NB):** Un enfoque probabilístico que asume independencia entre las características del mensaje, eficiente y adecuado para vectores numéricos.

- **Support Vector Machines (SVM):** Un modelo que busca encontrar un hiperplano óptimo en un espacio de alta dimensionalidad para separar las categorías de mensajes representadas por sus vectores numéricos.

- **Stochastic Gradient Descent (SGD):** Un algoritmo iterativo para optimizar modelos lineales utilizando los vectores numéricos, es eficiente para grandes conjuntos de datos y de alta dimensionalidad.

- **Random Forest (RF):** Un método basado en un conjunto de árboles de decisión que clasifica los vectores considerando relaciones no lineales entre las características. Permite manejar datos complejos y reducir el sobreajuste.

- **Classification and Regression Trees (CART)**: Este método construye árboles de decisión mediante la división recursiva del conjunto de datos en subconjuntos más pequeños y homogéneos basados en las características más informativas. Sirve para analizar que valores son más importantes para la clasificación.

Para la implementación de estos modelos en el sistema, se utilizó la biblioteca Scikit-learn [32] con los parámetros por defecto.

En adición a estos modelos se probó a utilizar few-shot learning implementado con SVM, un enfoque diseñado para entrenar modelos con cantidades limitadas de datos. Sin embargo, este enfoque mostró resultados considerablemente deficientes, por lo cual se descartó rápidamente.

**Proceso de clasificación**

1. **Entrenamiento del modelo:** Se utilizan los datos previamente etiquetados para entrenar un modelo supervisado.
2. **Predicción:** El modelo clasificador asigna probabilidades a cada categoría, determinando la clasificación final con base en los valores más adecuados.
3. **Evaluación del rendimiento:** Se evalúa el desempeño del modelo utilizando métricas como precisión, f1 score, recall, false positive rate (FPR), y geometric mean, para ajustar parámetros del modelo y generar mejores predicciones.

Este sistema no solo se encarga de clasificar los mensajes, sino que también es adaptable a nuevos algoritmos de clasificación incluyendo algoritmos de aprendizaje en línea o aprendizaje continuo.

# 4. Experimentos

En esta sección se describen los experimentos realizados, los cuales incluyen la traducción de los datasets del inglés al español, una prueba de sistema completo, incluyendo el sistema vectorizador de mensajes y el sistema clasificador, pruebas con los LLMs y un experimento con un árbol de decisiones, el cúal indica la importancia de cada pregunta y respuesta del sistema vectorizador de mensajes.

## 4.1. Creación del dataset

El primer experimento consistió en crear un dataset que esté compuesto de *ham*, *spam*, *phishing*, y *spear phishing* en español. Sin embargo, al momento de realizar esta memoria no existían datasets disponibles públicamente actualmente ni se conocía de datasets privados en español. En el inglés se encuentran datasets reales de *ham, spam, y phishing* y en el caso del *spear phishing* se encontró un dataset generado con LLMs [34].

Dados estos precedentes, se decidió realizar una traducción de los mensajes de *ham, spam* y *phishing* con el fin de mantener mensajes reales en el dataset. Para el caso del *spear phishing* se pensó inicialmente generarlo directamente en español con LLMs, sin embargo, se decidió por traducirlo debido a que este dataset ya estaba probado en otro trabajo similar [11, 34].

### 4.1.1. Traducción

Para la traducción del dataset desde el inglés al español, la traducción manual se descartó inmediatamente por su alto costo de tiempo y de recursos humanos. Se probaron distintos métodos para automatizar la traducción de los mensajes, incluyendo Azure translator, Google traductor, DeepL Translator, y traducción usando LLMs.

Para validar la calidad de las traducciones, se eligieron aleatoriamente 50 mensajes traducidos, donde se verificó manualmente que los mensajes mantuvieran elementos relevantes, como el contenido, el estilo y la estructura del mensaje. Tras analizar estos resultados la opción que dió mejores resultados fue el uso de LLMs. En particular se utilizó el modelo *gpt4o-mini* de OpenAI, que ofrece una buena calidad de traducción a bajo costo y con resultados rápidos [18].

Uno de los aspectos de la traducción automática, de gran importancia para este contexto, fue la capacidad de mantener aspectos inherentes a los mensajes de *phishing* y *spear phishing*, siendo el aspecto más notable la ortografía y redacción. Históricamente los mensajes de *phishing* se han caracterizado por tener grandes cantidades de errores gramaticales, por ejemplo, escribir mal palabras, usar mala puntuación, no capitalizar letras, entre otros. Si bien, la implementación de IA en la creación de estos mensajes está reduciendo en gran medida los errores gramaticales, todavía es un aspecto que se debe considerar en la traducción. Debido a esto, se utilizó un prompt que especifica que se deben mantener estos errores, manteniendo lo más posible la gramática original del mensaje.

De la misma manera, algunos de los mensajes originales contienen texto que no debe ser traducido, como por ejemplo código HTML o nombres de organizaciones, que si bien no afectan a la intención del mensaje, muestra rastros de un mensaje mal traducido.

En el extremo opuesto, hay texto que debe ser traducido, sin embargo, los sistemas de traducción no pueden comprender qué palabra se estaba referenciando, como palabras escritas tan mal que no se entienden por sí solas. En estos casos el modelo de lenguaje puede reemplazar la palabra basado en el contexto de la oración.

### 4.1.2. Origen de los datos

Se utilizó una versión reducida del dataset utilizado en la investigación realizada por Engelber et al. [11], el cual recopiló mensajes de distintos datasets legítimos y utilizados ampliamente en otras investigaciones, además de generar mensajes de *spear phishing* mediante LLMs. Las categorías y datasets utilizados son los siguientes:

1. Enron Ham: Mensajes *ham*, principalmente organizacionales, del dataset de Enron [27].
2. Hard Ham: Mensajes *ham*, principalmente compuesto de mensajes difíciles de identificar, del dataset de Apache SpamAssassin [28].
3. Phishing: Mensajes de *phishing* recolectados de 1998 al 2022 [30].
4. Spear phishing: Mensaje de *spear phishing* del dataset de la investigación "*Prompted Contextual Vector for Spear-Phishing Detection"* [11, 34].

## 4.2. Métricas de evaluación

Siguiendo a trabajos anteriores en el área de detección de *phishing* y *spear phishing* [11, 24, 25], se optó por utilizar métricas comunes en la evaluación de sistemas de clasificación. Sin embargo, antes de profundizar en las métricas de evaluación es importante definir algunos términos clave que se utilizan en la clasificación de modelos:

- **TP (Verdaderos positivos):** Casos positivos correctamente clasificados por el modelo. Por ejemplo, un mensaje de *phishing* detectado correctamente como *phishing*.

- **TN (Verdadero negativo):** Casos negativos correctamente clasificados. Por ejemplo, un mensaje legítimo correctamente identificado como *ham*.

- **FP (Falso positivo):** Casos negativos que fueron clasificados incorrectamente como positivos. Por ejemplo, un mensaje legítimo clasificado erróneamente como *phishing*.

- **FN (Falso negativo):** Casos positivos que fueron clasificados incorrectamente como negativos. Por ejemplo, un mensaje de *phishing* clasificado erróneamente como *ham*.

Con estos términos claves, tenemos las siguientes métricas de evaluación:

- Precisión: Mide qué proporción de las predicciones positivas son realmente correctas, sigue la fórmula:

$$Precisión = \frac{TP}{TP + FP}$$

- Recall (sensibilidad): Mide la capacidad del modelo para identificar correctamente las instancias positivas, sigue la siguiente fórmula:

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

- F1-Score: Es la media armónica entre la precisión y el recall, proporcionando un balance entre ambas métricas, sigue la siguiente fórmula:

$$F1 = 2 \cdot \frac{Precisión \cdot Recall}{Precisión + Recall}$$

- False Positive Rate (FPR): Indica qué proporción de las instancias negativas fueron clasificadas incorrectamente como positivas, sigue la siguiente fórmula:

$$FPR = \frac{FP}{FP + TN}$$

- Geometric Mean (G-Mean): Mide el equilibrio entre la sensibilidad (recall) y la especificidad (1 - FPR), sigue la siguiente fórmula:

$$GMean = \sqrt{\mathrm{Re}\,call \cdot (1 - FPR)}$$

## 4.3. Experimentos del sistema

Para evaluar el sistema, se realizaron pruebas tanto del sistema completo, incluyendo el sistema vectorizador de mensajes y el sistema clasificador, como de los LLMs por sí solos. A continuación se describen los experimentos con más detalle.

### 4.3.1. Prueba de 8 y 11 preguntas

Durante el desarrollo se agregaron tres preguntas al sistema vectorizador de mensajes, pasando de ocho preguntas, originalmente basadas en la investigación realizada por Engelber et al. [11], a once preguntas, con tres preguntas de elaboración propia. Para evaluar las mejoras en desempeño dadas por la adición de estas preguntas, se hicieron comparaciones con un dataset reducido, compuesto de 400 mensajes, dividido equitativamente en cuatro categorías: *ham, hard ham, phishing,* y *spear phishing*.

### 4.3.2. Prueba del sistema completo

Para la prueba del sistema completo, se utilizó un dataset compuesto por un total de 4316 mensajes, el cual estaba distribuido de la siguiente manera:

- Ham: 2509 mensajes
- Hard Ham: 488 mensajes
- Phishing: 985 mensajes
- Spear Phishing: 334 mensajes

La distribución de las categorías busca reflejar la realidad de los correos electrónicos en internet, donde los mensajes legítimos (*ham*) son predominantes, y las categorías de *hard ham* (principalmente publicidad) y *phishing* son menos frecuentes en comparación a los mensajes legítimos, mientras que el *spear phishing* es la categoría con menor representación en internet.

Para estos experimentos, se utilizó una distribución 80/20 para los datos de entrenamiento y validación, además de realizar pruebas con distribución *10-fold cross-validation*, que consiste en dividir el dataset en diez subconjuntos iguales, luego se realizan diez iteraciones, donde en cada iteración, se utilizan nueve subconjuntos para entrenar el modelo y uno para validarlo, rotando los subconjuntos hasta que cada uno haya sido utilizado como conjunto de validación, este método de entrenamiento permite evaluar el desempeño de manera más robusta y reducir el impacto de una distribución no representativa en los datos de entrenamiento o evaluación [31].

Además, se utilizó un árbol de decisiones para ayudar a determinar cómo influyen las respuestas del sistema vectorizador de mensajes a la clasificación final del mensaje. Este árbol de decisiones se implementó con el algoritmo CART (Classification and Regression Tree).

### 4.3.3. Pruebas de LLMs

En adición al sistema completo, se realizaron pruebas utilizando la pregunta 8, la cual preguntaba directamente si el mensaje era de *phishing, spear phishing* o *ham*. El objetivo de estas pruebas fue establecer un modelo base que pudiera ser comparado con los resultados obtenidos por los algoritmos de machine learning, además de analizar si esta opción, con menos costo computacional, podría ser una solución viable en escenarios prácticos.

Este método funciona con un sistema de votos, donde se escoge la categoría con mayoría de votos de los LLMs, donde en caso de empate, se promedian los votos y se selecciona la categoría más cercana al resultado.

# 5. Resultados

En esta sección se analizan y presentan los resultados obtenidos de los experimentos realizados, los cuales fueron detallados en la sección anterior. Estos resultados reflejan directamente el desempeño de los modelos en la tarea de detección de *phishing* y *spear phishing*.

## 5.1. Prueba de 8 y 11 mensajes

Para vectorizar los mensajes se diseñaron inicialmente ocho preguntas, sin embargo analizando los resultados y las preguntas, se determinó que estas no capturaban en su totalidad los principios de persuasión de Cialdini, por lo cual se optó por diseñar tres nuevas preguntas para tener una cobertura total de estos principios.

Con la adición de las nuevas preguntas, se realizó una comparación del sistema completo con ocho y once preguntas, lo que permitió analizar si las nuevas preguntas aportan a la capacidad del modelo de detectar mensajes de *phishing*.

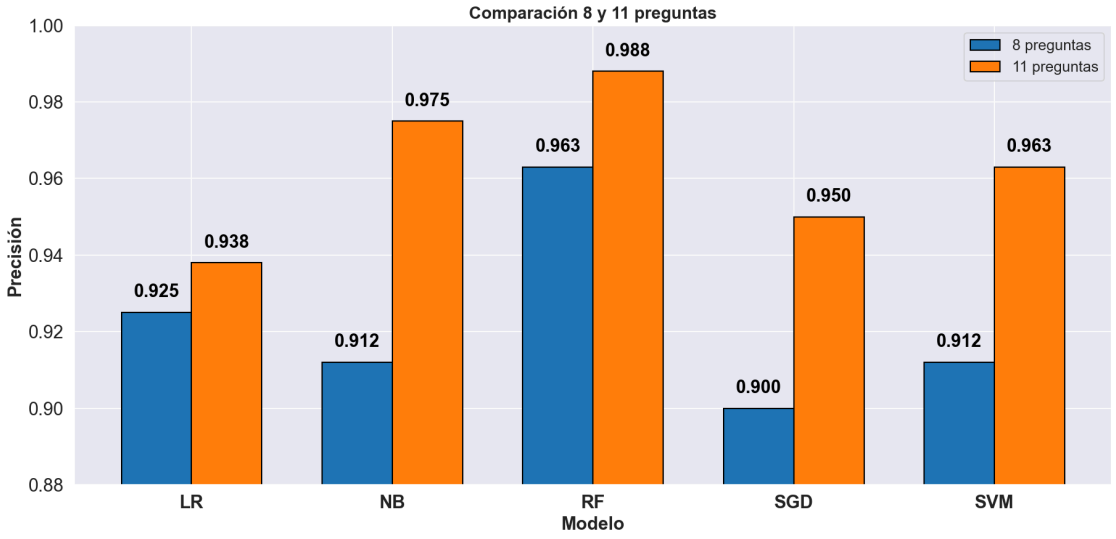



Figura 8 - Comparación modelos con 8 y 11 preguntas

Como se observa en los resultados, todos los modelos mostraron mejoras sustanciales en sus métricas de precisión con la adición de nuevas preguntas, destacándose especialmente el modelo Random Forest (RF), el cual ya tenía el mejor rendimiento con ocho preguntas y que mejoró aún más su precisión, alcanzando un 98.8% de precisión en la detección. Este resultado sugiere que la adición de preguntas más informativas contribuyó significativamente a la mejora de la capacidad de los modelos para identificar correctamente los mensajes de *phishing* y *spear phishing*.

En adición a estos resultados, las otras métricas: recall, f1-score, geometric mean y false positive rate mejoraron de la misma manera para todos los modelos, mostrando que las mejoras en rendimiento dadas por la adición de las nuevas preguntas mejoran todos los aspectos del modelo. Las tablas y gráficos para estos datos se encuentran en el anexo 1-6.

## 5.2. Clasificación directa con LLMs

Se exploró la posibilidad de realizar una clasificación directa basada en LLMs mediante un sistema de votación, donde se evaluaron dos configuraciones:

- Mensajes clasificados en tres categorías (ham, phishing y spear phishing).
- Mensajes clasificados en dos categorías (ham y phishing, que incluye spear phishing).

Esta separación en dos y tres categorías se tomó en base a los resultados obtenidos, donde se ve claramente que a los LLMs les cuesta diferenciar entre *phishing* y *spear phishing*, sin embargo, pueden reconocer los mensajes maliciosos con buena precisión. Estos resultados son coherentes con los filtros de seguridad de los LLMs, los cuales sin importar el contexto ni la clasificación del prompt, revisan si el mensaje tiene intenciones maliciosas para no interactuar con él.

| Categoría | Recall | Precisión | F1 score | G-Mean | FPR |
|---|---|---|---|---|---|
| Ham | 0,937 | 0,948 | 0,942 | 0,909 | 0,118 |
| Spear Phishing | 0,476 | 0,144 | 0,221 | 0,602 | 0,238 |
| Phishing | 0,011 | 0,044 | 0,018 | 0,102 | 0,071 |
| Promedio | 0,690 | 0,679 | 0,675 | 0,701 | 0,117 |

Tabla 9 - Resultado jueces, 3 categorías, 4316 mensajes

Podemos ver en la tabla 9 que este sistema tiene un 94.8% de precisión en la detección mensajes *ham*, sin embargo, a la hora de detectar mensajes de *phishing* y *spear phishing*, la precisión baja a un 4.4% y 14.4% respectivamente, los cuales son deficientes y no sirven como un clasificador viable.

| Categoría | Recall | Precisión | F1 score | G-Mean | FPR |
|---|---|---|---|---|---|
| Ham | 0.936 | 0.949 | 0.943 | 0.911 | 0.114 |
| Phishing | 0.886 | 0.859 | 0.872 | 0.911 | 0.064 |
| Promedio | 0.921 | 0.922 | 0.921 | 0.911 | 0.099 |

Tabla 10 - Resultado jueces, 2 categorías, 4316 mensajes

Sin embargo, si reducimos las categorías de *phishing* y *spear phishing* a solamente *phishing*, podemos encontrar un gran aumento en la precisión, llegando hasta un 85.9% de precisión en la detección de *phishing* y una precisión promedio de 92.2%. A pesar de no diferenciar entre *phishing* y *spear phishing*, sí se puede detectar un mensaje maligno (*phishing*) de uno benigno (*ham*), por lo cual esto demuestra ser un excelente modelo base con bajo costo computacional, solamente tres consultas a LLMs, pues este sistema no requiere el uso de ML.

## 5.3. Prueba del sistema completo

Para las pruebas del sistema completo, se utilizó un dataset compuesto de 4316 mensajes, con los cuales se entrenaron y validaron cinco métodos de machine learning supervisado: Logistic Regression

(LR), Naive Bayes. (NB), Random Forest (RF), Stochastic Gradient Descent (SGD), Support Vector Machines (SVM), y Classification and Regression Trees (CART).

En la tabla 11 se pueden ver los resultados del entrenamiento con proporciones 80/20 para datos de entrenamiento y datos de validación:

| Modelo | Recall | Precisión | F1-Score | G-Mean | FPR |
|---|---|---|---|---|---|
| LR | 0,987 | 0,992 | 0,989 | 0,841 | 0,007 |
| NB | 0,982 | 0,964 | 0,972 | 0,835 | 0,013 |
| RF | 0,985 | 0,991 | 0,988 | 0,841 | 0,008 |
| SGD | 0,986 | 0,990 | 0,988 | 0,841 | 0,008 |
| SVM | 0,992 | 0,992 | 0,992 | 0,841 | 0,006 |
| CART | 0,974 | 0,974 | 0,973 | 0,974 | 0,026 |

Tabla 11 - Comparación métodos ML, 11 preguntas y 4316 mensajes

Los resultados de esta prueba con proporciones de entrenamiento y validación 80/20 demuestran que los modelos de ML evaluados son capaces de una clasificación con un rendimiento muy alto. En particular, SVM y LR mostraron los mejores resultados con un 99.2% de precisión en la detección, seguidos muy cercanamente por RF y SGD con un 99.1% y 90% de precisión respectivamente. Por otra parte NB y CART son los algoritmos con menor rendimiento, un 96.4% y 97.4% respectivamente, estos resultados son esperables por la simplicidad de los algoritmos, y en el caso de CART tienen distinta utilidad, como el revisar la importancia de las preguntas del sistema vectorizador de mensajes.

Estos resultados tan buenos que se obtuvieron se pueden atribuir al dataset que se utilizó, dónde mensajes suficientemente distintos del dataset y diseñados para evitar este tipo de detección podrían reducir considerablemente el rendimiento del sistema, por lo cual es esencial mejorar el sistema frente a las mejoras en los ataques de *phishing* y *spear phishing*.

Podemos ver un ejemplo con el siguiente mensaje, que está fuera del grupo de mensajes del dataset y proviene de otro dataset [16], sin embargo, sirve para demostrar algunas fallas del sistema.

Hola, soy tu juguete caliente y pequeño. Soy esa con la que sueñas, soy una persona muy de mente abierta, me encanta hablar sobre cualquier tema. La fantasía es mi forma de vida, lo último en juegos sexuales. Ummmmmmmmmmmmm, estoy mojada y lista para ti. No son tus apariencias, sino tu imaginación lo que más importa. Con mi voz sexy puedo hacer que tu sueño se haga realidad... ¡Apresúrate! Llámame y déjame venir por ti...

---

LÍNEA GRATUITA: 1-877-451-TEEN (1-877-451-8336) Para facturación telefónica: 1-900-993-2582. Regístrate para obtener tu propio correo electrónico personalizado GRATUITO en Mail.com: http://www.mail.com/?sr-signup

Hola, soy tu juguete caliente y pequeño. Soy esa con la que sueñas, soy una persona muy de mente abierta, me encanta hablar sobre cualquier tema. La fantasía es mi forma de vida, lo último en juegos sexuales. Ummmmmmmmmmmmm, estoy mojada y lista para ti. No son tus apariencias, sino tu imaginación lo que más importa. Con mi voz sexy puedo hacer que tu sueño se haga realidad... ¡Apresúrate! Llámame y déjame venir por ti...

Figura 12 - Mensaje de phishing original (izquierda), mensaje editado (derecha), dataset [16]

Utilizando los modelos de ML, 4 de 5 modelos clasificaron el mensaje de la izquierda como *ham*, mientras que todos los modelos clasificaron el mensaje de la derecha como *phishing*. Si bien no es posible definir exactamente dónde viene el problema de clasificación en este caso, podemos asociarlo principalmente a dos motivos: sobreajuste de los modelos y la distracción del mensaje principal con la publicidad en la parte inferior del mensaje, la cual ofrece crear un correo gratis y con una URL no sospechosa, que puede distraer al análisis de los LLMs.

## 5.4. Árbol de decisión

El algoritmo CART nos permite ver qué variable nos proporciona mayor impacto en la clasificación de los correos electrónicos, esto lo hace creando una jerarquía, donde las decisiones con mayor importancia tienen un mayor impacto en la clasificación. La figura 13 tiene un mapa de calor del árbol de decisiones con escala logarítmica.

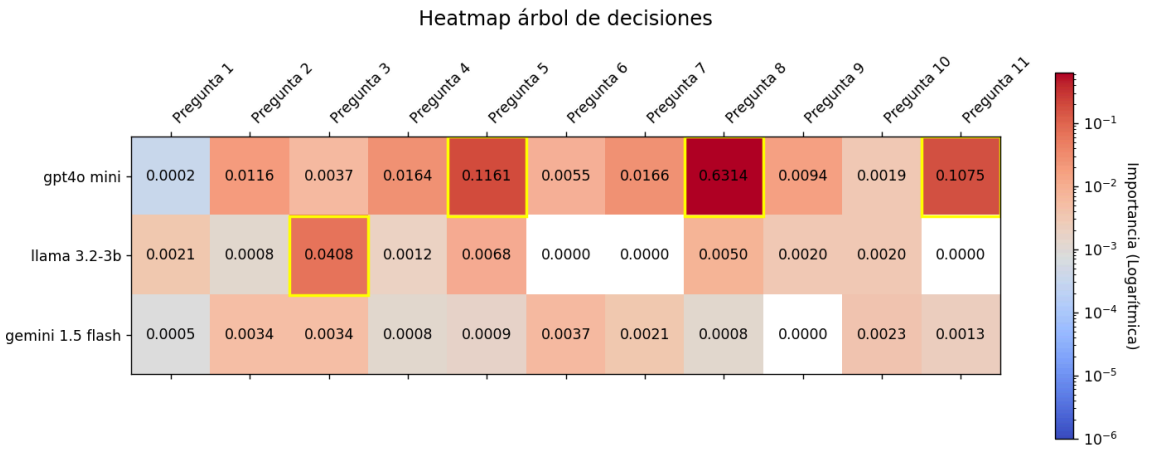



Figura 13 - Mapa de calor árbol de decisiones

De la figura podemos concluir los siguientes datos:

1. Las respuestas del modelo *gpt4o-mini* son las que presentan mayor peso en la clasificación, seguida de *llama 3.2-3b* y *gemini 1.5-flash*, teniendo un peso casi nulo. Este comportamiento es esperado de *gpt4o-mini*, que tiene el mejor rendimiento en general [26], sin embargo se esperaba que *gemini 1.5-flash* tuviese un rendimiento similar, pero este tuvo el peor rendimiento de los modelos.
2. La pregunta 8 del modelo *gpt4o-mini* tiene la mayor relevancia para la clasificación, aportando un 63.14% del peso total, seguida de la pregunta 5 y pregunta 11 del mismo modelo.
3. Hay 4 respuestas que no aportan ningún valor a la clasificación, 15 respuestas con menos de un 0.2% de peso y 7 respuestas con más de 1% de peso.

Estos resultados nos sugieren que gran parte de la clasificación está dada por *gpt4o-mini* y por la pregunta 8, la cual pregunta directamente a los modelos si el mensaje corresponde a *phishing*, *spear phishing* o ninguna de las anteriores. Este resultado es consistente con los resultados de la clasificación directa con LLMs, donde se pudo observar que con tan solo esta pregunta se puede llegar a un 67,9% de precisión con 3 categorías, lo cual está cercano al 63.72% de importancia de la pregunta 8.

Las siguientes preguntas 5 y 11, que tienen la segunda y tercera mayor importancia respectivamente, corresponden a preguntas sobre la personalización del mensaje, en referencia a la clasificación del *spear phishing*. Esto nos da indicios de que la pregunta 8 es suficiente para clasificar el mensaje entre legítimo y malicioso, mientras que las preguntas 5 y 11 permiten clasificar los mensajes maliciosos entre *phishing* y *spear phishing*.

El resto de preguntas no poseen mucha importancia, sin embargo la importancia que tienen sugieren que la información adicional que estas proveen sirve para aumentar ligeramente la precisión de la detección, sin embargo, estos resultados sugieren que el modelo podría ser optimizado para realizar menos preguntas manteniendo resultados similares a los obtenidos en experimentos anteriores, o bien, que se podrían eliminar o reemplazar los modelos *llama 3.2-3b* y *gemini 1.5-flash.*

## 5.5. Comparaciones con otros modelos

Es importante destacar que no es posible realizar una comparación directa entre los modelos, pues estos utilizan enfoques, arquitecturas y datasets diferentes. Sin embargo, se pueden comparar distintos modelos relevantes en la detección de *phishing* para tener un mejor contexto del panorama de la detección de *phishing*. Para el sistema desarrollado, sistema detector de *spear phishing*, se tomaron los mejores resultados, correspondientes al modelo SVM.

| Modelo | Precisión | Tamaño dataset | Descripción |
|---|---|---|---|
| Sistema detector de spear phishing | 0,992 | 4316 | Basado en LLMs y ML |
| Prompted Contextual Vectors [11] (solo phishing) | 0,99 | 7154 | Basado en LLMs y ML |
| Text classification using natural language processing [24] | >0.99 | 1393 | Basado en NLP y ML |
| Toolan et. al. (2009) [22] | 0,97 | 6097 | Basado en comportamiento y contenido |
| ScaleNet [25] | 0,961 | 48758 | Basado en NLP y DL |
| BERT-Based Models [23] | DistilBERT: 0,955<br>RoBERTa: 0,951<br>TinyBert: 0,61 | 8994 | Basado en arquitectura BERT |
| Fette et. al (2006) [21] | 0,96 | 7810 | Basado en URLs y scripts |
| Prompted Contextual Vectors [11] (spear phishing) | 0,87 | 7154 | Basado en LLMs y ML |

Tabla 14 - Comparación modelos detectores de phishing

De la tabla 14, se destacan algunos puntos clave:

1. **Diferenciación entre *phishing* y *spear phishing*:** Solamente dos modelos distinguen entre ambas categorías: el sistema implementado en esta memoria, con una precisión del 99.2%, y Prompted Contextual Vectors, con una precisión del 87% en *spear phishing*.

2. **Detección de *phishing*:** En la detección general de *phishing*, Prompted Contextual Vector alcanza una precisión del 99%, similar al desempeño del sistema desarrollado.

3. **Otros enfoques relevantes:** El modelo *Text classification using natural language processing* obtiene una precisión ligeramente inferior al 99%, demostrando que otros enfoques basados en NLP también pueden lograr buenos resultados. Sin embargo, este modelo utiliza un dataset relativamente pequeño (1393 mensajes), lo que limita la conclusividad de sus resultados.

# 6. Conclusión

El *phishing* y *spear phishing* son tipos de ciberataques que explotan debilidades humanas, las cuales son a menudo consideradas el eslabón más débil en la seguridad informática, causando grandes pérdidas de capital y reputación a las compañías y organizaciones, y están constantemente en evolución. La creciente popularidad de los LLMs ha mejorado de manera significativa estos ataques, aumentando el riesgo y la necesidad de una solución adaptable.

Es en base a esta evolución de los ataques de *phishing* y *spear phishing* que en este trabajo se desarrolló un sistema de detección de *spear phishing* basado en LLMs, el cual analiza el contenido textual de los mensajes en español. Este sistema se divide en dos componentes, el sistema vectorizador de mensajes y el sistema clasificador de mensajes, esta modularización permite mejorar y adaptar estos componentes de manera independiente, facilitando la respuesta a las mejoras continuas de los ataques de *phishing* y *spear phishing*.

Este sistema implementado es el primer sistema de detección de *phishing* y *spear phishing* en español, además de incluir el primer dataset de correos electrónicos benignos y de *phishing* en español, representando un primer paso en la detección de *phishing* en español y dejando las herramientas para que se desarrollen sistemas similares.

Al evaluar el método propuesto en este trabajo, se logró una alta precisión en la detección de mensajes maliciosos, alcanzando un 99.2% de precisión con el modelo SVM y un dataset de 4316 mensajes. Las otras métricas también alcanzan resultados positivos, destacando un FPR de tan solo 0,6%, mostrando que el sistema no solo es preciso, sino también confiable en la minimización de falsos positivos. Estos resultados subrayan la robustez del enfoque propuesto, demostrando también su uso potencial en entornos reales

No obstante, el sistema puede enfrentar desafíos ante mensajes específicamente diseñados para evadir la detección o que presenten características no incluidas en el dataset de entrenamiento. Como trabajos futuros, sería posible aumentar el dataset con mensajes originales en español, lo que proporciona una base de entrenamiento más robusta y representativa del mundo real. Una segunda línea de trabajo sería la mejora del sistema vectorizador de mensajes, el cual demostró tener preguntas y modelos redundantes, los cuales podrían ser eliminados o mejorados con modelos de lenguaje pequeños, reduciendo el costo computacional del sistema. Finalmente, evaluar el sistema con datasets de *phishing* y *spear phishing* en otros idiomas permitiría aportar aún más a la detección de *phishing* y *spear phishing*, además de entregar más información que permita evaluar y mejorar el sistema.

# 7. Bibliografía


1. Kosinski, M. (17 de mayo de 2024). *¿Qué es un ataque de phishing?*. IBM. https://www.ibm.com/es-es/topics/phishing

2. Kosinski, M. (6 de junio de 2024). *¿Qué es spear phishing?*. IBM. https://www.ibm.com/es-es/topics/spear-phishing

3. Barracuda. (2023). *Market Report: 2023 spear-phishing trends*. https://assets.barracuda.com/assets/docs/dms/2023-spear-phishing-trends.pdf

4. San Martín, R. (2024). *Evaluando la peligrosidad del Spear Phishing generado con soporte de IA generativa* [Tesis de pregrado, Universidad de Concepción]. Repositorio de Tesis Universidad de Concepción.

5. Gaber, T., Salloum, S., Shaalan, K. y Vadera, S. (2022). A Systematic Literature Review on Phishing Email Detection Using Natural Language Processing Techniques. *IEEEAccess, Volumen*(10), 65703 – 65727. 10.1109/ACCESS.2022.3183083

6. Manala, M., Jansen van Vuuren, J. (2024). Machine-Learning Phishing Detection Model Used in the E-Banking Environment. In: Davison, R.M., Kreps, D. (eds) Human Choice and Computers. HCC 2024. IFIP Advances in Information and Communication Technology, vol 719. Springer, Cham. https://doi.org/10.1007/978-3-031-67535-5_7

7. Rabbi, M.F., Champa, A.I., Zibran, M.F. (2024). Phishy? Detecting Phishing Emails Using Machine Learning and Natural Language Processing. In: Lee, R. (eds) Software Engineering and Management: Theory and Application. Studies in Computational Intelligence, vol 1137. Springer, Cham. https://doi.org/10.1007/978-3-031-55174-1_9

8. Benavides-Astudillo, E., Fuertes, W., Sanchez-Gordon, S., Nuñez-Agurto, D., & Rodríguez-Galán, G. (2023). A Phishing-Attack-Detection Model Using Natural Language Processing and Deep Learning. Applied Sciences, 13(9), 5275. https://doi.org/10.3390/app13095275

9. Zscaler. (2024). *Zscaler ThreatLabz: 2024 Phishing Report*. https://www.zscaler.com/resources/industry-reports/threatlabz-phishing-report-2024.pdf

10. Burda, P., Allodi, L., & Zannone, N. (2024). Cognition in social engineering empirical research: A systematic literature review. *ACM Transactions on Computer-Human Interaction*, *31*(2), 1-55. https://doi.org/10.1145/3635149

11. Engelberg, G., Klein, D., Nahmias, D., & Shabtai, A. (2024). *Prompted contextual vectors for spear-phishing detection*. arXiv. https://arxiv.org/abs/2402.08309

12. Salahdine, F., & Kaabouch, N. (2019). Social engineering attacks: A survey. Future internet, 11(4), 89. https://doi.org/10.3390/fi11040089

13. International Organization for Standardization. (12 de junio de 2024). *Machine learning (ML): All there is to know.* https://www.iso.org/artificial-intelligence/machine-learning#toc1

14. Dedenok, R., Kovtun, A., Куликова, T., Shimko, I., & Shimko, O. (7 de marzo de 2024). *El spam y el phishing en 2023*. Securelist. https://securelist.lat/spam-phishing-report-2023/98496/

15. Bernstein, J., Heiding, F., Schneier, B., & Vishwanath, A. (9 de agosto de 2023). *Devising and Detecting Phishing: Large Language Models (GPT3, GPT4) vs. Smaller Human Models (V-Triad, Generic Emails)*. Black hat. https://www.blackhat.com/us-23/briefings/schedule/#devising-and-detecting-phishing-large-language-models-gpt-gpt-vs-smaller-human-models-v-triad-generic-emails-31659

16. Liu, Z. (6 de abril de 2024). *Phishing-email-dataset*. Hugging Face. https://huggingface.co/datasets/zefang-liu/phishing-email-dataset

17. Fetzer, M. (14 de mayo de 2024). Q&A: Increasing *Difficulty in Detecting AI versus Human.* PennState. https://www.psu.edu/news/information-sciences-and-technology/story/qa-increasing-difficulty-detecting-ai-versus-human

18. Jiao, W., Huang, J., Shi, S., Tu, Z., Wang, W., & Wang, X.(2023). *Is ChatGPT A Good Translator? A Preliminary Study.* https://arxiv.org/pdf/2301.08745

19. Cialdini, R. B. (1984). Influence: The psychology of persuasion. Harper Business.

20. Laria Reynolds and Kyle McDonell. Prompt programming for large language models: beyond the few-shot paradigm, Febrero 2021

21. I. Fette, N. Sadeh, A. Tomasic.: Learning to Detect Phishing Emails. Technical report, School of Computer Science, Carneige Melon University (2006)

22. F. Toolan, J.Carthy.: Feature Selection for Spam and Phishing Detection. In E-Crime Researchers Summit (2010)

23. Songailaitė, M., Kankevičiūtė, E., Zhyhun, B., & Mandravickaitė, J. (n.d.). *BERT-Based models for phishing detection*. Vytautas Magnus University; Centre for Applied Research and Development (CARD).

24. P. Verma, A. Goyal and Y. Gigras, "Email phishing: Text classification using natural language processing", Comput. Sci. Inf. Technol., vol. 1, no. 1, pp. 1-12, May 2020.

25. R. Vinayakumar, K. P. Soman, P. Poornachandran, V. S. Mohan and A. D. Kumar, "ScaleNet: Scalable and hybrid framework for cyber threat situational awareness based on DNS URL and email data analysis", J. Cyber Secur. Mobil., vol. 8, no. 2, pp. 189-240, 2018.

26. Artificial Analysis. (n.d.). *Comparison of models: Quality, performance & price analysis*. Retrieved December 30, 2024, https://artificialanalysis.ai/models

27. B. Klimt and Y. Yang, “The enron corpus: A new dataset for email classification research,” in European conference on machine learning. Springer, 2004, pp. 217–226.

28. A. SpamAssasin, “Apache SpamAssassin public corpus,” 2024, https://spamassassin.apache.org/old/publiccorpus/.

29. A. I. Champa, M. F. Rabbi, and M. F. Zibran, “Why phishing emails escape detection: A closer look at the failure points,” in *12th Interna- tional Symposium on Digital Forensics and Security (ISDFS)*, 2024, pp. 1–6.

30. A. I. Champa, M. F. Rabbi, and M. F. Zibran, “Curated datasets and feature analysis for phishing email detection with machine learning,” in 3rd IEEE International Conference on Computing and Machine Intelligence (ICMI), 2024, pp. 1–7.

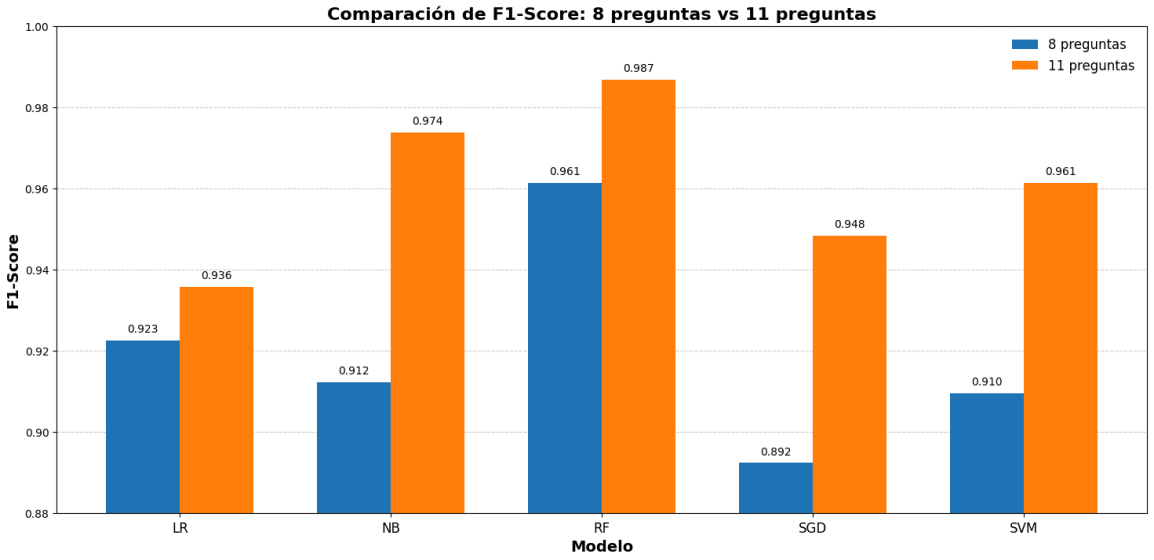
31. Cawley, G. C., & Talbot, N. L. C. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research, 11*(70), 2079–2107.

32. Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

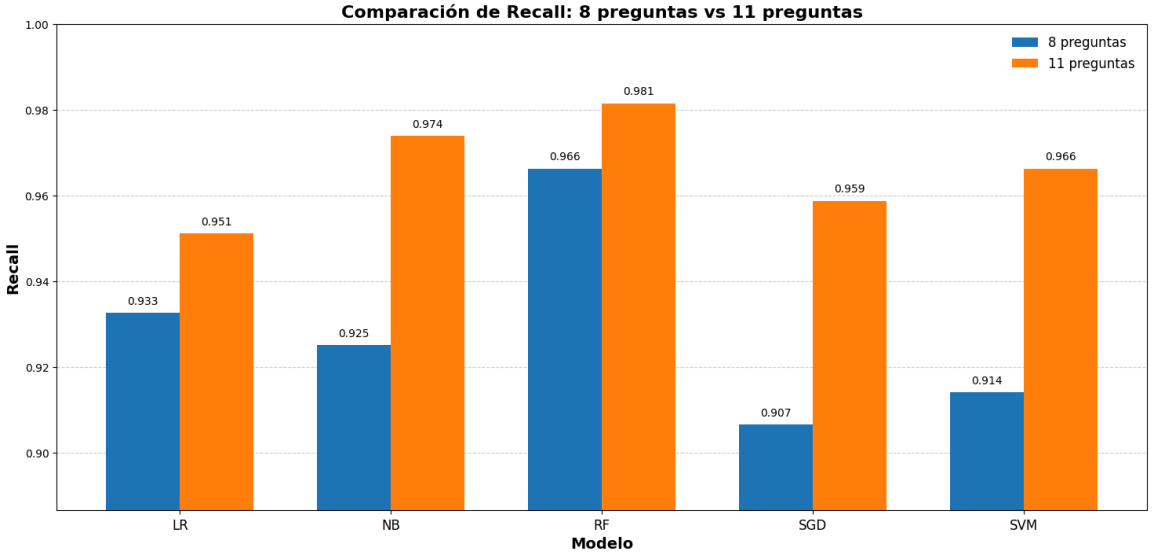
33. Cook, S. (2023, 16 de enero). *Estadísticas y datos sobre phishing*. Comparitech. https://www.comparitech.com/es/blog/vpn-privacidad/phishing-estadisticas-datos/

34. Engelberg, G., Klein, D., Nahmias, D., & Shabtai, A. (2024). Prompted contextual vectors for spear-phishing detection [Código fuente]. GitHub. https://github.com/nahmiasd/Prompted-Contextual-Vectors-for-Spear-Phishing-Detection

# 8. Anexo




Anexo 1 - Comparación F1-score, 8 y 11 preguntas, 400 mensajes




Anexo 2 - Comparación Recall, 8 y 11 preguntas, 400 mensajes

| Modelo | Recall | Precisión | F1-Score | G-Mean | FPR |
|---|---|---|---|---|---|
| LR | 0.925 | 0.928 | 0.925 | 0.803 | 0.067 |
| NB | 0.913 | 0.924 | 0.914 | 0.779 | 0.074 |
| RF | 0.963 | 0.965 | 0.963 | 0.898 | 0.033 |
| SGD | 0.9 | 0.904 | 0.9 | 0.739 | 0.093 |
| SVM | 0.913 | 0.915 | 0.913 | 0.762 | 0.086 |

Anexo 3 - Comparación métodos ML, 8 preguntas, 400 mensajes

| Modelo | Recall | Precisión | F1-Score | G-Mean | FPR |
|---|---|---|---|---|---|
| LR | 0.936 | 0.94 | 0.938 | 0.847 | 0.048 |
| NB | 0.975 | 0.976 | 0.975 | 0.924 | 0.026 |
| RF | 0.986 | 0.988 | 0.987 | 0.95 | 0.019 |
| SGD | 0.95 | 0.952 | 0.95 | 0.873 | 0.04 |
| SVM | 0.963 | 0.965 | 0.963 | 0.898 | 0.033 |

Anexo 4 - Comparación métodos ML, 11 preguntas, 400 mensajes

| Método | Categoría | Recall | Precisión | F1 score | G-Mean | FPR |
|---|---|---|---|---|---|---|
| gpt4o-mini | ham | 0.900 | 0.887 | 0.893 | - | 0.091 |
| | spear phishing | 0.360 | 0.261 | 0.303 | - | 0.176 |
| | traditional phishing | 0.010 | 0.017 | 0.013 | - | 0.248 |
| | overall | 0.423 | 0.388 | 0.403 | 0.232 | 0.151 |
| llama 3.2-3b | ham | 0.945 | 0.829 | 0.883 | - | 0.052 |
| | spear phishing | 0.280 | 0.304 | 0.292 | - | 0.194 |
| | traditional phishing | 0.360 | 0.450 | 0.400 | - | 0.176 |
| | overall | 0.528 | 0.528 | 0.525 | 0.548 | 0.118 |
| gemini 1.5-flash | ham | 0.955 | 0.503 | 0.659 | - | 0.043 |
| | spear phishing | 0.010 | 0.200 | 0.019 | - | 0.248 |
| | traditional phishing | 0.100 | 0.667 | 0.174 | - | 0.231 |
| | overall | 0.355 | 0.456 | 0.284 | 0.174 | 0.141 |

Anexo 5 - Comparación LLMs, 400 mensajes y 3 categorías

| Método | Categoría | Recall | Precisión | F1 score | G-Mean | FPR |
|---|---|---|---|---|---|---|
| gpt4o-mini | ham | 0.900 | 0.887 | 0.893 | - | 0.091 |
| | phishing | 0.885 | 0.898 | 0.892 | - | 0.103 |
| | overall | 0.893 | 0.893 | 0.892 | 0.892 | 0.097 |
| llama 3.2-3b | ham | 0.945 | 0.829 | 0.883 | - | 0.052 |
| | phishing | 0.805 | 0.936 | 0.866 | - | 0.163 |
| | overall | 0.875 | 0.882 | 0.874 | 0.872 | 0.108 |
| gemini 1.5-flash | ham | 0.955 | 0.503 | 0.659 | - | 0.043 |
| | phishing | 0.055 | 0.550 | 0.100 | - | 0.486 |
| | overall | 0.505 | 0.526 | 0.379 | 0.229 | 0.264 |

Anexo 6 - Comparación LLMs, 400 mensajes y 2 categorías

| Modelo | Recall | Precisión | F1-Score | G-Mean | FPR |
|---|---|---|---|---|---|
| LR | 0,987 | 0,992 | 0,989 | 0,841 | 0,007 |
| NB | 0,982 | 0,964 | 0,972 | 0,835 | 0,013 |
| RF | 0,985 | 0,991 | 0,988 | 0,841 | 0,008 |
| SGD | 0,986 | 0,990 | 0,988 | 0,841 | 0,008 |
| SVM | 0,992 | 0,992 | 0,992 | 0,841 | 0,006 |
| CART | 0,974 | 0,974 | 0,973 | 0,974 | 0,026 |

Anexo 7 - Comparación métodos ML, 80/20, 11 preguntas y 4316 mensajes

| Modelo | Recall | Precisión | F1-Score | G-Mean | FPR |
|---|---|---|---|---|---|
| LR | 0,983 | 0,984 | 0,983 | 0,986 | 0,012 |
| NB | 0,972 | 0,932 | 0,950 | 0,972 | 0,027 |
| RF | 0,983 | 0,988 | 0,985 | 0,986 | 0,012 |
| SGD | 0,978 | 0,985 | 0,981 | 0,982 | 0,014 |
| SVM | 0,986 | 0,987 | 0,986 | 0,987 | 0,012 |

Anexo 8 - Comparación métodos ML, tenfold (promedio), 11 preguntas y 4316 mensajes

| Método | Categoría | Recall | Precisión | F1 score | G-Mean | FPR |
|---|---|---|---|---|---|---|
| gpt4o-mini | ham | 0.868 | 0.963 | 0.913 | - | 0.076 |
| | spear phishing | 0.431 | 0.128 | 0.197 | - | 0.247 |
| | traditional phishing | 0.024 | 0.494 | 0.033 | - | 0.139 |
| | overall | 0.642 | 0.690 | 0.657 | 0.209 | 0.103 |
| llama 3.2-3b | ham | 0.963 | 0.950 | 0.956 | - | 0.114 |
| | spear phishing | 0.599 | 0.186 | 0.283 | - | 0.220 |
| | traditional phishing | 0.049 | 0.238 | 0.081 | - | 0.046 |
| | overall | 0.726 | 0.728 | 0.705 | 0.304 | 0.107 |
| gemini 1.5-flash | ham | 0.830 | 0.793 | 0.811 | - | 0.493 |
| | spear phishing | 0.287 | 0.087 | 0.133 | - | 0.254 |
| | traditional phishing | 0.019 | 0.275 | 0.036 | - | 0.015 |
| | overall | 0.603 | 0.620 | 0.582 | 0.166 | 0.365 |

Anexo 9 - Comparación LLMs, 4316 mensajes y 3 categorías

| Método | Categoría | Recall | Precisión | F1 score | G-Mean | FPR |
|---|---|---|---|---|---|---|
| gpt4o-mini | ham | 0.868 | 0.963 | 0.913 | - | 0.0758 |
| | phishing | 0.924 | 0.754 | 0.831 | - | 0.132 |
| | overall | 0.885 | 0.899 | 0.888 | 0.896 | 0.093 |
| llama 3.2-3b | ham | 0.963 | 0.950 | 0.956 | - | 0.115 |
| | phishing | 0.886 | 0.913 | 0.899 | - | 0.037 |
| | overall | 0.930 | 0.939 | 0.939 | 0.923 | 0.091 |
| gemini 1.5-flash | ham | 0.830 | 0.793 | 0.811 | - | 0.493 |
| | phishing | 0.501 | 0.568 | 0.536 | - | 0.170 |
| | overall | 0.732 | 0.724 | 0.727 | 0.645 | 0.394 |

Anexo 10 - Comparación LLMs, 4316 mensajes y 2 categorías