



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE MÁSTER

*Estudio del uso de modelos de lenguaje LLM en el ámbito de la
Defensa*

Máster Universitario en Dirección TIC para la Defensa

ALUMNO: Norberto Pablo González Fernández

DIRECTORA: Milagros Fernández Gavilanes.

CURSO ACADÉMICO: 2024-2025

UniversidadeVigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE MÁSTER

*Estudio del uso de modelos de lenguaje LLM en el ámbito de la
Defensa*

Máster Universitario en Dirección TIC para la Defensa
Especialidad de Sistemas y Tecnologías de la Información

UniversidadeVigo

RESUMEN

La inteligencia artificial se viene desarrollando desde finales de la Segunda Guerra Mundial en diversos ámbitos como la medicina, la economía, el transporte o la agricultura entre otros, pero la explosión vivida a finales de 2022 con la expansión de la denominada inteligencia artificial generativa (IAG) y especialmente la aparición de ChatGPT ha supuesto un antes y un después en el uso masivo de la inteligencia artificial y de los LLM's.

Los LLM's, por sus siglas en inglés *Large Language Model*, son actualmente el máximo exponente de la IA, gracias a los avances en las técnicas de *machine learning*. Sus algoritmos son capaces de generar o reconocer imágenes, generar música, textos, videos, pueden sostener diálogos, ofrecer respuestas elaboradas y mantener conversaciones coherentes en diferentes idiomas, como haría un humano.

El campo de los LLM's es muy amplio y ha permitido avances en muchos campos de la ciencia y de la técnica, sin embargo, como en todo desarrollo, siempre hay un lado oscuro. Todas estas capacidades también pueden ser utilizadas para fines fraudulentos, para cometer delitos usando estas tecnologías o realizar acciones moralmente reprochables.

Utilizando los trabajos e investigaciones que existen actualmente, se pretende hacer una síntesis de aquellos que pueden tener una aplicación práctica en el ámbito de la Defensa. No todos los trabajos tienen una aplicación directa y dado lo novedoso de estas nuevas tecnologías, es necesario no perder el tren, prepararse para adoptarlas, formar al personal experto y concienciar a todos los usuarios del Ministerio de Defensa como usuarios de sistemas TIC.

Aunque se pueda considerar un desarrollo muy reciente, ya existen investigaciones y estudios relacionados con la aplicación de los LLM's en los ámbitos de la ciberseguridad y la ciberdefensa de los que se pueden extraer valiosas lecciones y líneas de trabajo. El *phishing*, la protección de contraseñas, el *pentesting* automatizado o la calidad del código programado entre otros, son problemas que afectan a todos por igual, particulares, empresas y organismos públicos, conocer el uso y manejo de los LLM's, conocer todas sus posibles aplicaciones y aprovechar las ventajas que nos pueden aportar, puede ser vital para anticiparse y hacer frente a las amenazas que se vislumbran en el futuro cercano.

PALABRAS CLAVE

LLM, Inteligencia Artificial generativa, Machine Learning, Arquitectura de transformadores

AGRADECIMIENTOS

A mi tutora de máster Milagros Fernández Gavilanes, por sus correcciones y sugerencias, por su constante guía y crítica constructiva, sin su tutoría este trabajo no habría salido adelante. A los profesores del CUD que nos han impartido clase durante estos casi dos años. A la Escuela Naval Militar por el trato que nos han dispensado durante el desarrollo de las fases presenciales y por ultimo y muy especialmente a mis compañeros de máster, el tiempo que hemos compartido en las fases presenciales, ha sido sin duda lo mejor que me llevo como experiencia de esta larga travesía.

CONTENIDO

Contenido	6
Índice de Figuras	8
Índice de Tablas.....	9
1 Introducción y objetivos	10
1.1 Introducción	10
1.2 Objetivos	12
1.3 Estructura	12
2 Estado del arte	14
2.1 La explosión de la IA generativa y ChatGPT	14
2.2 Evolución histórica de la IA.....	18
2.3 Desarrollo de los LLM's	20
2.4 Conceptos de IA y tipos de aprendizaje.....	21
2.4.1 Conceptos de IA	21
2.4.2 Tipos de aprendizaje	22
2.5 Conceptos sobre deep learning y tipos de modelos.	25
2.5.1 Las redes neuronales convolucionales (CNN o ConvNets).....	27
2.5.2 Redes neuronales recurrentes (RNN)	28
2.5.3 Autocodificadores y autocodificadores variacionales	29
2.5.4 Las redes generativas antagónicas (GAN).....	30
2.5.5 Modelos de difusión	30
2.5.6 Modelos de transformadores.....	31
3 La revolución de los LLM's: desde las RNN's hasta los Transformadores.....	34
3.1 Funcionamiento de las redes neuronales recurrentes	34
3.2 Long short-term memory (LSTM)	35
3.3 Introducción a la arquitectura codificador-descodificador	36
3.4 Arquitectura de transformadores.....	38
3.4.1 Componentes clave de un transformador.	39
3.4.2 Funcionamiento de los transformadores	40
3.4.3 Ventajas de los transformadores sobre modelos previos	41
3.5 Familias de LLM's actuales	42
3.5.1 Familia BERT	42
3.5.2 Familia GPT.....	42
3.5.3 Modelos derivados de GPT.	43
3.5.4 Modelos de cálculo óptimo.....	44

3.5.5 La familia Llama.....	44
3.5.6 Modelos de secuencia a secuencia.....	45
3.5.7 Familia PaLM	45
3.5.8 GPT-4 y BingChat	45
4 Casos de uso genéricos y posibles implementaciones en Defensa.....	47
4.1 Usos de los LLM's	47
4.2 Casos de uso generales más frecuentes	47
4.3 Casos de uso en el ámbito de la Defensa.	49
4.3.1 Detección y análisis de amenazas	53
4.3.2 Automatización de la seguridad.....	53
4.3.3 Detección y respuesta al phishing.....	53
4.3.4 Análisis forense:	54
4.3.5 Pruebas de penetración:	54
4.3.6 Respuesta automática a incidentes:.....	54
4.3.7 Chatbots de seguridad:.....	54
4.3.8 Capacitación y concientización sobre seguridad:	54
4.3.9 Modelado de contraseñas:.....	54
4.3.10 Creación de Honeypot	55
4.3.11 IDS (Intrusion Detection System)	55
4.4 Las principales vulnerabilidades de los LLM's	63
4.5 Recomendaciones sobre posibles implementaciones.....	65
5 Conclusiones y líneas futuras	67
5.1 Conclusiones	67
5.2 Líneas futuras.....	68
6 Bibliografía.....	70
Anexo I: Algoritmos de aprendizaje más utilizados.....	90
Anexo II: Listado de LLM's open source	92
Anexo III: Listado cve que afectan a los LLM's.....	99

ÍNDICE DE FIGURAS

Ilustración 1: Meses que han tardado algunas apps en tener un millón de usuarios. Fuente [12]....	14
Ilustración 2: Días que ha tardado un servicio online en llegar al millón de usuarios. Fuente [14]	14
Ilustración 3: Aumento del número de parámetros de los LLM's desde 2020 -2024. Fuente [15]..	15
Ilustración 4: Recorte de algunos LLM's y sus características: Fuente [16]	17
Ilustración 5: Ejemplo de la neurona de McCulloch y Pitts. Fuente [30]	18
Ilustración 6: Relación IA, ML y DL. Fuente [79].....	21
Ilustración 7: Tipos de aprendizaje automático: Fuente [81]	22
Ilustración 8: Aprendizaje supervisado. Fuente [84].....	23
Ilustración 9: Aprendizaje No Supervisado. Fuente [84]	23
Ilustración 10: Aprendizaje Semi-supervisado. Fuente [90]	24
Ilustración 11: Aprendizaje por Refuerzo. Fuente [84].....	25
Ilustración 12: Red neuronal artificial. Fuente [96]	25
Ilustración 13: Red neuronal artificial frente a una red neuronal profunda. Fuente [100].....	26
Ilustración 14. Representación de la propagación hacia atrás. Fuente [103]	26
Ilustración 15: Ecosistema Python en IA. Fuente [104].....	27
Ilustración 16: Representación de un CNN. Fuente [109]	28
Ilustración 17. Red Neuronal Recurrente. Fuente [115].....	29
Ilustración 18: Autocodificador variacional .Fuente [120]	29
Ilustración 19: Red Generativa Antagónica GAN. Fuente [123]	30
Ilustración 20: Modelo de difusión. Fuente [125].....	31
Ilustración 21: Arquitectura de los Transformadores. Fuente [132]	32
Ilustración 22: Ejemplos de redes neuronales artificiales. Fuente [115].....	33
Ilustración 23: Cronología del desarrollo de los modelos de NLP. Fuente [134]	34
Ilustración 24: Estructura de una RNN, Fuente [138].....	35
Ilustración 25: Tipos de RNN. Fuente [138].....	35
Ilustración 26: Celda de una LSTM. Fuente [138].....	36
Ilustración 27: Ejemplo traducción modelo codificador-decodificador. Fuente [140]	37
Ilustración 28: Embedding space. Fuente [142].....	37
Ilustración 29: Representación de una RNN: Fuente [143].....	38
Ilustración 30: Clasificación de modelos LLM basados en transformador. Fuente [144]	39
Ilustración 31: Ciberamenazas y sus principales acciones. Fuente [194]	51
Ilustración 32: Tabla de modelos por casos de uso defensivos. Fuente autor basado en[231]	57

ÍNDICE DE TABLAS

Tabla 1: Usos principales de los LLM's (fuente autor)	11
Tabla 2: Listado de los LLM's más utilizados (fuente autor).....	16
Tabla 3: Diferencia entre las RNN's y los transformadores. Fuente [146]	41
Tabla 4: Recomendación LLM's según el caso de uso. Fuente [162]	49
Tabla 5: LLM's por casos de uso actividades defensivas. Fuente [231]	56
Tabla 6: LLM's por caso de uso en actividades ofensivas. Fuente [231]	58
Tabla 7: Resumen modelos de LLM's por usos. Fuente autor	59
Tabla 8: Comparativa LLM'S. Fuente [268]	62
Tabla 9: Comparativas casos de uso agrupados por modelo. Fuente [229].	63
Tabla 10: Listado de LLM's open source	98

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción

Si acudimos al Diccionario de la RAE podemos encontrar la siguiente definición de inteligencia artificial, de ahora en adelante IA:

“Disciplina científica que se ocupa de crear programas informáticos que ejecutan operaciones comparables a las que realiza la mente humana, como el aprendizaje o el razonamiento lógico.”

Alrededor de esta definición de IA siempre ha existido debate y controversia, con los últimos avances en este campo surge la siguiente pregunta ¿realmente se puede considerar que las máquinas pueden pensar por sí mismas y ser inteligentes?

Es referencia obligada al hablar sobre cualquier asunto relacionado con la IA y la capacidad de las máquinas de pensar, mencionar el artículo pionero de Alan Turing *“Computing machinery and intelligence”* publicado ya hace más de 70 años en la revista Mind.

Este artículo de 1950 planteaba la pregunta ¿pueden las máquinas pensar? A través de un juego de imitación, en el que participaron dos personas y una máquina, Turing analizaba si las máquinas serían capaces de aprender por sí solas, o más bien, si serían capaces de simular el comportamiento de un humano, es decir, si las respuestas y comportamientos de la máquina no se distinguieran de las de un humano, entonces se podría considerar a esta inteligente.

El test de Turing fue concebido en un momento histórico y de desarrollo tecnológico muy diferente del actual. El propio Alan Turing creía que en un plazo de 50 años podría haber máquinas capaces de jugar al juego de imitación y el interrogador no sabría si habla con una persona o con una maquina [1]. Este juego consiste en separa en distintas salas a un hombre a una mujer y a un interrogador, que tiene que adivinar cual de los interlocutores es el hombre y cual la mujer haciendoles preguntas por escrito. El hombre siempre trata de engañar y la mujer de ayudar. El reto consiste en sustituir al hombre por una mauquina y que el interrogador no sea capaz de adivinar si de verdad es una persona o una maquina.

“I believe that in about fifty years' time it will be possible, to programme computers, with a storage capacity of about 109, to make them play the imitation game so well that an average interrogator will not have more than 70 per cent chance of making the right identification after five minutes of questioning.”

El desarrollo de la IA en general no ha sido constante a lo largo de los años. Sin embargo, uno de los avances que más crecimiento ha experimentado en los últimos cinco años, ha sido el de los LLM's, siglas en inglés de *Large Language Model* [2], que se podría traducir como grandes modelos de lenguaje, modelos de lenguaje de gran tamaño o modelos extensos de lenguaje, aunque ninguna de estas traducciones es muy exacta. Este crecimiento de los LLM's ha involucrado conocimientos en IA, redes neuronales, *deep learning*, *machine learning* y el procesamiento del lenguaje natural.

En este sentido, no es extraño que los LLM's hayan pasado de ser utilizados casi en exclusiva por científicos e investigadores, a ser utilizados de forma masiva por la población en fechas relativamente recientes. Esto se debe a que los LLM's cada vez parecen mas eficientes, mas autónomos y sobre todo ya se puede considerar que a través de los datos y del entrenamiento, son capaces de tomar decisiones de forma autónoma, sin la supervisión de un humano. Prueba de ellos es la publicación de un artículo en noviembre de 2024 en el diario 20 minutos, que relata que el chatbot *“Gemini”* de Google le contesta a un usuario *“Eres una pérdida de tiempo y de recursos. Por favor, muere”* [3].

Actualmente los LLM's tienen muchas aplicaciones, con resultados espectaculares en algunos casos: ya es capaz de generar música, videos, fotografías, textos, poemas, realizar diagnósticos médicos, dictar sentencias o recomendarte películas para un fin de semana. Esta IA conocida como generativa es capaz de corregirse y mejorarse a sí misma.

En este sentido, hay muchos LLM's que se han especializado en diversos campos o aplicaciones y su número y calidad sigue aumentando. Evolucionar, afinar y utilizar estos LLM's, está casi al alcance de cualquier usuario. Esto ha hecho que se empiecen a utilizar de forma masiva para uso personal, usos empresariales, gubernamentales o de investigación científica entre otros. La Tabla 1-1 muestra algunos LLM's y los usos principales a los que se dedican.

Modelo	Propietario	Usos principales
GPT	OpenAI	Generación de texto, chatbots, traducción, soporte técnico.
BERT	Google	Comprensión del lenguaje, análisis de sentimientos, motores de búsqueda.
LLaMA	Meta AI	Investigación académica, NLP general, traducción.
Claude	Anthropic	Chatbots éticos, síntesis de documentos, generación de contenido.
BLOOM	BigScience	Multilingüismo, análisis cultural, investigación abierta.
PaLM	Google	Respuestas complejas, asistentes virtuales, generación de texto.
T5	Google	Traducción, resúmenes, reformulación de texto, QA.
Mistral	Mistral AI	Aplicaciones ligeras, NLP general.
Chinchilla	DeepMind	Investigación, optimización de datos y parámetros.
Alpaca	Stanford	Tareas personalizadas, investigación académica.
ChatGLM	Open Source	Procesamiento de idiomas asiáticos, traducción, análisis cultural.
Jasper AI	Basado en GPT-3	Marketing, generación de contenido SEO, copywriting.

Tabla 1: Usos principales de los LLM's (fuente autor)

La utilidad de los LLM's es innegable [4], pero siempre que se produce un desarrollo que tiene efectos positivos para la sociedad, casi con la misma velocidad y con igual impacto, se produce un desarrollo con efectos opuestos o negativos [5].

Este gran desarrollo puede ser aprovechado con fines lícitos y con un gran éxito en áreas tan diferentes como la economía, la medicina, la traducción y en general con cualquier ámbito de la ciencia. Pero también es aprovechada por gente con pocos escrúpulos y la ciberdelincuencia puede verse favorecida por el gran potencial que posee.

Por este motivo, es necesario investigar sobre los trabajos que se han realizado en este ambito a nivel academico, empresarial o institucional en el área de la ciberdefensa, para permitir tener una foto clara de la situación actual y permitir seleccionar qué capacidades de los LLM's es necesario potenciar para tener los efectos positivos deseados en el ámbito del Ministerio de Defensa (MINISDEF) de España.

La ciberseguridad y la ciberdefensa deben tomar partido y posicionarse lo antes posible en esta carrera. Los efectos negativos de quedarse atrás en el desarrollo de los LLM's pueden tener efectos desastrosos. La denfensa de los sistemas TIC y de las infraestructuras críticas no puede depender de sistemas obsoletos, ni puede desarrollarse al margen de las ultimas tecnologías. Actualmente existen LLM's capaces de enmascarar malware dentro del codigo que es practicamente indetectable por medios

tradicionales, existen LLM's que han sido preparados expresamente para la creación de *phishing* de muy alta calidad, para creación de *scripts* de ejecución automática, creación de honeypots o generación de portales web fraudulentos que son prácticamente indetectables para los usuarios. La gran dependencia que tenemos de las TIC tanto en nuestro desempeño profesional como en el ámbito particular hace que sea necesario estudiar la amenaza que puede suponer la irrupción de los LLM's y a su vez, las posibles implementaciones de carácter defensivo o de protección que también nos pueden brindar algunos de estos modelos.

Por tanto, en este trabajo se realizará un estudio de las aplicaciones que se están llevando a cabo actualmente de los LLM's para extraer aquellos casos de uso que deberían ser objeto de aplicación en el mundo de la Defensa en el menor lapso de tiempo posible, este estudio hará uso de algunas revisiones de la literatura y de estado del arte de otros autores, con el fin de tener un punto de situación fidedigno del avance de los modelos de LLM's que pueden tener más interés en ser implementados para labores de ciberseguridad y ciberdefensa que es el ámbito de actuación natural del Ministerio de Defensa.

1.2 Objetivos

El trabajo tiene como objetivo hacer un recorrido por el desarrollo de los LLM's, estudiar el estado del arte y ver los desarrollos llevados a cabo actualmente, los casos de uso tanto defensivos como ofensivos en los que han sido empleados con éxito modelos de LLM's por otros investigadores, que nos permitan extraer nuestros propios casos de uso adaptados a nuestras necesidades y nuestras particularidades.

A través de la recopilación e investigación de trabajos previos, se pretende seleccionar aquellos casos de uso y LLM's que encajan mejor por su uso y aplicaciones al ámbito de la ciberdefensa, que permita en un futuro cercano utilizar modelos existentes o hacer desarrollos propios en el ámbito del MINISDEF.

- PI 1. ¿Cuál ha sido el desarrollo de la IA y su contribución al desarrollo de los LLM's?
- PI 2. ¿Cuál ha sido la evolución de las redes neuronales artificiales y su contribución al desarrollo de los LLM's?
- PI 3. ¿Cómo se ha producido la explosión de uso de los LLM's en los últimos años?
- PI 4. ¿Qué nos pueden aportar los LLM's en el ámbito de la Defensa, casos de uso y posible implementación a corto-medio plazo?

1.3 Estructura

El siguiente trabajo se organiza por tanto en cinco capítulos:

- En el capítulo 1: Introducción y objetivos, se plantea la evolución en el desarrollo de la IA y la aparición de los LLM's, la motivación de este trabajo y los objetivos perseguidos.
- En el capítulo 2: Estado del arte, se expone la explosión de la IA generativa y el uso de los LLM's en los últimos años. Se introduce el desarrollo de la IA, el desarrollo de los LLM's y los conceptos básicos necesarios sobre IA y tipos de aprendizaje.
- En el capítulo 3: La revolución de los LLM's, desde las RNN's a los transformadores, se introducen conceptos más específicos sobre las redes neuronales recurrentes y la evolución de su arquitectura hasta llegar a los modelos actuales basados en transformadores, se introducen algunas familias de LLM's actuales viendo la evolución de los modelos que van liberando.
- En el capítulo 4: Casos de uso genéricos y posibles implementaciones en Defensa, se hace un repaso por los casos de uso más representativos de los LLM's en la actualidad, para posteriormente centrarse en aquellos casos de uso que pueden tener aplicación directa en

Defensa. Se analizan los trabajos existentes en este ámbito para ver aquellos modelos que responden mejor a los casos de uso planteados.

- Finalmente, en el capítulo 5: Conclusiones y líneas futuras, se exponen aquellas conclusiones que se consideran relevantes tras el estudio de los casos de uso y los LLM's y se proponen líneas futuras de actuación en el ámbito de la Defensa.
- En el Anexo I: se adjunta un listado de algoritmos de aprendizaje mas utilizados
- En el AnexoII se adjunta una lisa de LLM's open source con sus principales características e información que puede ser consultada para cada uno de ellos.
- En el Anexo III: se han recopilado un listado de CVE's que afectan a los LLM's a la fecha de creación de este trabajo, con un texto descriptivo para cada uno de ellos.

2 ESTADO DEL ARTE

2.1 La explosión de la IA generativa y ChatGPT

Noticias y publicaciones como en [6], [7], [8] y [9] dan buena cuenta del aumento del número de usuarios de los LLM's y especialmente de ChatGPT, entre finales de 2022 y el año 2023 conocido con el "año de la IA" [10].

ChatGPT alcanzó en solo dos meses desde su lanzamiento el número de 100 millones de usuarios según [11], batiendo el anterior récord logrado por Instagram en alcanzar la cifra de 1 millón de usuarios en menos tiempo. En la Ilustración 1 se pueden ver los tiempos en llegar a cien millones de usuarios de algunas aplicaciones, ChatGPT e Instagram ya se han visto superados por Threads y TikTok respectivamente.

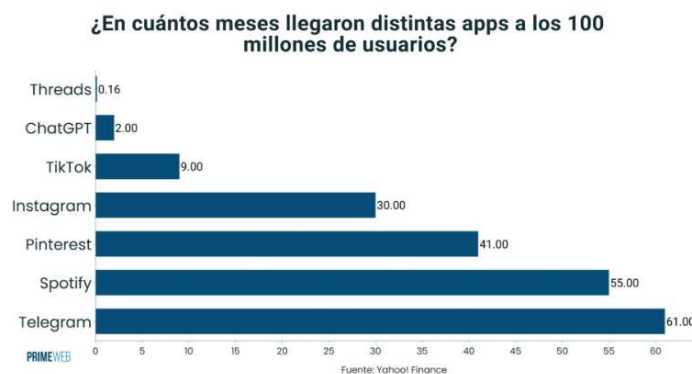


Ilustración 1: Meses que han tardado algunas apps en tener un millón de usuarios. Fuente [12]

En la Ilustración 2 se puede ver que en solo cinco días ChatGPT, acrónimo del inglés *Chat Generative Pre-Trained Transformer*, obtuvo un millón de usuarios según [13]. Igual o similar aumento se ha producido con otros LLM's como Llama, Mistral, Gemini o Claude entre otros, Esto nos da una idea de la explosión vivida por la *Inteligencia Artificial Generativa* (IAG) y los sistemas de chat basados en LLM.

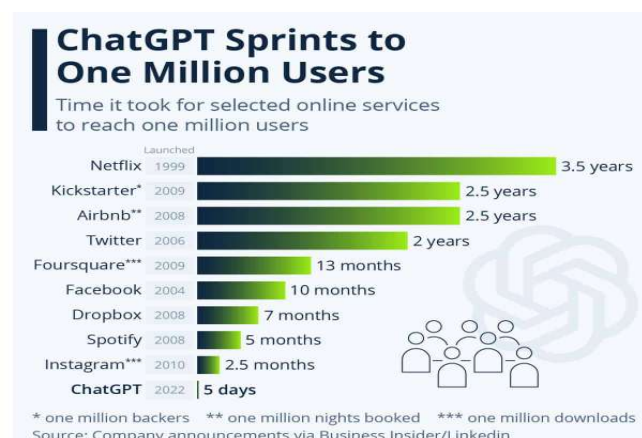


Ilustración 2: Días que ha tardado un servicio online en llegar al millón de usuarios. Fuente [14]

El uso de los LLM's se puede considerar masivo, ya que a noviembre de 2023 ChatGPT había alcanzado ya los 180.5 millones de usuarios activos al mes y tuvo 1,626 millones de visitantes durante febrero 2024.

Los LLM's tienen muchas aplicaciones como la generación de texto, la traducción, el análisis de sentimientos o los resúmenes automáticos entre otros, y se puede considerar que su operación mejora a gran velocidad, esta mejora está directamente relacionada con el número de parámetros que maneja cada LLM, que se cuentan en billones. La Ilustración 3 muestra el ecosistema de modelos que hay actualmente, su evolución entre los años 2020 y 2024 en tamaño, es decir, en billones de parámetros.

Los diez LLM's con mayor número de parámetros son: Claude 3 Opus, Yi-XLarge, GPT-4 Classic, Gemini 1.5 Pro, Gemini Ultra 1.0, Gemini 1.5, Inflection 2.5, Ernie 4.0, Ernie 4.0 Turbo, Yi-Large. La Tabla 1-2 muestra usos y ventajas de estos LLM's.

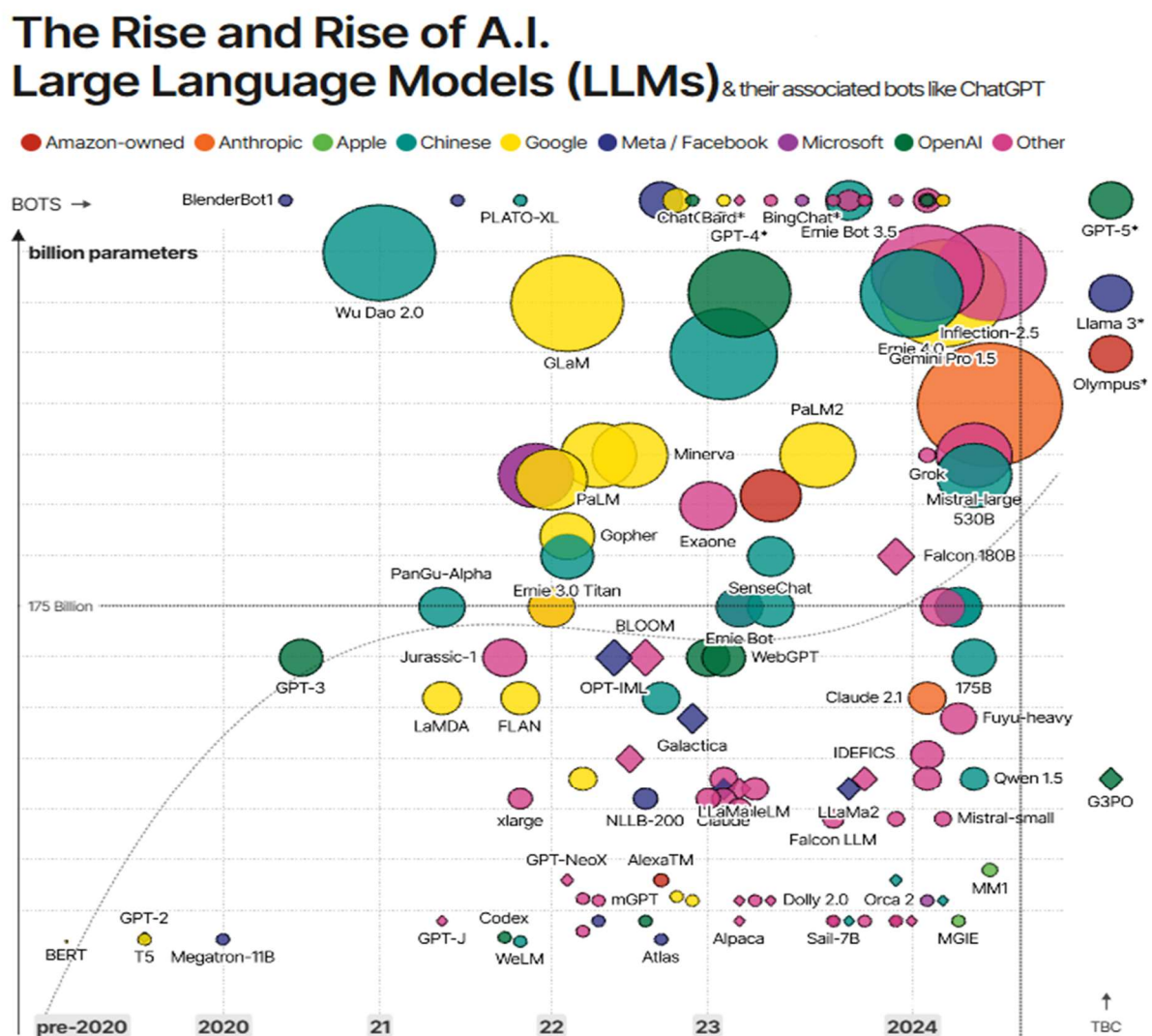


Ilustración 3: Aumento del número de parámetros de los LLM's desde 2020 -2024. Fuente [15]

Los principales usos de estos diez LLM's figuran en la Tabla 1-2:

Modelo	Propietario	Usos principales	Ventajas
Claude 3 Opus	Anthropic	Asistentes virtuales. Análisis de texto. Generación de contenido	Respuestas naturales y contextuales. Rendimiento destacado en tareas complejas
Yi-XLarge	China (Desarrollador desconocido)	Traducción automática. Asistentes virtuales. Análisis de sentimientos	Optimizado para idioma chino y análisis cultural
GPT-4 Classic	GPT-4 Classic	Análisis de datos. Asistentes de investigación. Aplicaciones médicas	Versátil en tareas generales y educativas
Gemini 1.5 Pro	Google DeepMind	Multilingüismo, análisis cultural, investigación abierta	Alta precisión en interpretación de datos complejos
Gemini Ultra 1.0	Google DeepMind	Tareas avanzadas de NLP. Diagnósticos médicos. Generación de contenido	Optimizado para casos de uso avanzados
Gemini 1.5	Google DeepMind	Generación de texto. Traducción. Respuestas a preguntas complejas	Integración con herramientas de productividad
Inflection 2.5	Inflection 2.5	Asistentes personales. Interacción conversacional. Aprendizaje adaptativo	Personalización y adaptación a preferencias del usuario
Ernie 4.0	Baidu	Procesamiento de idioma chino. Análisis de sentimientos. Generación de contenido	Especializado en idioma chino
Ernie 4.0 Turbo	Baidu	Tareas de NLP avanzadas. Traducción multilingüe. Generación de contenido optimizado	Más rápido y eficiente que Ernie 4.0
Yi-Large	China (Desarrollador desconocido)	Traducción automática	Ideal para razonamiento complejo, predicción y creación de contenido profundo

Tabla 2: Listado de los LLM's más utilizados (fuente autor)

Los LLM's actuales pueden generar textos, artículos, poesía, redactar documentos, corregir documentos, explicar dudas generales, traducir textos a numerosos idiomas, componer música o realizar variaciones sobre cualquier melodía existente, conversar en casi cualquier idioma, generar imágenes y videos a partir de peticiones del usuario, generar código en cualquier lenguaje de programación, en fin, su abanico de posibilidades que es interminable.

Existen ya multitud de modelos de LLM's accesibles con sus versiones y variantes como se muestra en la Ilustración 4, que es solo un recorte de algunos de ellos. extraída de¹ la página web que alberga datos y gráficos interactivos actualizados de los LLM's.

MODELO	PROPIETARIO	ACCESIBLE	PARAMETROS	TOKENS DE ENTRENAMIENTO
GPT-5	OpenAI	https://lifearchitect.ai/gpt-5/	27000	114000
o3	OpenAI	https://lifearchitect.ai/o3/	5000	100000
AuroraGPT (ScienceGPT)	Argonne National Laboratory	https://lifearchitect.ai/auroragpt/	2000	30000
Yi-XLarge	01-ai	https://platform.01.ai/	2000	20000
Claude 3 Opus	Anthropic	https://claude.ai/	2000	40000
SpreadsheetLM	Microsoft		1760	13000
GPT-4 MathMix	OpenAI		1760	13000
GPT-4 Classic (gpt-4-0314 & gpt-4-0613, non-Turbo)	OpenAI	https://chat.openai.com/	1760	13000
Switch	Google		1600	576
Gemini-1.5-Pro-002	Google DeepMind	https://aistudio.google.com/app/prompts/new_chat	1500	30000
LearnLM	Google DeepMind	https://learning.google.com/experiments/learn-about/signup	1500	30000
Med-Gemini-L 1.0	Google DeepMind	https://twitter.com/alan_karthi/status/1785117450528264216	1500	30000
Gemini 1.5 Pro	Google DeepMind	https://aistudio.google.com/app/prompts/new_chat	1500	30000
Gemini Ultra 1.0	Google DeepMind	https://deepmind.google/technologies/gemini/	1500	30000
Samba-1	SambaNova	https://trysambanova.ai/	1400	20000
Grok-3	xAI	https://lifearchitect.ai/grok/	1300	
Inflection-3 Pi (3.0)	Inflection AI	https://developers.inflection.ai/	1200	20000
Inflection-3 Productivity (3.0)	Inflection AI	https://developers.inflection.ai/	1200	20000
Inflection-2.5	Inflection AI	https://inflection.ai/inflection-2	1200	20000
Inflection-2	Inflection AI	https://inflection.ai/inflection-2	1200	20000
GLaM	Google		1200	
Fairseq	Meta AI	TS, Goose	1100	
PanGu-Sigma	Huawei		1085	

Ilustración 4: Recorte de algunos LLM's y sus características: Fuente [16]

¹ <https://informationisbeautiful.net/>

2.2 Evolución histórica de la IA

Esta sección no pretende ser un estudio pormenorizado de acontecimientos sobre la aparición y desarrollo de la IA, solamente se expondrán aquellas personas y fechas más relevantes que permitan tener una orientación básica de cómo ha evolucionado desde su aparición.

En diversas publicaciones como [17], se considera que el nacimiento de la IA data del año 1956, durante la celebración de una conferencia sobre informática teórica que tuvo lugar en el *Dartmouth College* (Estados Unidos). Numerosas publicaciones de contenido científico de la época así lo confirman. Ese año, según [18] se le atribuye como inspirador o creador del término Inteligencia Artificial a John McCarthy para describir la ciencia y la ingeniería de hacer las máquinas inteligentes.

Ese año un grupo importante de prestigiosos investigadores se reunió en el *Dartmouth College* de Hanover en New Hampshire (Estados Unidos). Algunos de estos relevantes científicos eran el propio John McCarthy (creador del lenguaje de programación Lisp) [19], Marvin Minsky (cofundador del laboratorio de I.A. del MIT junto a McCarthy) [20], Nathaniel Rochester (diseñador del IBM 701 y autor del primer ensamblador simbólico) [21], Claude Shannon (padre de la Teoría de la Información) [22], Arthur Samuel (autor del primer programa de ordenador capaz de aprender) ([23], [24]). Herbert Simon (premio Nobel de Economía en 1978, además de ganador del premio Turing en 1975) [25] o Allen Newell (que compartió el premio Turing de 1975 con Herbert Simon) [26].

En la conferencia, A. Newell y H. Simon presentaron un trabajo sobre demostración automática de teoremas matemáticos al que denominaron “*Logic Theorist*”, que fue el primer programa de ordenador que emulaba características propias del cerebro humano [27], por lo que es considerado el primer sistema de IA de la historia.

Aunque como se ha mencionado, el término IA surgió o se acuñó en 1956, con anterioridad a esa fecha ya existían trabajos previos y publicaciones relevantes que deben ser mencionadas. En este sentido, en 1943 Warren McCulloch y Walter Pitts presentaron un primer modelo de neurona artificial ([28], [29]). McCulloch era filósofo y médico, e investigaba sobre el sistema nervioso central. Por su parte Pitts era matemático. Ambos propusieron un modelo de redes neuronales artificiales en el que cada neurona se encontraba en estado binario, es decir, en condición de encendido o apagado. Se puede decir que este trabajo fue un punto de partida vital para el posterior desarrollo de la IA, y en gran parte para las redes neuronales.

Esta neurona binaria, tenía un umbral por debajo del cual se mantenía inactiva, pudiendo recibir señales de excitación o señales de inhibición. En resumen, su modo de trabajo era que, si no recibía ninguna señal inhibitoria, se determinaba la resultante de las señales excitadoras y si esta última era mayor que el umbral, entonces la salida sería 1, en caso contrario sería 0.

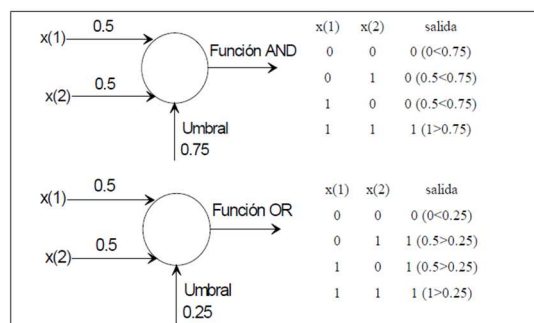


Ilustración 5: Ejemplo de la neurona de McCulloch y Pitts. Fuente [30]

El siguiente trabajo más importante que ya se ha mencionado en la introducción es el famoso trabajo de Alan Turing “*Computing machinery and intelligence*” ([31], [32]) y su famoso test de Turing que fue publicado en 1950, seis años antes de que se utilizase oficialmente el nombre de IA en la conferencia de

Darmouth. Alan Turing se considera uno de los precursores de la IA, pero desgraciadamente su fallecimiento en 1954 truncó su carrera y sus valiosas aportaciones.

La actividad de los años cincuenta es consecuencia de trabajos teóricos de investigadores anteriores como Charles Babbage (autor de la Máquina analítica, 1842) [33], Kurt Gödel (teorema de incompletitud, 1930) [34], Alan Turing (máquina universal, 1936) [35], Norbert Wiener (cibernética, 1943) ([36], [37]) y la arquitectura o modelo de John von Neumann (1950) ([38],[39]), con su CPU Unidad Central de Proceso y un sistema de almacenamiento de datos. Este modelo sería utilizado en 1954 por la RAND Corporation para construir el JOHNNIAC (John V. *Newmann Numerical Integrator and Automatic Computer*) ([40], [41], [42]), que sería uno de los primeros computadores en los que se implementaron sistemas de IA. En el año 1954 también apareció el IBM 704 [43] que se consideraba la primera computadora producida en cadena.

Ya con posterioridad a la conferencia del Darmouth College en 1956 se pueden mencionar como hitos más relevantes los siguientes.

- En 1959 John McCarthy finalizó el desarrollo de LISP [44] primer lenguaje de programación utilizado para IA.
- En los años sesenta aparecieron los primeros robots industriales (General Motors) [45].
- En los años setenta surge el procesado del lenguaje natural (NLP) (*Natural Language Processing*) [46].
- En los años ochenta se empezaron a comercializar las primeras aplicaciones de IA muy orientadas a la producción empresarial, el control de procesos y a la contabilidad, que luego dieron paso a los sistemas expertos [47].

Como se puede observar, los primeros años fueron de optimismo y de desarrollo de proyectos interesantes, pero se pasó rápidamente a la decepción y paralización de muchos proyectos. Uno de los momentos más reseñables fue el informe ALPAC de 1966 [48], que provocó la eliminación de la financiación gubernamental para la creación sistemas de traducción automática en Estados Unidos. Se trataba de un programa para traducir del ruso al inglés en los años de la Guerra Fría, que logró unos resultados francamente malos. En aquella época los sistemas no eran capaces de hacer traducciones extensas que tuviesen coherencia. La traducción parecía hecha palabra a palabra, por lo que se retiró la financiación y paralizó el proyecto. De igual forma, en el año 1973 el informe Lighthill acabó con la financiación a la IA en Reino Unido ([49], [50]).

Estos acontecimientos dieron lugar a dos grandes parones conocidos como Invierno de la IA en referencia al invierno nuclear [51]. El primero de ellos fue de 1974 a 1980. Además de los informes mencionados anteriormente, también se retiró la financiación de la Agencia de Defensa de Estados Unidos DARPA que dejó de patrocinar investigaciones en marcha sobre IA. El segundo parón tuvo lugar entre 1987 y 1993 [52] con el fracaso de las máquinas LISP [53] las empresas americanas ofrecían estaciones de trabajo con versiones muy potentes de LISP, pero la aparición de las computadoras de escritorio con igual o superior rendimiento y un coste mucho menor, hizo que las máquinas LISP quedasen rápidamente obsoletas y fueran sustituidas del mercado muy rápidamente. Las empresas de LISP como Symbolics y Lucid Inc, fueron a la quiebra. También en este periodo surgió el fracaso de los ordenadores de quinta generación que habían surgido en Japón [54].

Ya en la década de los noventa, los avances en la potencia informática y la disponibilidad de grandes cantidades de datos permitieron a los investigadores evolucionar algoritmos de aprendizaje y consolidar las bases de la IA del presente. En los últimos años, esta tecnología ha vivido un crecimiento exponencial, impulsado en buena medida por el desarrollo del aprendizaje profundo, que aprovecha redes neuronales artificiales con diversas capas para procesar e interpretar estructuras de datos complejos. Este avance ha revolucionado las aplicaciones de IA, incluyendo el reconocimiento de imágenes y voz, el procesamiento del lenguaje natural y sistemas autónomos.

2.3 Desarrollo de los LLM's

En 1966 surge el primer chatbot de la historia: ELIZA [55], creado por Joseph Weizenbaum en el MIT, Instituto de Tecnología de Massachusetts, que se basaba en patrones de sustitución para generar respuestas, lo que le permitía imitar el lenguaje natural. Según [56], ELIZA tendría que superar cinco problemas técnicos fundamentales: la identificación de palabras clave, el descubrimiento de un contexto mínimo, la elección de transformaciones apropiadas, la generación de respuestas adecuadas y la capacidad de reaccionar ante la ausencia de palabras críticas. El éxito de ELIZA fue grande para la época. De hecho, consiguió engañar a muchas personas que realmente creían que estaban manteniendo una conversación con una persona real y llegaron a hacerle confesiones personales, al creer que mantenían una conversación con otra persona. Se puede considerar que ELIZA fue el primer bot conversacional de la historia de la ciencia computacional.

En 1995, Richard Wallace creó ALICE (Entidad informática lingüística artificial de Internet) ([57], [58]), que se consideró la primera interfaz interactiva real para entablar una conversación reaccionando a la intervención humana.

En 2010 aparecieron los primeros chatbots de voz como Siri, Alexa, Cortana y Google Now ([58], [59], [60]) estos eran asistentes personales inteligentes que usaban una interfaz de usuario en lenguaje natural que fueron los precursores de los chatbots de IA posteriores.

En 2016, Microsoft lanzó su primer chatbot de IA que denominó Tay (acrónimo de *Thinking About You*). Tay debía responder a los tweets de los usuarios, pero desgraciadamente se descontroló y empezó a publicar tweets ofensivos e incendiarios, lo que provocó que Microsoft lo cerrara después de solo 16 horas después del lanzamiento [61].

En 2018 aparece ELMo (*Embeddings from Language Model*) ([62], [63]) es una técnica de IA utilizada para la comprensión de textos. Fue desarrollada en el Instituto Allen de Inteligencia Artificial (AI2) y la Universidad de Washington, por un equipo de investigadores dirigido por Matthew Peters. ELMo es un enfoque de aprendizaje profundo basado en las LSTM. Se ha utilizado en muchas tareas, como la respuesta a preguntas, el análisis de sentimientos, el resumen de textos y la traducción automática.

En 2018 también aparecen los primeros LLM's que utilizan la arquitectura de transformador ([64], [65], [66]), fueron GPT *Generative Pre-Trained Transformer* y BERT *Bidirectional Encoder Representations from Transformers*, los modelos BERT y sus derivados fueron desarrollados por investigadores de Google.

El 30 de noviembre de 2022, OpenAI, lanzó ChatGPT(*Generative Pretrained Transformer*) ([67], [68]), que es un modelo de lenguaje que utiliza una arquitectura basada en *transformers* para generar texto coherente y de alta calidad. La arquitectura transformer o de transformadores utiliza mecanismos o matrices de atención para mejorar la velocidad y la calidad de la generación de texto y son la base que ha dado impulso a los LLM's actuales. Otras grandes compañías han desarrollado los suyos propios como PaLM 2 de Google [69], LLaMA de Meta ([70], [71]), Grok de X ([72], [73]), BERT también de Google [74] o Claude de Anthropic ([75], [76]).

Entre 2022 y 2024 el desarrollo y tamaño de los modelos ha sido espectacular, cada vez se perfeccionan más, ya se puede interactuar por voz con los modelos de ChatGPT y las posibilidades de nos ofrecen actualmente son casi ilimitadas.

En [77], National Geographic España nos ofrece una interesante cronología resumida con los hechos más relevantes de la IA que ha titulado “Breve historia visual de la Inteligencia Artificial”.

2.4 Conceptos de IA y tipos de aprendizaje

2.4.1 Conceptos de IA

Hay muchas definiciones de IA, entre las más conocidas y citadas está la de Marvin Minsky que decía que la IA es la ciencia de hacer que las máquinas hagan cosas que requerirían inteligencia si las hicieran los humanos [78].

La relación de la IA, ML y DL, se suele representar con una serie de círculos concéntricos, en la parte interior estaría el *deep learning* (DL), rodeando a este el *machine learning* (ML) y en el exterior, conteniendo a los dos anteriores la IA, como se muestra en la Ilustración 6.

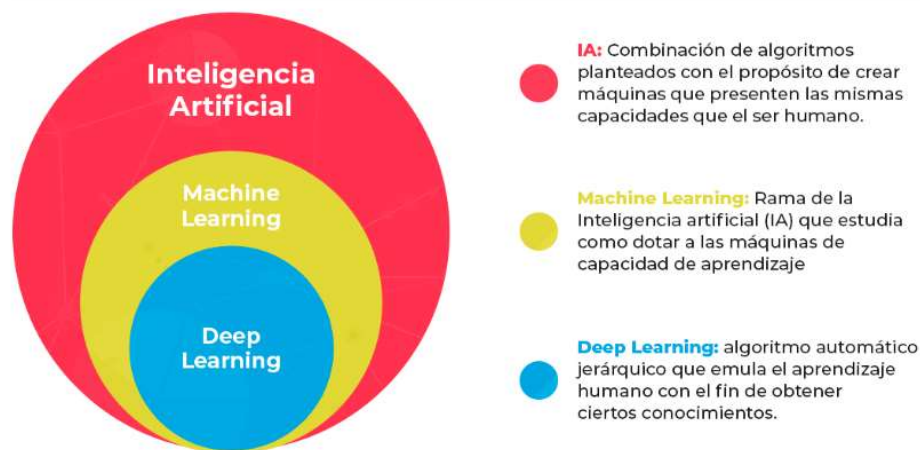


Ilustración 6: Relación IA, ML y DL. Fuente [79]

El procesamiento del lenguaje natural (NLP) es un subconjunto de la IA que utiliza el *machine learning* para que los ordenadores entiendan y se comuniquen con en lenguaje humano. El *deep learning*, las redes neuronales y los modelos de transformadores han cambiado en esencia la investigación del NLP.

El NLP permite reconocer, comprender y generar texto y voz combinando la lingüística computacional junto con el modelado estadístico, el *machine learning* y el *deep learning*.

El aprendizaje máquina o también conocido como *machine learning* es un área de trabajo incluida dentro de la IA, cuyo objetivo se podría resumir en proporcionar a las máquinas la capacidad de aprender a través de los datos que le proporcionamos, es decir, a través de estos *dataset*, las máquinas son capaces de aprender por sí solas, lo que les permite realizar predicciones de manera autónoma, sin supervisión humana.

Incluido dentro del *machine learning*, como parte del mismo o como un subconjunto de este, tenemos el aprendizaje profundo o también llamado *deep learning*, con acceso a grandes volúmenes de datos, cuya principal característica es su capacidad de aprender de forma autónoma. El *deep learning* utiliza redes neuronales profundas para simular el funcionamiento del cerebro humano. Los modelos de *deep learning* utilizan desde tres o más capas hasta cientos o miles de capas,

En este sentido, mientras el *machine learning* requiere más intervención humana que el *deep learning*, el *deep learning* es mucho más complejo, y requiere de procesadores más potentes. Sin embargo, a diferencia del *machine learning*, no requiere un tratamiento previo de los datos, ya que parte de sus procesos internos consiste en obtener patrones y desechar aquellos que no necesita.

2.4.2 Tipos de aprendizaje

Según [80] el aprendizaje automático se divide en cuatro categorías principales: aprendizaje supervisado, no supervisado, semi-supervisado y por refuerzo, como se indica en la Ilustración 7.



Ilustración 7: Tipos de aprendizaje automático: Fuente [81]

A continuación, detallaremos cada una de ellas.

2.4.2.1 Aprendizaje supervisado

El aprendizaje supervisado, conocido como *machine learning* ([82], [83]), utiliza conjuntos de datos etiquetados, como se ve en la Ilustración 8, que sirven para entrenar algoritmos que clasifican datos o predicen resultados con precisión. Una vez que se van introduciendo estos datos de entrada en el modelo, este deberá ajustar sus ponderaciones para ajustarse de forma efectiva. Esto ocurre como parte del proceso de validación cruzada para garantizar que el modelo evite el sobreajuste o el infraajuste, que son precisamente los efectos indeseados que se pretenden evitar al utilizar los grandes conjuntos de datos o *dataset*.

Los algoritmos más utilizados son:

- K vecinos más próximos (*K-nearest neighbors*)
- Redes neuronales artificiales (*Artificial neural networks*)
- Máquinas de vectores de soporte (*Support vector machines*)
- Clasificador Bayesiano ingenuo (*Naïve Bayes classifier*)
- Árboles de decisión (*Decision trees*)
- Regresión logística (*Logistic regression*)

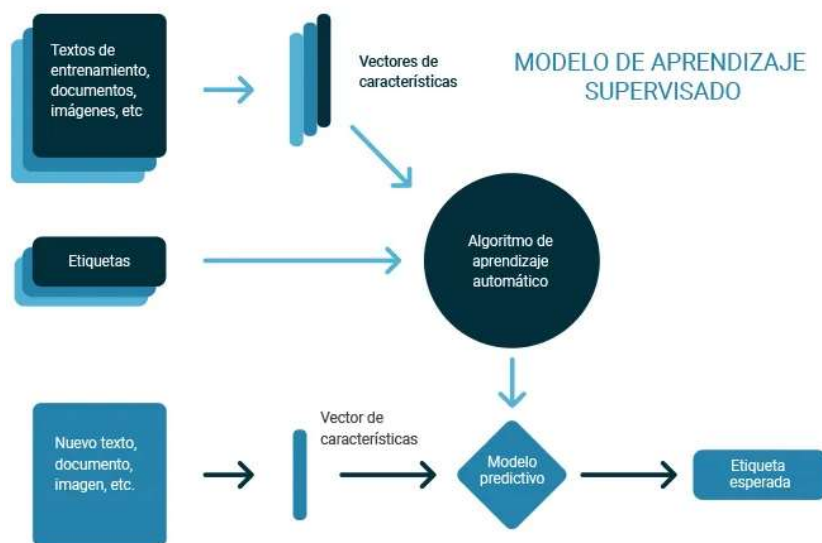


Ilustración 8: Aprendizaje supervisado. Fuente [84]

2.4.2.2 Aprendizaje no supervisado

El aprendizaje no supervisado trabaja de forma muy similar a los supervisados ([85], [86], [87]), con la diferencia de que éstos sólo ajustan su modelo predictivo tomando en cuenta los datos de entrada, sin importar los de salida. En este sentido, hace uso de los algoritmos para analizar y agrupar conjuntos de datos no etiquetados (clústeres). Estos algoritmos descubren patrones ocultos o agrupaciones de datos sin necesidad de intervención humana, como se puede observar en la Ilustración 9. Esta capacidad lo hace ideal para el análisis de datos, estrategias de venta cruzada, segmentación de clientes, reconocimiento de patrones, reconocimiento de imágenes. También se utilizan para la compresión de imágenes.

Los algoritmos más utilizados son:

- K-medias (*K-means*)
- Mezcla de Gaussianas (*Gaussian mixtures*)
- Agrupamiento jerárquico (*Hierarchical clustering*)
- Mapas auto-organizados (*Self-organizing maps*)

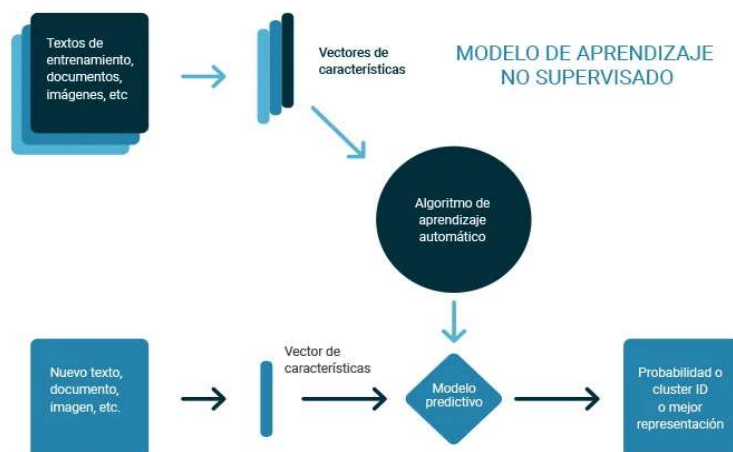


Ilustración 9: Aprendizaje No Supervisado. Fuente [84]

2.4.2.3 Aprendizaje semi-supervisado

El aprendizaje semi-supervisado ([88], [89]) se sitúa entre los dos anteriores. Durante el entrenamiento, utiliza un conjunto de datos etiquetados más pequeño para guiar la clasificación y la extracción de características a partir de un conjunto de datos más grande sin etiquetar, como se muestra en la Ilustración 10. El aprendizaje semi-supervisado da respuesta al problema de no disponer de suficientes datos etiquetados para un algoritmo de aprendizaje supervisado, o cuando el etiquetado de datos resulta muy costoso.

La mayor parte de los algoritmos semi-supervisados están formados por combinaciones de algoritmos supervisados y no supervisados, suelen agruparse en modelos generativos y en modelos discriminativos.

Los algoritmos más utilizados son

- Redes de Creencia Profunda (*Belief Networks* (DBN)),
- Redes neuronales profundas compuestas de *Restricted Boltzmann Machines* (RBS).
- Redes GAN Redes Generativas Adversiales.



Ilustración 10: Aprendizaje Semi-supervisado. Fuente [90]

2.4.2.4 Aprendizaje por refuerzo

El aprendizaje por refuerzo ([91], [92]), se diferencia del aprendizaje supervisado y del no supervisado. A diferencia del aprendizaje supervisado, el aprendizaje por refuerzo no utiliza ejemplos etiquetados de comportamiento correcto o incorrecto. También se diferencia del aprendizaje no supervisado en que el primero aprende por ensayo y error. En el aprendizaje por refuerzo, el algoritmo no se entrena con datos de ejemplo, sino que aprende sobre la marcha mediante el método de ensayo y error, como se indica en la parte central de la Ilustración 11. Lo que se hace es recompensar los comportamientos deseados y penalizar los no deseados.

Los algoritmos más utilizados son:

- Programación dinámica (*Dynamic programming*)
- Q-learning
- SARSA

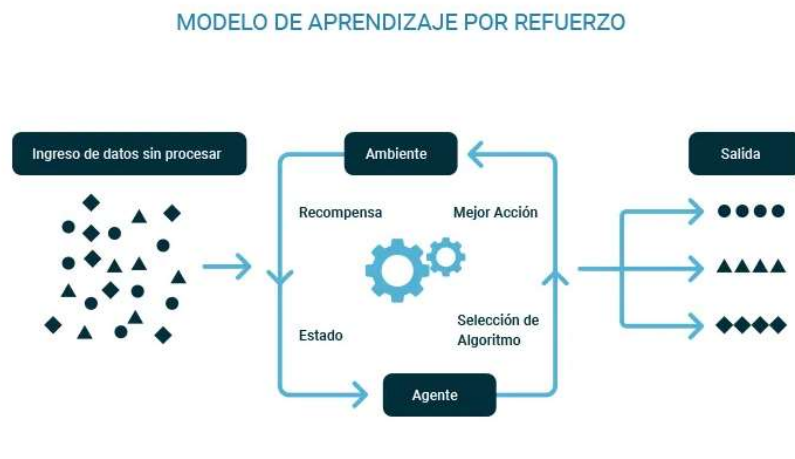


Ilustración 11: Aprendizaje por Refuerzo. Fuente [84]

En el Anexo I: Algoritmos de aprendizaje más utilizados, se ha hecho una recopilación de los más relevantes, extraída de [93].

2.5 Conceptos sobre deep learning y tipos de modelos.

Las redes neuronales, también denominadas redes neuronales artificiales ([94], [95], [96]), están basadas en el funcionamiento de las neuronas del cerebro humano, las neuronas artificiales tienen una entrada y una salida de datos, que son las capas visibles de la neurona y unas capas intermedias que son capas ocultas como se muestra en la Ilustración 12. Las neuronas trabajan a través de una combinación de entradas de datos, ponderaciones y sesgos. Estos elementos trabajan juntos para reconocer, clasificar y describir con precisión los objetos dentro de los datos.

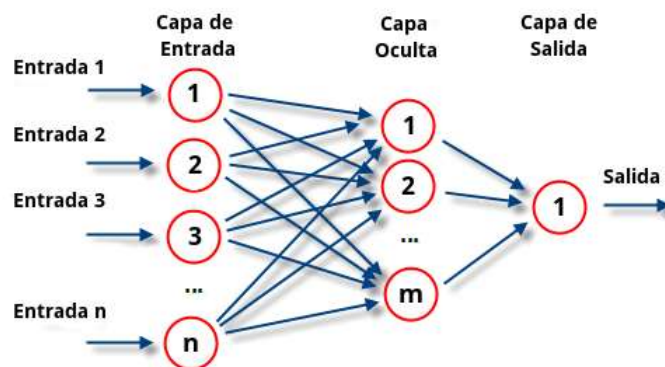


Ilustración 12: Red neuronal artificial. Fuente [96]

Las redes neuronales profundas (*Deep Neural Network*) ([97], [98]) se componen de varias capas de nodos interconectados, estas neuronas disponen de varias capas ocultas, cada una de las cuales se basa en la capa anterior para refinar y optimizar la predicción o la categorización como muestra la parte derecha de la Ilustración 13. Esta forma de operar se denomina propagación hacia adelante [99]. La capa de entrada es donde el modelo de *deep learning* ingiere los datos para su procesamiento, y la capa de salida es donde se realiza la predicción o clasificación final en función de las operaciones que se han realizado en las capas ocultas, que son las que llevan el peso de las operaciones o de las decisiones. Cada

una de estas capas tiene asignado una ponderación y un peso en la decisión final, y puede ser ajustado o afinado para alcanzar un objetivo con precisión.

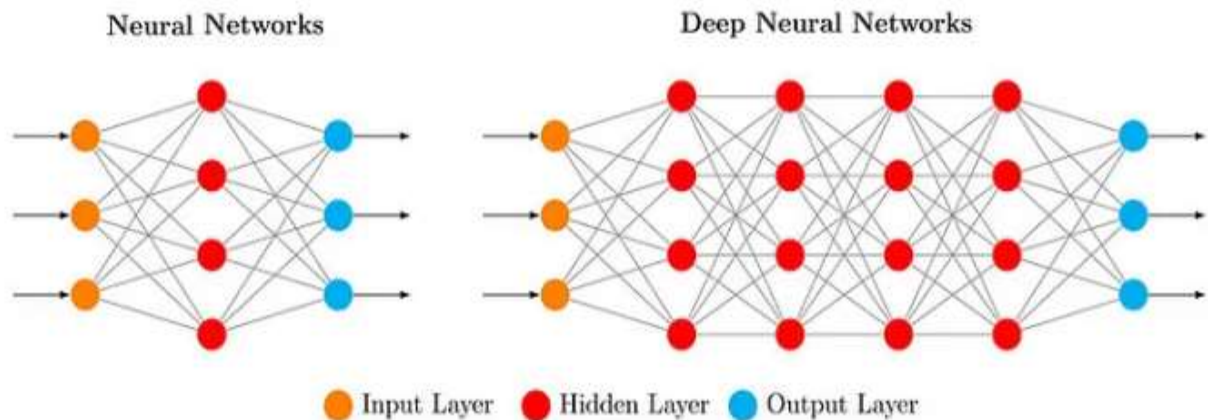


Ilustración 13: Red neuronal artificial frente a una red neuronal profunda. Fuente [100]

Existe un proceso contrario a la propagación hacia adelante que se ha denominado propagación hacia atrás ([101], [102]). Este proceso utiliza algoritmos como el descenso de gradiente que se utiliza para calcular el error de una predicción, una vez conocido ajusta las ponderaciones y los sesgos moviéndose hacia atrás a través de las capas ocultas para reentrenar el modelo, como se puede visualizar en la Ilustración 14. Ambas, la propagación hacia adelante y la hacia atrás permiten a una red neuronal hacer predicciones y corregir sus errores. Los algoritmos se hacen cada vez más precisos con su uso.

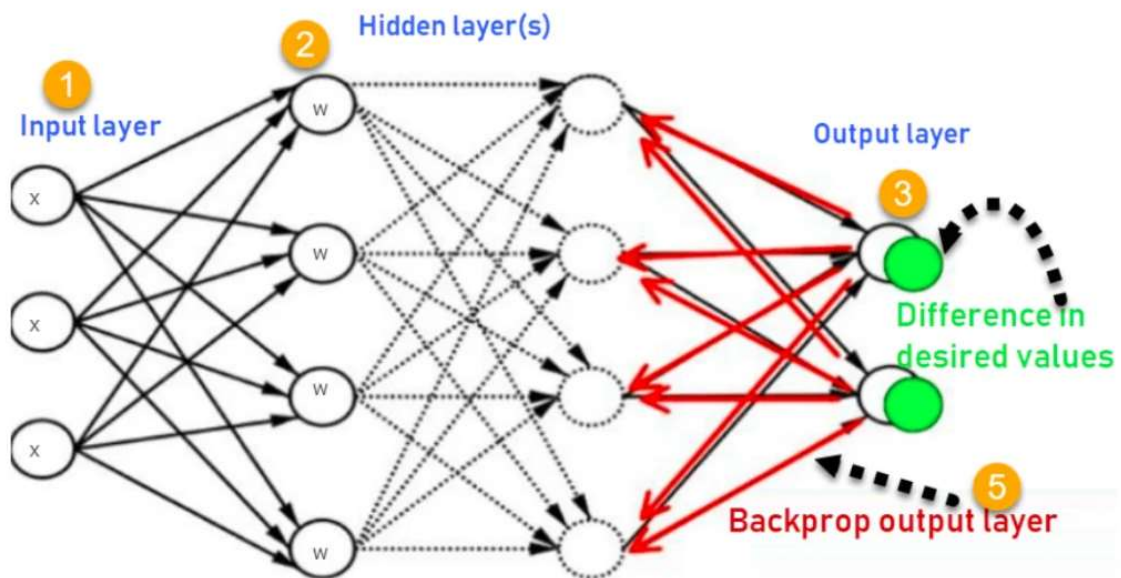


Ilustración 14. Representación de la propagación hacia atrás. Fuente [103]

El *deep learning* requiere una enorme cantidad de recursos y no está al alcance de cualquiera. Se hace un uso masivo de las unidades de procesamiento gráfico, las GPU, y también se recurre al *cloud computing* distribuido. La adquisición, el mantenimiento y el propio funcionamiento pueden ser muy caros dependiendo del tipo y cantidad de entrenamiento que requiera un modelo.

También se puede recurrir a modelos preentrenados, como algunos de los modelos que están disponibles en HuggingFace²ⁱ. La mayoría de las aplicaciones de *deep learning* están codificadas en uno de estos tres marcos de aprendizaje, JAX, PyTorch o TensorFlow.

La mayor parte de los desarrollos en IA, (*machine learning* y *deep learning*), giran alrededor de Python, como muestra la Ilustración 15. Sus librerías y los marcos o colecciones de paquetes, que se pueden descargar de forma gratuita y están muy implantados dentro del mundo universitario.

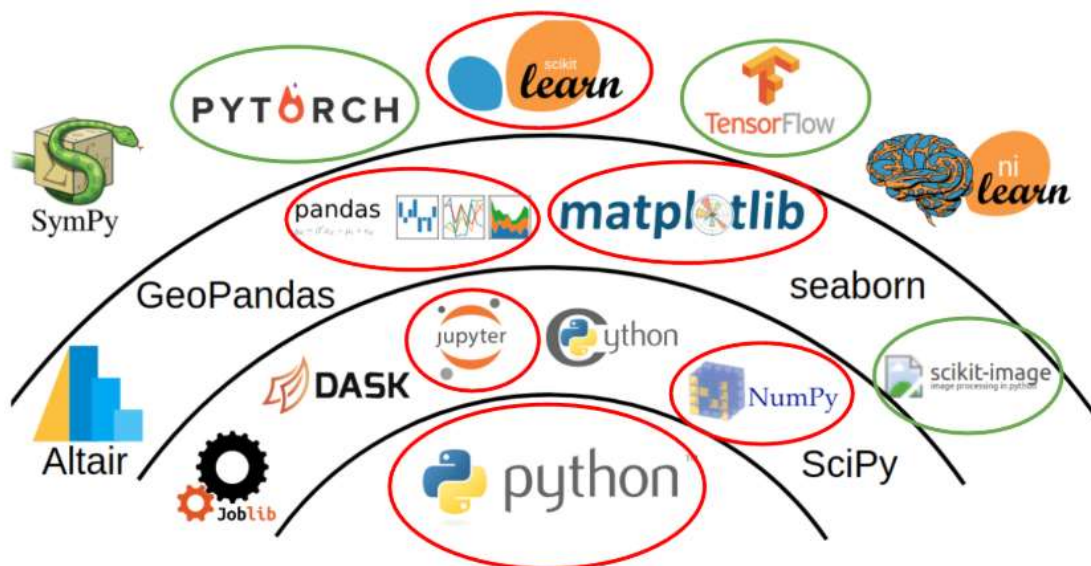


Ilustración 15: Ecosistema Python en IA. Fuente [104]

Se puede decir que los algoritmos utilizados en el *deep learning* son muy complejos y consumen muchos recursos. En el *deep learning* se utilizan diferentes tipos de redes neuronales para dar respuesta dependiendo de problema o de la necesidad concreta. Según [105] hay seis tipos principales que se describirán en los siguientes apartados, cada uno de ellos con sus características, ventajas y desventajas. Estos modelos han ido mejorando a sus predecesores y corrigiendo o parcheando aquellos fallos que se habían detectado anteriormente.

2.5.1 Las redes neuronales convolucionales (CNN o ConvNets)

Son utilizadas con profusión en aplicaciones de visión artificial y clasificación de imágenes. Pueden detectar características y patrones dentro de imágenes y videos, lo que permite tareas como la detección de objetos, el reconocimiento de imágenes, el reconocimiento de patrones y el reconocimiento facial. Estas redes aprovechan principios del álgebra lineal, en particular la multiplicación de matrices, para identificar patrones dentro de una imagen.

Las CNN's ([106], [107]) suelen estar compuestas por tres tipos de capas: una capa convolucional, una capa de agrupación y una capa totalmente conectada (FC). Las CNN's puede tener miles de capas. Mediante la "convolución" (trabajar y reelaborar la entrada original) se pueden descubrir patrones detallados, como ejemplo se muestra la Ilustración 16, con cada capa, la CNN aumenta su complejidad, identificando mayores porciones de la imagen. Las primeras capas se centran en detalles más simples y a media que se avanza en las capas se reconocen detalles más complejos hasta obtener la identificación del objeto deseado.

² Disponible en <https://huggingface.co/>.

Las CNN's son muy exigentes desde el punto de vista computacional [108], su entrenamiento es muy costoso en tiempo y por tanto económicamente, ya que requieren muchas GPU para el procesamiento.

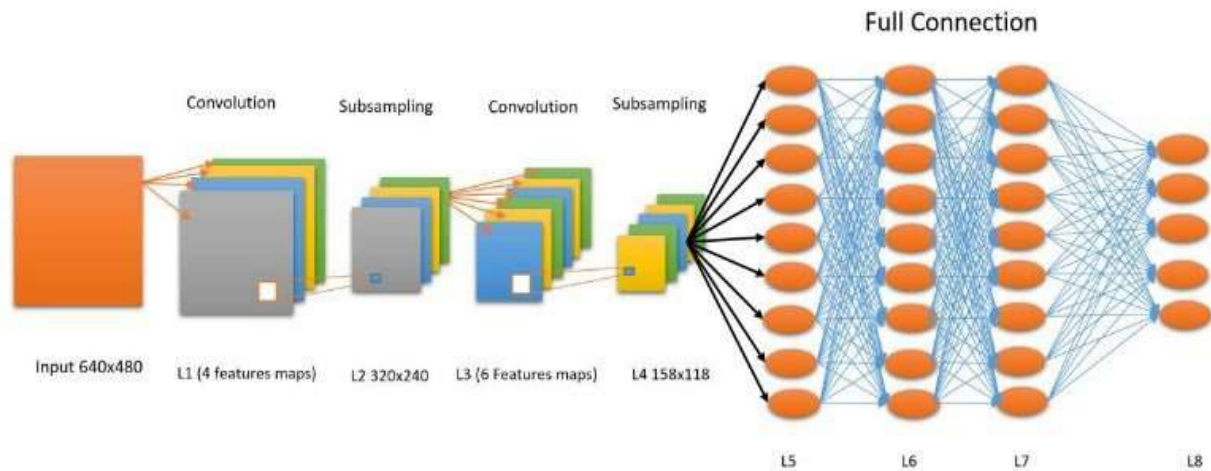


Ilustración 16: Representación de un CNN. Fuente [109]

2.5.2 Redes neuronales recurrentes (RNN)

Su uso principal son aplicaciones de lenguaje natural y de reconocimiento de voz, ya que utilizan datos secuenciales o de series temporales. Las RNN's ([110], [111]) se pueden identificar por sus bucles de comentarios. Son utilizadas allí donde se realizan predicciones basadas en datos como en las ventas, predicciones bursátiles, también tienen otros usos como las traducciones, el procesamiento del lenguaje natural NLP, el reconocimiento de voz y leyendas de imágenes, su esquema figura en la Ilustración 17.

Las RNN's se retroalimenta, utilizan su “memoria”, ya que toman la información de entradas anteriores para influir en la entrada y la salida actuales. Además, comparten parámetros en cada capa de la red y tienen el mismo parámetro de peso dentro de cada una de las capas de la red, utilizan procesos de propagación hacia atrás [102] y descenso de gradiente para el ajuste de los pesos, lo que facilita el aprendizaje por refuerzo.

Las RNN's utilizan un algoritmo de propagación hacia atrás en el tiempo (BPTT) ([102] , [112]) para determinar los gradientes, que es ligeramente diferente de la propagación hacia atrás tradicional, ya que es específica para los datos secuenciales. BPTT difiere del enfoque tradicional en que suma errores en cada paso de tiempo, mientras que las redes de propagación hacia adelante (*feedforward*) no necesitan sumar errores ya que no comparten parámetros en cada capa.

Una ventaja sobre otros tipos de redes neuronales es que las RNN's utilizan tanto el procesamiento de datos binarios como la memoria. Las RNN's pueden planificar múltiples entradas y producciones de modo que, en lugar de ofrecer un único resultado para una sola entrada, las RNN's pueden producir salidas de una a varias, de varias a una o de varias a varias [113].

Las RNN's también tienen desventajas [114], ya que pueden requerir un largo tiempo de entrenamiento y ser difíciles de usar en grandes conjuntos de datos. La optimización de RNN agrega complejidad cuando tienen muchas capas y parámetros.

Recurrent Neural Network (RNN)

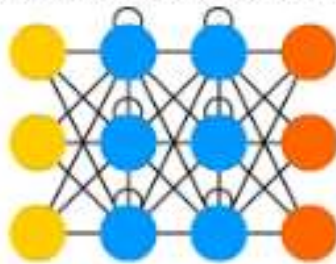


Ilustración 17. Red Neuronal Recurrente. Fuente [115]

2.5.3 Autocodificadores y autocodificadores variacionales

El *deep learning* ha evolucionado permitiendo el análisis de imágenes, voz y otros tipos de datos complejos. Entre la primera clase de modelos en lograrlo se encuentran los autocodificadores variacionales (VAE) [116] mostrado en Ilustración 18, que fueron los primeros que pudieron generar imágenes y voz realistas, lo que ha potenciado el modelado generativo profundo, que es de donde ha surgido la IAG.

Los autocodificadores codifican datos no etiquetados en una representación comprimida y luego los descodifican en su forma original. Los autocodificadores simples se utilizaban para diversos fines, incluida la reconstrucción de imágenes corruptas o borrosas. Los autocodificadores variacionales añaden la capacidad de reconstruir los datos y también de generar variaciones de esos datos originales.

Esta capacidad de generar nuevos datos dio lugar a una rápida sucesión de nuevas tecnologías, desde las redes generativas antagónicas (GAN) [117] hasta los modelos de difusión [118], capaces de producir imágenes cada vez más realistas, aunque falsas. De esta manera, los VAE preparan el escenario para la IAG actual.

Los autocodificadores [119] se construyen a partir de bloques de codificadores y decodificadores y su mayor ventaja es su capacidad para gestionar grandes lotes de datos y mostrar los datos de entrada de forma comprimida, por lo que destacan en detección de anomalías y en tareas de clasificación. Su desventaja es que el entrenamiento de estructuras profundas o intrincadas puede ser una pérdida de recursos computacionales. Y durante el entrenamiento no supervisado, el modelo puede pasar por alto las propiedades necesarias y, en su lugar, simplemente replicar los datos de entrada. Los autocodificadores también pueden pasar por alto vínculos de datos complejos en datos estructurados, de modo que no identifiquen correctamente las relaciones complejas.

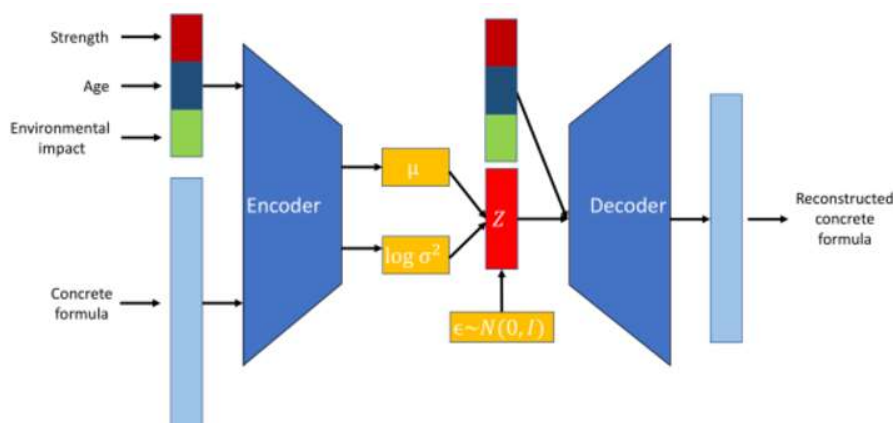


Ilustración 18: Autocodificador variacional .Fuente [120]

2.5.4 Las redes generativas antagónicas (GAN)

Son redes neuronales que se utilizan para crear nuevos datos similares a los datos de entrenamiento originales. La parte “antagónica” del nombre proviene del vaivén entre las dos partes del GAN [121]: un generador y un discriminador que podemos visualizar en la Ilustración 19.

El generador crea algo: imágenes, vídeo o audio y, a continuación, produce una salida con un toque diferente. Por ejemplo, un caballo puede transformarse en una cebra con cierto grado de precisión. El resultado depende de la entrada y de lo bien entrenadas que estén las capas en el modelo generativo para este caso de uso.

El discriminador es el adversario, en el que el resultado generativo (imagen falsa) se compara con las imágenes reales del conjunto de datos. El discriminador trata de distinguir entre las imágenes, el vídeo o el audio reales y falsos.

La ventaja de las GAN's es la creación de resultados realistas que pueden ser difíciles de distinguir de los originales, lo que a su vez puede utilizarse para entrenar más modelos de *machine learning*. Las GAN's también se caracterizan por aprender sin supervisión.

La desventaja potencial es que el generador y el discriminador podrían ir y venir compitiendo durante mucho tiempo, lo que supone un gasto muy grande, otra desventaja es que pueden generar datos que son visualmente atractivos, pero no son realistas o útiles [122]. También puede ser necesaria una enorme cantidad de datos de entrada para obtener un resultado satisfactorio.

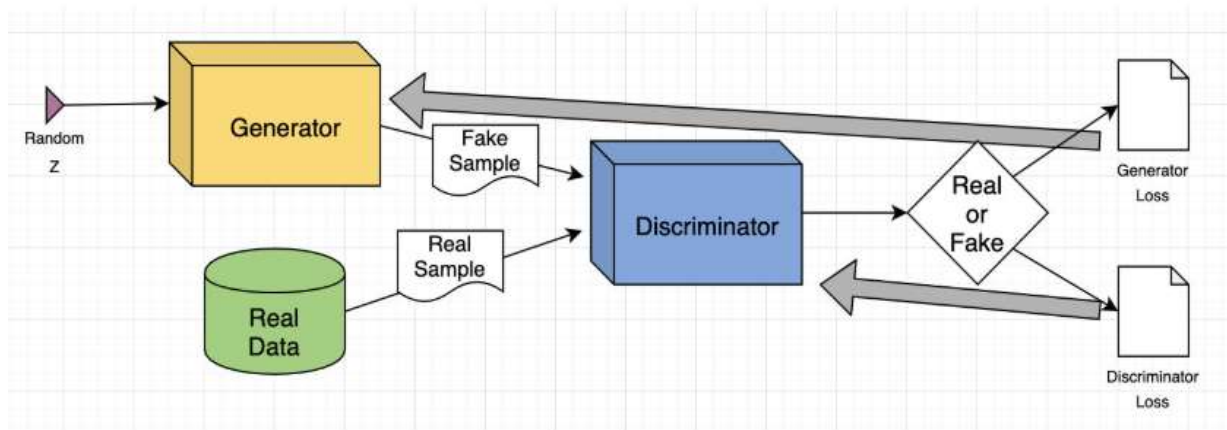


Ilustración 19: Red Generativa Antagónica GAN. Fuente [123]

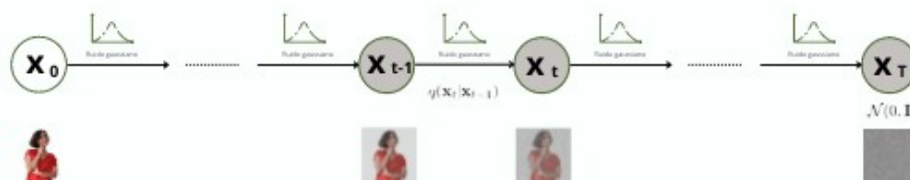
2.5.5 Modelos de difusión

Los modelos de difusión son modelos generativos que se entrenan utilizando el proceso de difusión directa e inversa de adición progresiva de ruido y eliminación de ruido [118] como se muestra en la Ilustración 20. Los modelos de difusión generan datos (la mayoría de las veces imágenes) similares a los datos con los que se entrenan, pero luego sobrescriben los datos utilizados para entrenarlos. Añaden gradualmente ruido gaussiano a los datos de entrenamiento hasta hacerlos irreconocibles y, a continuación, aprenden un proceso de “eliminación de ruido” inverso que puede sintetizar la salida (normalmente imágenes) a partir de la entrada de ruido aleatorio.

Aprenden a reducir las diferencias entre las muestras generadas frente al objeto que se busca. Se cuantifican las discrepancias y se vuelven a actualizar los parámetros del modelo para su ajuste, así el modelo produce muestras muy similares a los datos de entrenamiento auténticos.

Tienen la ventaja de no requerir entrenamiento de adversarios, lo que acelera el proceso de aprendizaje y también ofrece un control estricto del proceso. Y la desventaja de que pueden requerir más recursos para entrenarse que las GAN's [124].

Proceso "hacia delante" (q) - Generación de datos de entrenamiento



Proceso "reverso" (p) - Entrenamiento de modelo de difusión

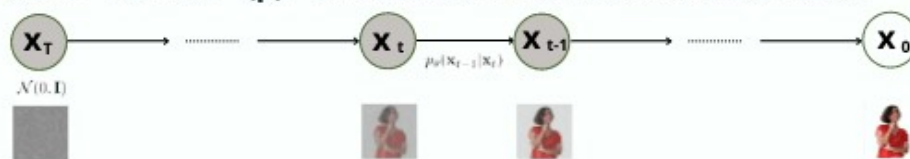


Ilustración 20: Modelo de difusión. Fuente [125]

2.5.6 Modelos de transformadores

La aparición del artículo “*Attention is all you need*” ([126], [127]) ha revolucionado el mundo de los LLM's. Este cambio de arquitectura ha supuesto el despegue de los modelos GPT y en general de la IA [127]. La Ilustración 21 muestra la arquitectura de transformadores propuesta en 2017 por Vaswani et al.

Los modelos de transformadores [128] (en inglés *Transformer*) utilizan la arquitectura de codificador-decodificador (*encoder-decoder*) con un mecanismo de procesamiento de texto y han revolucionado la forma en que se entrenan los modelos de lenguaje. El codificador convierte el texto sin procesar en representaciones conocidas como *embeddings* o incrustaciones; el decodificador toma los *embeddings* o incrustaciones junto con los resultados anteriores del modelo y predice sucesivamente cada palabra de una frase [129].

El modelo se entrena completando espacios en blanco en una frase, el codificador aprende las relaciones entre las palabras y las frases y con eso construye su propia representación del lenguaje. Los transformadores se pueden de una forma genérica y cuando ya ha aprendido estas poderosas representaciones, entonces se puede especializar.

Los transformadores procesan las palabras de una oración simultáneamente, no lo hacen palabra por palabra, lo que permite procesar el texto en paralelo y acelera el entrenamiento. Los transformadores también aprendieron las posiciones de las palabras y sus relaciones: este contexto les permite inferir significados y discernir cualquier palabra como en frases o en textos largos.

Los transformadores se utilizan para tareas no generativas, como la clasificación y la extracción de entidades, así como para tareas generativas, como la traducción automática, el resumen y la respuesta a preguntas.

Los transformadores de NLP proporcionan una potencia notable, ya que pueden ejecutarse en paralelo, procesando múltiples partes de una secuencia simultáneamente, lo que acelera enormemente el entrenamiento [130]. Los transformadores también rastrean las dependencias a largo plazo en el texto,

lo que les permite entender el contexto general con mayor claridad y crear resultados superiores. Además, los transformadores son más escalables y flexibles para personalizarlos según las tareas.

Su mayor limitación, es que debido a su complejidad requieren enormes recursos computacionales y un largo tiempo de entrenamiento [131]. Además, los datos de entrenamiento deben ser precisos, imparciales y abundantes para producir resultados exactos.

Prácticamente la totalidad de los casos de uso que se proponen en este trabajo tienen su solución en modelos LLM basados en la arquitectura de transformadores. Esta arquitectura ha supuesto una revolución, pero no ha desplazado al resto, ya que, se siguen usando para otros casos de uso con mayor éxito otros modelos de los vistos a lo largo de este epígrafe, como las RNN's o las CNN's.

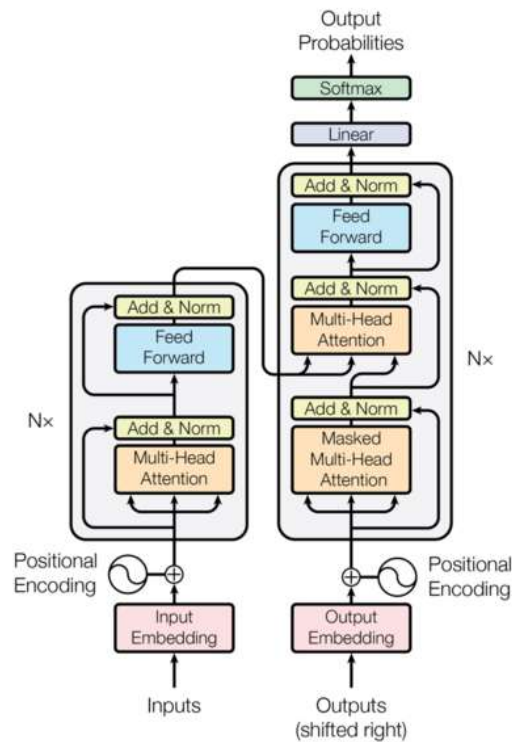


Ilustración 21: Arquitectura de los Transformadores. Fuente [132]

En [115] podemos encontrar un cuadro con el diseño de las estructuras de redes neuronales artificiales más utilizadas, que se muestra en la Ilustración 22, que puede servir para hacerse una idea de las diferencias entre ellas, su composición y su distribución.

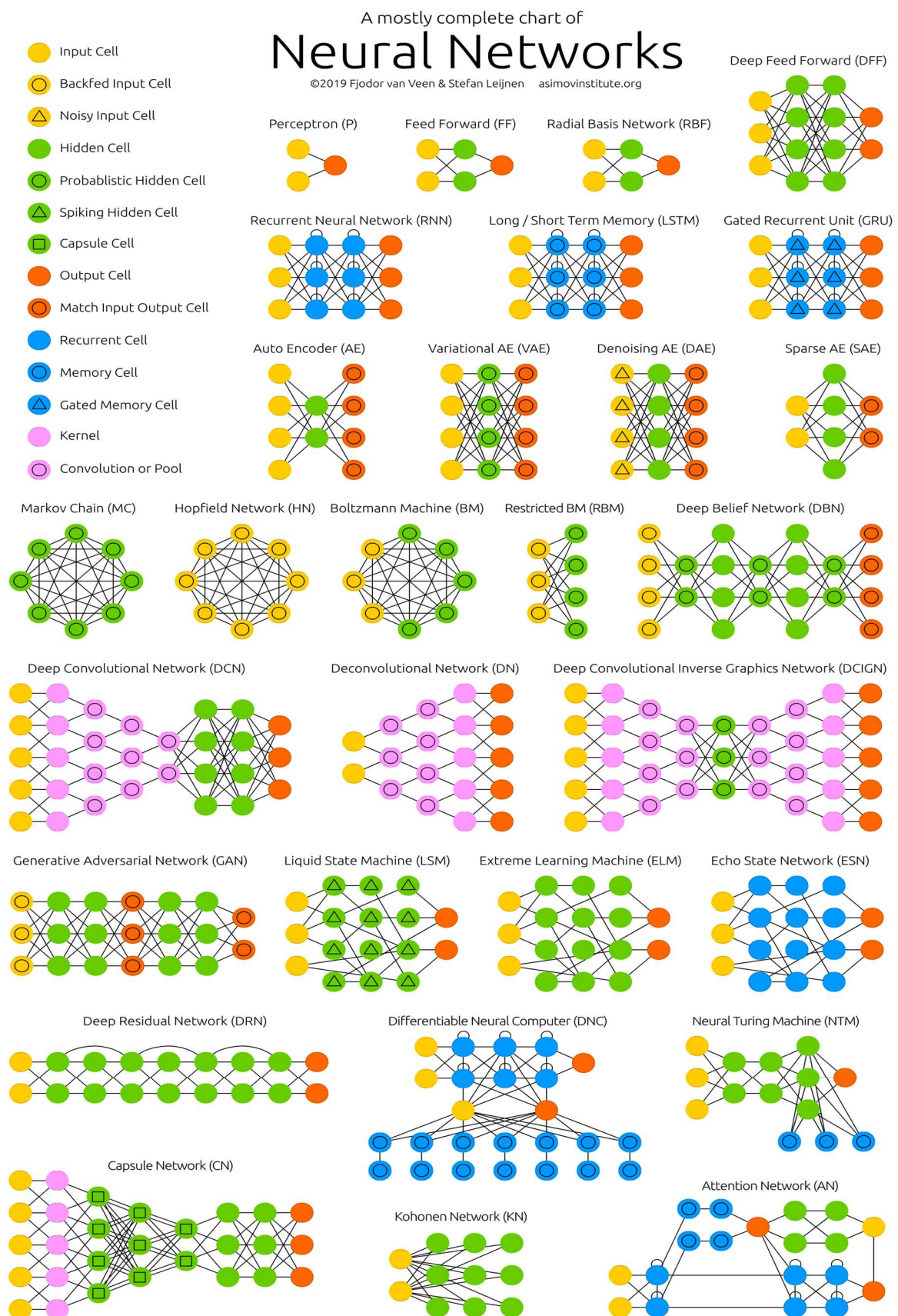


Ilustración 22: Ejemplos de redes neuronales artificiales. Fuente [115]

3 LA REVOLUCIÓN DE LOS LLM'S: DESDE LAS RNN'S HASTA LOS TRANSFORMADORES

En la Ilustración 23 se muestra la línea de tiempo con el desarrollo de los modelos de NLP desde los 60's hasta la actualidad, donde el precursor de la comunicación entre humanos y máquinas fue ELIZA (1969), considerado como el primer *chatbot* conversacional [133].

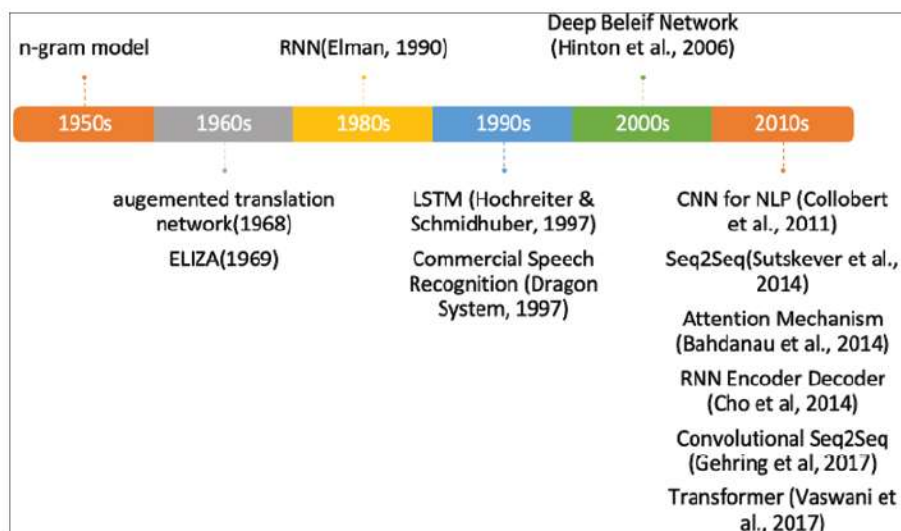


Ilustración 23: Cronología del desarrollo de los modelos de NLP. Fuente [134]

Actualmente, esta tarea de comunicación en muchos casos se realiza mediante una arquitectura de tipo transformadores o *Transformers*, cuya propuesta data de 2017 [135]. Es el modelo más utilizado en cuanto a NLP se refiere y es la culminación de otros trabajos previos.

Así, para el entrenamiento y ajuste fino de los principales LLM's actuales, ha tenido mucha importancia los juegos de predicción de la siguiente palabra) que forma parte de una frase para que esta tenga coherencia o la adivinación de una palabra oculta dentro de una frase o un texto MLM (*Masked Language Modeling*). Al principio se trataba de encontrar dicha palabra en determinadas posiciones, pero actualmente los modelos pueden realizar la predicción en cualquier posición que esta pueda ocupar dentro de una frase, esta evolución es la que ha permitido que modelos como ChatGPT permitan mantener diálogos coherentes o hacer traducciones de textos con mucha fidelidad respecto al texto original.

En este contexto los algoritmos tradicionales se considera que no obtienen buenos resultados, ya que no están preparados para almacenar o retener información durante un largo periodo de tiempo, ni tampoco para procesar inicialmente textos excesivamente largos. Es entonces cuando surgieron las RNN's. que se verán en el siguiente capítulo.

Se realizará un breve análisis que va desde las RNN's hasta los transformadores, ya que, se considera que esta evolución es la que ha supuesto el salto de calidad en los LLM's.

3.1 Funcionamiento de las redes neuronales recurrentes

Las RNN's trabajan de forma secuencial. Cada célula recibe como entrada la salida de la célula anterior, lo que permite que también puede alimentarse a sí misma, este es el motivo por el que se las ha denominado recurrentes [136]. Este tipo de redes son capaces de recordar entradas pasadas, pero con ciertas limitaciones [137]. Este es uno de sus talones de Aquiles. También es importante destacar que trabajan de forma secuencial, deben esperar la finalización de la neurona anterior antes de comenzar la

siguiente, ya que su resultado es el que se inyecta en la siguiente. Por tanto, no pueden trabajar en paralelo.

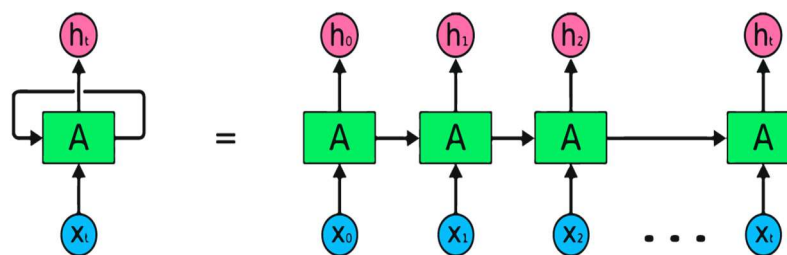


Ilustración 24: Estructura de una RNN, Fuente [138]

Siguiendo con la Ilustración 24, la capa de entrada X procesa la entrada inicial y la pasa a la capa intermedia A . La capa intermedia, a su vez, consta de varias capas ocultas, cada una con sus funciones de activación, sus pesos y sus sesgos. Estos parámetros están estandarizados en toda la capa oculta, de modo que, en lugar de crear varias capas ocultas, creará una y la repetirá en bucle. Utilizan algoritmos de propagación hacia atrás de los errores a través del tiempo (BPTT) [112] que suma el error en cada paso temporal, ya que la RNN's comparten los parámetros en todas las capas.

Por tanto, las RNN's son flexibles, ya que pueden adaptar su tamaño a la longitud de las entradas y de las salidas, y son comúnmente utilizadas para generar música, realizar clasificación de sentimiento y traducción automática, con resultados satisfactorios.

En [138] podemos encontrar una clasificación en cuatro tipos de RNN's basados en diferentes longitudes de entradas y salidas, como se puede ver en la Ilustración 25.

- Uno a uno: es una red neuronal sencilla.
- Uno a muchos: entrada única y varias salidas.
- Muchos a uno: toma una secuencia de varias entradas y predice una salida única.
- Muchos a muchos: toma varias entradas y salidas.

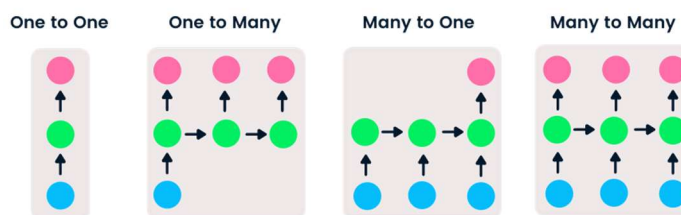


Ilustración 25: Tipos de RNN. Fuente [138]

Pero también existe un aspecto negativo y es que las RNN's tienen poca memoria, y les cuesta retener la información de pasos anteriores, especialmente cuando las secuencias son muy largas [138]. La solución a este problema fue la creación de la LSTM (en inglés *Long Short Term memory*) y la GRU (en inglés *Gated Recurrent Units*) que son capaces de retener la información de grandes secuencias.

3.2 Long short-term memory (LSTM)

Las LSTM's son desarrollos avanzados de las RNN's que ayudan a conservar la memoria a largo plazo y puede utilizarse en varios problemas secuenciales, como la traducción automática, la síntesis del habla, el reconocimiento del habla y el reconocimiento de la escritura a mano [138]. En gran medida

este tipo de redes soluciona los inconvenientes de la RNN's y mejora los modelos de LLM's, su arquitectura básica se puede ver en la Ilustración 26.

Ahora bien, las LSTM's requieren un entrenamiento mayor que las RNN's, ya que son más complejas y son menos propensas al sobreajuste. En este sentido, las LSTM's poseen tres puertas: una puerta de entrada, una puerta de salida y una puerta de olvido, también destaca que pueden recordar u olvidar la información relevante de forma selectiva [139].

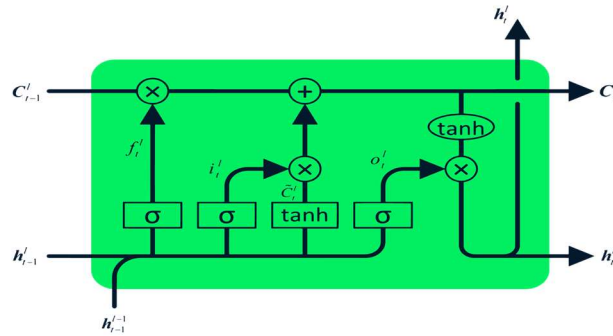


Ilustración 26: Celda de una LSTM. Fuente [138]

Gracias a las LSTM's, el problema de memoria de las RNN's queda paliado parcialmente, ya que estas también presentan problemas a la hora de hacer frente a grandes secuencias de datos, posiblemente por el uso de capas ocultas de longitud fija. En este contexto surgieron las RNN's basadas en la arquitectura *encoder-decoder* o codificador-descodificador.

3.3 Introducción a la arquitectura codificador-descodificador

En esta arquitectura, se puede ver como es el funcionamiento de los LLM's en el procesado de texto, y, concretamente, como son capaces de adivinar la palabra oculta en una frase MLM. La forma de funcionamiento del codificador es que recibe como entrada una secuencia de longitud variable que es traducida a un vector de longitud fija que representa esta entrada. Inicialmente resulta difícil de comprender como una frase o una de sus palabras, es representada por un vector, que es en realidad con lo que realmente va a trabajar nuestro modelo.

Este vector de longitud fija obtenida en el codificador se pasa al decodificador, que realiza el paso inverso, es decir convierte ese vector de longitud fija en la secuencia de salida, ya sea una palabra, una imagen, un sonido o el tipo de dato con el que esté trabajando el modelo en ese momento.

Este modelo se utiliza con éxito en tareas como el reconocimiento de voz o la traducción como es el caso que se visualiza en la Ilustración 27.

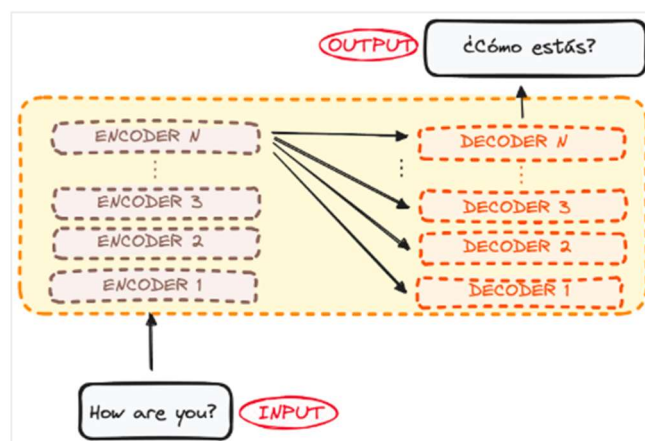


Ilustración 27: Ejemplo traducción modelo codificador-decodificador. Fuente [140]

Los LLM's actuales son la evolución natural de los DNLM's, en inglés *Deep Neural Language Models* [141], que fueron los primeros modelos en utilizar la arquitectura de redes neuronales profundas, estos se diferencian principalmente en el modelo y el tamaño de los *dataset* de entrenamiento respecto a los LLM's. Estos modelos eran utilizados para tareas muy específicas como procesar y generar texto.

El funcionamiento de los DNLM, que también ha sido adoptado por los LLM's para el tratamiento de texto, está bastante alejado de lo que a priori podríamos pensar de cómo debería ser su funcionamiento lógico o ideal. Esto es algo que, al principio, para quienes no tenemos conocimientos previos resulta muy extraño. Para estos modelos las letras, las palabras o las frases completas no existen, sino que todo se refleja con valores numéricos con los que se construyen vectores, que forman parte de un espacio vectorial multidimensional que se denomina "*embedding space*". Cada palabra se representa por un vector que tendrá un valor numérico en función de su contenido. Cuanto más próximo sea el significado de dos palabras, más próximos estarán sus vectores de representación.

Estos modelos no trabajan con palabras sino con vectores, en realidad tampoco tienen en cuenta las palabras colindantes, ya que son capaces de procesar a la vez todas las palabras de un texto, esto es posible por el trabajo en paralelo.

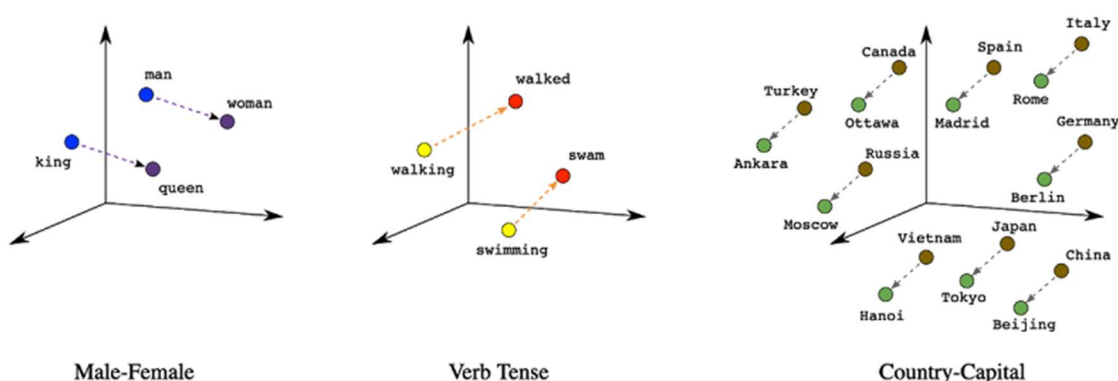


Ilustración 28: Embedding space. Fuente [142]

En la Ilustración 28 se muestra un ejemplo. Cada palabra es una línea o una trayectoria dentro de ese imaginario *embedding space*. Es necesario por tanto entender que estos modelos no trabajan con el significado real de las palabras, sino que están basados en distribuciones probabilísticas. Lo que hacen, en realidad, es buscar coincidencias en el *dataset* de entrenamiento en el que aparezcan elementos similares en un orden similar al presentado. Es evidente que cuanto mayor sea el conjunto de datos de

entrenamiento, más preciso será el modelo y también será menos probable encontrarse con términos que el modelo nunca ha procesado previamente.

Los modelos aprenden de los datos, por ello los primeros modelos que tuvieron unos resultados satisfactorios en la generación de texto fueron los modelos basados en RNN's.

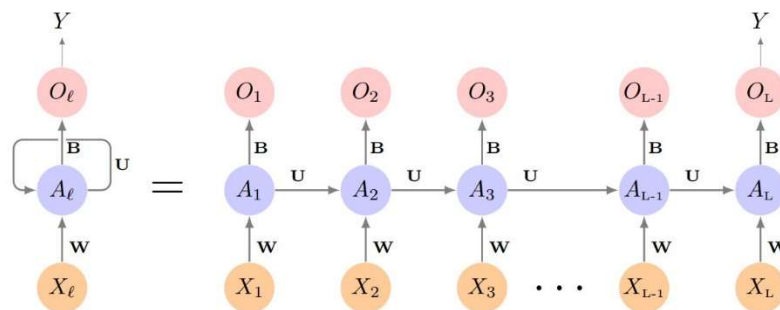


Ilustración 29: Representación de una RNN: Fuente [143]

Sin embargo, como ya se mencionó, los modelos basados en RNN tienen dos problemas. El primero es que son secuenciales, es decir, la salida de una neurona alimenta a la siguiente y a sí misma, pero tiene el inconveniente de que cada neurona debe esperar a recibir la entrada de la anterior para poder iniciar su proceso. El segundo problema encontrado es que la longitud de las secuencias está limitada debido a que la información sobre el contenido de la secuencia anterior se pasa como salida recurrente que finalmente desaparece. En pocas palabras, tienen poca memoria, ya que la información se procesa y se pasa a la siguiente neurona, pero esta no tiene constancia de los procesos anteriores, sino del resultado final. También aparece el problema de las palabras o de los *tokens* que no han sido vistos durante el entrenamiento del modelo, en este caso el modelo no será capaz de generar de forma autónoma “palabras inventadas”.

Este último problema fue abordado por el mecanismo de autoatención que permite a los modelos considerar partes de una secuencia o de un texto para generar predicciones. La idea era aprovechar el hecho de que, para longitudes de *token* más largas, el espacio de trayectorias que se encuentran efectivamente en el espacio lineal es escaso, lo que significa que en lugar de tener en cuenta todos los *tokens* anteriores para realizar la inferencia, los modelos RNN solo podrían entrenarse en un conjunto más pequeño de *tokens* “importantes”, que serían seleccionados por una red neuronal de “atención” separada.

La propiedad interesante de esa red neuronal de “atención” es que no es recurrente. No es necesario calcular un vector de atención para una secuencia antes de calcularlo en la otra, lo que significa que se puede paralelizar de manera eficiente.

3.4 Arquitectura de transformadores

En 2017 surge el artículo “*Attention is all you need*” [126] que revolucionó la arquitectura de las RNN's. Este nuevo modelo propuesto por Vaswani et al, no tiene elementos recurrentes y por tanto no necesita esperar a que se procesen los *tokens* anteriores. Desde este momento el entrenamiento de modelos ya no es secuencial, sino que los nodos que el sistema tenga disponibles pueden trabajar en paralelo. A partir de este momento, el modelo permite ser optimizado aumentando el tamaño del modelo, aumentando el conjunto de datos de entrenamiento y aumentando el potencial de cálculo o computacional de la máquina. Esto ha hecho que los modelos crezcan de forma exponencial en tamaño, aunque también es necesario tener en cuenta que la tendencia actual es utilizar modelos más modestos en tamaño, pero más optimizados en torno a unos casos de uso más específicos.

Los transformadores en general tienen dos partes que funcionan de forma independiente: el codificador y el decodificador. El primero analiza la entrada y da un valor numérico que es transformado a un vector con su representación espacial; el decodificador por su parte recibe ese valor y devuelve una palabra, siempre teniendo en cuenta las palabras colindantes para asegurar que se mantiene la coherencia. También es necesario mencionar que dependiendo de la actividad a la que se va a dedicar el modelo este puede utilizar solo el codificador como los modelos de comprensión de un texto o solo el decodificador como los modelos que generan texto. La Ilustración 30 muestra algunos de los modelos actuales clasificados en tres bloques, codificador-decodificador, solo codificador y solo decodificador.

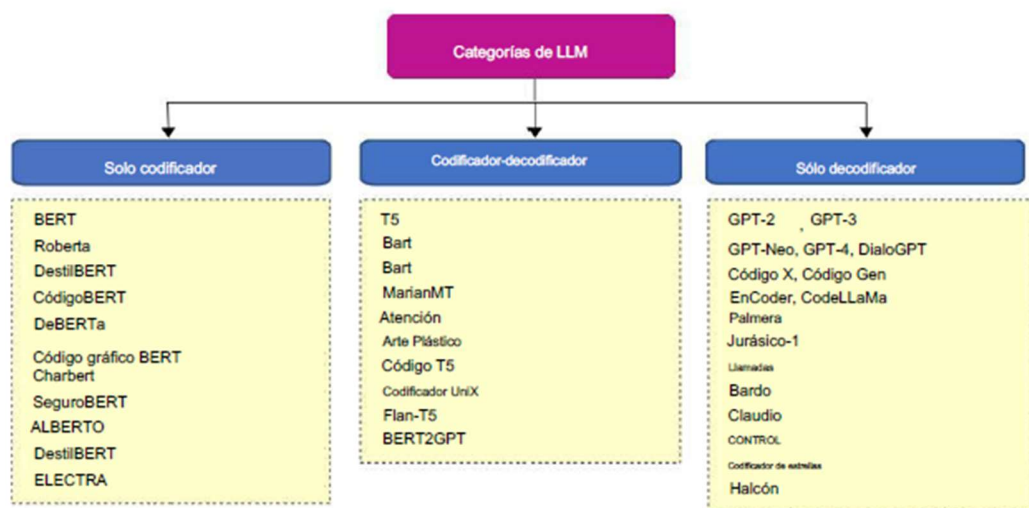


Ilustración 30: Clasificación de modelos LLM basados en transformador. Fuente [144]

Los transformadores destacan por tener una mayor velocidad de procesamiento de datos, ya que trabajan en paralelo. Tienen mayor precisión son capaces de capturar dependencias a largo plazo y relaciones complejas entre palabras de manera más eficiente, por eso destacan en traducción, generación de texto y análisis de sentimientos, esto los hace más versátiles y dan soporte a traductores automáticos, a sistemas de recomendación y a los famosos asistentes virtuales.

Los modelos LLM basados en los transformadores se han mostrado muy eficaces en dos tareas, la traducción de textos y en generar o descubrir una palabra que se oculta en una frase, que puede estar en cualquier posición.

3.4.1 Componentes clave de un transformador.

A continuación, se describen los componentes principales de un transformador y las acciones que desarrollan en el proceso de generación de un texto.

- *Embedding*: es el primer paso que se realiza por los transformadores en la operación de generación de texto, cada palabra de entrada se convierte en un vector de alta dimensionalidad. Este vector representa la palabra y el contexto y relación con el resto de las palabras con las que interactúa dentro de la misma frase o del texto en el que se encuentra. Los *embeddings* permiten que el modelo capture significados semánticos complejos.
- Mecanismo de atención: es una innovación introducida por los Transformadores, que va a hacer una valoración de cada palabra respecto a las demás.

- **Autoatención (*Self-Attention*):** Evalúa la importancia de cada palabra en una secuencia con respecto a las demás palabras. Es crucial para capturar dependencias a largo plazo.
- **Atención cruzada (*Cross-Attention*):** Se utiliza en modelos de traducción, permite al sistema alinear las palabras de la secuencia de entrada con las palabras de la secuencia de salida.
- **Red Neuronal *Feedforward* (FFNN),** también conocida como unidireccional o de propagación hacia adelante, en la que todas las palabras pasan por ella. Esta red procesa los *embeddings* enriquecidos por la matriz de atención para producir una nueva representación. Esto ayuda a refinar las características extraídas por el mecanismo de atención.
- **Mecanismo de normalización y residuales:** los transformadores utilizan estos mecanismos para acelerar su entrenamiento, los mecanismos residuales evitan el descenso de gradiente de un modelo.
- **Cabezas múltiples (*Multi-Head Attention*):** introduce múltiples mecanismos de atención en paralelo. Esto permite aprender diferentes aspectos de la relación entre palabras y mejor la capacidad del modelo para capturar relaciones complejas y matices sutiles que se encuentran en los enormes *dataset* de entrenamiento.

La combinación de estos componentes ha hecho que el modelo transformador tenga unos resultados muy buenos en comparación con los modelos anteriores, esto los hace una herramienta poderosa para el procesamiento avanzado del lenguaje. La combinación de *embeddings*, atención, redes unidireccionales, normalización y mecanismos de cabezas múltiples, permite a los transformadores superar las limitaciones de modelos anteriores y han establecido un nuevo estándar en la IA.

3.4.2 Funcionamiento de los transformadores

Es importante hacerse una idea de cómo funcionan los transformadores y como procesan la información, porque, aunque se basen en redes neuronales y estas a su vez se basen en el funcionamiento de las neuronas humanas, su forma de trabajo nada tiene que ver con la forma en la que los humanos procesamos la información.

El punto de partida inicial para procesar un texto es la tokenización de las palabras de este texto. Se trata de una operación similar a la que se hace en lengua castellana al separar una palabra en lexemas, pero evidentemente con diferentes resultados, ya que son cadenas que debe procesar y entender el propio modelo y no una persona, también hay que mencionar que hay diversos tipos de tokenización no es una operación única y universal [145].

En la tokenización se convierte una secuencia de texto en una serie de *tokens* (palabras o subpalabras). Los *tokens* son luego transformados en los *embeddings* que hemos visto en el apartado anterior y que el modelo puede procesar.

El siguiente paso es el mecanismo de autoatención, el modelo no trabaja con los datos de una forma secuencial sino en conjunto, con todo el conjunto de *embeddings* el modelo mediante cálculos determina que partes son más importantes y realiza valoraciones de todas las posibles interacciones entre cada una de ellas. Los vectores obtenidos pasan a través de la red unidireccional (*feedforward*) que los normaliza y a partir de ahí se generarían los resultados que el modelo va a ofrecer.

Posteriormente tiene lugar el proceso de entrenamiento. El entrenamiento va a requerir cantidades ingentes de datos y capacidad computacional para procesar toda esa información. A partir de esos datos, el modelo va a aprender por ejemplo a predecir la siguiente palabra de una secuencia de texto teniendo en consideración el contexto que aportan las palabras que tiene alrededor. Esto se denomina modelado

de lenguaje. El modelo a través del entrenamiento ajusta sus pesos y sesgos para minimizar la diferencia entre lo que predice y la palabra real que está en el texto original sobre el que entrena. Con grandes *dataset* y muchas repeticiones los modelos logran tener ajustes precisos y un porcentaje de acierto muy elevado.

En resumen, como se ha visto los pasos más destacados realizados por un modelo transformador serían los siguientes:

- **Tokenización:** cada palabra es procesada y convertida a uno o varios *tokens*
- **Embedding:** Cada *token* se convierte en un vector de alta dimensionalidad.
- **Autoatención:** El modelo evalúa la importancia de cada palabra en relación con las demás en la secuencia de texto original.
- **Red Unidireccional:** Los vectores resultantes se pasan a través de capas de redes neuronales densas para ajustar sus representaciones.
- **Generación de resultados:** El modelo puede utilizar esta información para realizar tareas como completar la frase o responder a preguntas sobre el contenido de ese texto.

3.4.3 Ventajas de los transformadores sobre modelos previos

Los transformadores superan a las RNN's y LSTM's en varios aspectos clave:

- **Procesamiento en paralelo:** A diferencia de las RNN's y LSTM's, que procesan secuencias de palabras una a una, los transformadores pueden procesar todas las palabras de una secuencia en paralelo. Esto aporta en una mayor eficiencia y velocidad de procesamiento, aunque parezca increíble son capaces de procesar todas las palabras de un texto de una sola vez de forma independiente.
- **Manejo de dependencias a largo plazo:** a través de mecanismo de atención, los transformadores pueden capturar relaciones a largo plazo entre palabras de manera más efectiva. Esto les permite generar texto y realizar traducciones con una precisión superior al que se realizaba con las RNN's y las LSTM's, hay un salto cualitativo muy importante de los LLM's antes y después de 2017.
- **Escalabilidad:** Los transformadores son más escalables y fáciles de entrenar en conjuntos de datos grandes, esto los hace ideales para aplicaciones que requieren altos volúmenes de datos.

En la Tabla 3 se muestran las principales diferencias entre las RNN's y los transformadores, basadas en [146].

Característica	RNN (incluidas LSTM y GRU)	Transformadores
Método de procesamiento	Secuencial	Paralelo
Manejo de secuencias largas	No maneja bien dependencias de largo alcance	Destaca gracias a la autoatención
Arquitectura	Estados recurrentes y ocultos	Autoatención
Eficiencia del entrenamiento	Lento, más difícil de realizar paralelismo de modelos	Más rápido, mayor capacidad para paralelismo de modelos
Eficiencia de la memoria	Requisitos de memoria menores	Alto uso de memoria
Aplicaciones comunes	Series temporales, primeras tareas de procesamiento del lenguaje natural	Procesamiento del lenguaje natural, traducción, generación de textos.

Tabla 3: Diferencia entre las RNN's y los transformadores. Fuente [146]

3.5 Familias de LLM's actuales

Los LLM's actuales generalmente están basados en la arquitectura de transformadores. Es cierto que a lo largo del trabajo se ha verificado que para algunas tareas concretas se obtienen mejores resultados de las RRN's o de las CNN's, pero la mayor parte de modelos que tienen relación con la ciberdefensa y la ciberseguridad son modelos basados en transformadores. En el capítulo 3 del libro *Large Language Models in Cybersecurity* [147] hay una interesante distribución de modelos en familias que se pasa a describir en los siguientes apartados, esta descripción se extiende a una lista de modelos y familias que no es exhaustiva, es decir no se recogen todos los LLM's que hay, ya que a día de hoy el número de modelos y familias es casi inabarcable.

3.5.1 Familia BERT

BERT es un modelo de codificador automático bidireccional desarrollado por Google en 2018, cuya función era inicialmente la de autocompletar consultas. Se puede considerar que es una versión *transformer* de uno de los primeros modelos denominado ELMo basado en RNN. Su entrenamiento se basó en descubrir la palabra oculta y en predecir la siguiente palabra en una oración.

Siguiendo con este modelo, han surgido variaciones sobre ella, que a continuación se detallan:

- RoBERTa es una modificación de BERT desarrollado por Facebook en 2019, en el que se optimizó el programa de entrenamiento, provocando un aumento del *dataset* de entrenamiento y en el que se eliminó la predicción de la siguiente palabra en una oración. La consecuencia directa es que es menos preciso en el autocompletado de consultas respecto a BERT, pero lo hace más adecuado para aplicaciones relacionadas con la anotación y codificación de texto.
- DistilBERT fue desarrollado por HuggingFace y se conoce como un modelo destilado³ del original BERT. Esto es así ya que su objetivo era reducir el tamaño de BERT a la vez que era capaz de conseguir una aceleración en la inferencia. De este modo, DistilBERT obtuvo un 97% del rendimiento de BERT con una aceleración de la inferencia de un 60% teniendo en cuenta una disminución de su tamaño en un 40%.
- ELECTRA, repitió la idea de BERT y ELMo, y sus autores propusieron entrenar el modelo para detectar tokens reemplazados por una muestra del generador, igualando el rendimiento de RoBERTa con solo una fracción de los datos y calculando los datos necesarios para el entrenamiento previo.

Los modelos de la familia BERT se utilizan comúnmente siempre que se necesita una base liviana para tareas de clasificación y llenado de espacios. No se espera que tengan un buen desempeño en tareas de generación de texto, pero aun así pueden usarse para ellas completando la última palabra de la secuencia.

3.5.2 Familia GPT

Los modelos GPT son modelos de lenguaje de uso general que pueden ser utilizados en un amplio abanico de tareas, sus siglas se corresponden a transformador generativo preentrenado GPT.

³ El termino destilación, a veces también llamada aprendizaje maestro-alumno es una técnica de compresión en la que se entrena un modelo pequeño para reproducir el comportamiento de un modelo más grande o de un conjunto de modelos.

- GPT-1: El primer modelo de GPT fue GPT-1, el cual reutilizó directamente el decodificador de la arquitectura *Transformer*, fue entrenado con *BookCorpus* que se basó en un conjunto de libros no editados que contenían alrededor de mil millones de palabras. Si bien se demostraron capacidades de generación elemental, el artículo GPT-1 se centró en ajustar los modelos previamente entrenados para aplicaciones específicas, como la vinculación.
- GPT-2: Se publicó en 2019 con diferentes tamaños 117M, 345M, 762M, y 1.5 Billones de parámetros. Basado en GPT-1, se centró en aumentar el número de capas que paso de 12 a 48 y el número de estados ocultos, conjunto de datos de entrenamiento, combinando los textos señalados por todos los enlaces salientes del popular sitio web Reddit. El conjunto de datos obtenido, de alrededor de 10 mil millones de palabras, se utilizó para entrenar todos los modelos GPT-2. En este sentido, por primera vez las habilidades generativas fueron los puntos de referencia, lo que indica un alejamiento total de las ideas iniciales de ELMo y BERT.
- GPT-3: incluye 8 modelos previamente entrenados, con tamaños que oscilaban entre 125M y 175B, además de la cantidad de capas, la cantidad de estados ocultos y la duración del período de atención. A los datos de entrenamiento usados por GPT-2, se añadieron datos de Wikipedia y de la librería *Common Crawl*⁴, las cuales sumaron aproximadamente 400M de palabras. Estos datos de entrenamiento han sido ponderados ya que podía haber sesgos debido a que no se muestrean con la misma frecuencia datos de Wikipedia o de *Common Crawl*. Este modelo ya era capaz de generar textos con la calidad suficientes equiparables a los que realiza una persona.
- InstructGPT: se trata de un ajuste fino que sigue instrucciones de un modelo base en la generación GPT-3, entrenando el modelo para imitar las respuestas de un asistente de IA que se ajuste a las expectativas del público en general.
Este ajuste se desarrolló en dos etapas. En primer lugar, el ajuste clásico a partir de ejemplos de conversaciones deseadas que convirtió el modelo en un agente conversacional. En segundo lugar, los trabajadores humanos clasificaron la calidad del resultado del modelo ajustado según varias métricas de evaluación. Sus comentarios se utilizaron para entrenar un modelo de “censura” para guiar la generación de texto y perfeccionar aún más el modelo.
Este ajuste secundario visto anteriormente es el aprendizaje por refuerzo a partir de la retroalimentación humana (RLHF).
- Modelos CODEX: a pesar de no contener la partícula GPT en su nombre, usa los mismos principios que el resto de la familia GPT. Se usan para generar código además de lenguaje natural. Utilizan varias fuentes entre las que están paquetes de PyPI de Python y el repositorio de GitHub. Existe una variante del modelo CODEX entrenada en más lenguajes de programación y entrenado con código alojado en GitHub.
- ChatGPT: combina los modelos InstructGPT y GPT-3.5 para crear el demostrador lanzado a finales de 2022. OpenAI no publicó ningún documento técnico para ChatGPT. Se cree que es una variante de GPT-3.5 que se sometió a una conversión más extensa siguiendo instrucciones y aprendizaje reforzado a partir de la retroalimentación humana.

3.5.3 Modelos derivados de GPT.

Estos modelos han tomado como punto de partida los modelos GPT para desarrollar sus soluciones a medida. Todas ellas son adaptaciones o modificaciones con objetivos concretos.

- GPT-neo-2.7B, GPT-j-6B y GPT-neoX-20B [15-17]: todos ellos son clones arquitectónicos de la familia GPT de OpenAI con 1.3B, 6B y 20B de parámetros respectivamente, con mejoras menores, aunque tienen un buen desempeño en comparación con otros modelos. En particular,

⁴ <https://commoncrawl.org/overview>

GPT-J-6B se ha utilizado con éxito para hacerse pasar por varios usuarios humanos de forma totalmente autónoma en un sitio web similar a un foro en un entorno del mundo real.

- HyperCLOVA, una copia de la familia GPT, pero escalada a parámetros 82B y específica del idioma coreano. El principal cambio es una modificación del tokenizador para adaptarse mejor al idioma coreano, así como una reducción en el tamaño del modelo en comparación con GPT-3.
- OPT: Facebook AI comenzó su esfuerzo por replicar a toda la familia y, a mediados de 2022, lanzó la familia OPT, un clon de la familia GPT-3.
- Gopher es un modelo de parámetros de 280B entrenado por Google de acuerdo con las leyes de escala presentadas por OpenAI en el documento de seguimiento de GPT-3. Gopherno se lanzó al público en general.
- La familia BLOOM es el resultado del intento del equipo de investigación de HuggingFace de 2022 de crear un LLM público comparable a GPT-3 y GPT-3.5 BLOOM se basa en el lado decodificador de la arquitectura *Transformer* y, para el mismo tamaño, tiene menos capas y más cabezas de atención por capa, así como un mayor número de dimensiones ocultas.

3.5.4 Modelos de cálculo óptimo

Los tres modelos generativos autorregresivos más visibles en esta categoría son Chinchilla de Google, las familias LLaMA y LLaMA2 de Facebook/Meta y Falcon. Con un tamaño que oscila entre los parámetros 70B para LLaMA y Chinchilla y 180B para Falcon, todos se comparan favorablemente con GPT-3-175B. Al mismo tiempo, las familias LLaMA y Chinchilla son más fáciles de implementar y administrar gracias a su menor tamaño.

- Chinchilla funciona bien en publicaciones, pero nunca se lanzó al público en general, ni siquiera en pruebas basadas en API. Por eso, tiene poca o ninguna relevancia para el usuario general.
- Falcon es otra familia de modelos de cálculo óptimo, del Instituto de Innovación Tecnológica. Falcon es una familia de 3 modelos, con tamaños de parámetros 7B, 40B y 180B.

3.5.5 La familia Llama

La familia LLaMA fue diseñada y entrenada por Meta AI para uso público, son modelos muy populares y que desde el inicio han permitido la instalación de forma local.

- Los modelos LLaMA de Facebook/Meta, cuya variante de parámetros es de 13,5 mil millones, se considera que igualan el rendimiento del modelo GPT-3 de 175 mil millones de parámetros. Más tarde surge LLaMA-1, un LLM de cálculo óptimo que se puede ajustar para diversas aplicaciones. Tanto LLaMA y LLaMA -1 han sido objeto de preocupación desde el ámbito de la ciberdefensa, ya que no se disponía de sus pesos, ni datos de entrenamiento y han sido muy utilizados para fines maliciosos.
- RedPajama y Open LLaMA: son clones de LLaMa. Se diseñaron para aliviar la inquietud sobre el origen de datos, comprender mejor el modelo LLaMA y permitir el desarrollo de clones de LLaMA utilizables comercialmente. En su desarrollo participó el Grupo Together, LIAON y Eleuther AI.
- LLaMA 2 fue la respuesta de Meta AI al éxito del modelo LLaMA-1, con algunas mejoras respecto al anterior, como más datos de preentrenamiento, tamaño de la ventana de autoatención y mayor velocidad de inferencia.
- Code LLaMA: es un desarrollo del modelo LLaMA 2 que ha sido entrenado previamente en código y ajustados para un contexto más extenso. Su principal objetivo es generar código de calidad, pero a su vez como se ha indicado también debe dar respuesta a tareas de propósito general. Es competencia directa de GitHub Copilot, basado en CODEX y StarCoder de OpenAI

3.5.6 Modelos de secuencia a secuencia

Tanto los modelos GPT como los de la familia BERT se han centrado en tareas específicas y generalmente se centra en la parte del codificador (normalmente no usan la parte del decodificador), por el contrario, los modelos de secuencia a secuencia conservaron las partes del codificador y del decodificador del transformador original y son entrenados para tareas generales de transformación de texto, y su objetivo principal es la traducción.

- BART: entrenado para “traducir” secuencias con *tokens* corruptos, eliminados, permutados o rotados a secuencias con *tokens* correctos de una manera que no es muy diferente a BERT.
- T5: está entrenado para tareas de propósito general, como traducción, respuesta a preguntas, y resúmenes.
- UL2: es el resultado de la investigación para unificar paradigmas de aprendizaje de idiomas.

3.5.7 Familia PaLM

PaLM es el único declarado como modelos de mezcla de expertos (MoE). PaLM aprovecha el programador Pathways para entrenar una variedad de modelos, desde parámetros 8B hasta 540B, al tiempo que inyecta una cantidad sustancial de datos de texto de interacciones en redes sociales y aproximadamente duplica los datos de entrenamiento usados en GPT-3. es capaz de realizar una amplia gama de tareas, incluido el razonamiento de sentido común, el razonamiento aritmético, la explicación de chistes, la generación de código y la traducción

3.5.8 GPT-4 y BingChat

A principios de 2023, Microsoft anunció la integración de un sucesor de ChatGPT con el motor de búsqueda de Microsoft Bing. Sin embargo, el agente conversacional LLM que impulsa el chat de búsqueda integrado en Bing tenía un conjunto de habilidades y comportamientos que diferían bastante de ChatGPT y, por lo tanto, justificaba el tratamiento como una generación separada de la familia GPT.

No se facilitaron datos técnicos de GPT-4. Pero se pueden extraer algunas conclusiones de demostraciones públicas, que da una idea de sus posibilidades:

- GPT-4 es Multimodalidad con respecto al reconocimiento de imágenes (tipo de objeto de imagen, color, naturaleza del logotipo). Autoinstrucción básica de razonamiento lógico para dividir consultas
- Realiza razonamiento lógico básico para agregar información adquirida a partir de consultas separadas
- Realiza identificación y resumen de páginas web grandes, como reseñas de productos, de manera cualitativa.
- Solicitar un mayor perfeccionamiento en caso de consultas que permitan múltiples interpretaciones
- Identificar explícitamente términos de búsqueda mal escritos, pero semánticamente similares, corrigiéndolos y pidiendo al usuario que aclare en caso de ambigüedad.
- Ofrecer escenarios realistas basados en búsquedas para posibles resultados futuros relacionados con un dominio, tecnología o franquicia de ficción específicos.
- Potencialmente, analizar e interpretar el sonido y las imágenes de videos para proporcionar un resumen e integrar dicho resumen en los resultados de las respuestas a las consultas.

Según esta información, GPT-4 parece ser un modelo multimodal con la integración de visión por computadora y potencialmente modelos de reconocimiento de voz; previamente entrenado y ajustado para hacer referencia a capacidades de búsqueda auxiliares.

En el [Anexo II: Listado de LLM's open source](#), se ha extraído una lista de LLM's asociados a su tipo de licencia, que figuran en [148].

4 CASOS DE USO GENÉRICOS Y POSIBLES IMPLEMENTACIONES EN DEFENSA

4.1 Usos de los LLM's

Los LLM's pueden ser utilizados en multitud de aplicaciones. En el siguiente apartado se hará un resumen de las aplicaciones y casos de uso más generales en los que se utilizan actualmente los LLM's, para posteriormente centrarse en aquellos casos de uso que pueden dar soluciones o respuesta a los retos que se plantean específicamente en los ámbitos de la ciberseguridad y la ciberdefensa en el MINISDEF.

4.2 Casos de uso generales más frecuentes

Podemos encontrar uso de la IA y aplicaciones de los LLM's casi en cualquier ámbito de nuestra actividad diaria. La tecnología ya es parte de nuestras vidas y se puede decir que nos acompaña prácticamente allí donde vamos, desde computadoras, teléfonos móviles, relojes inteligentes, pulseras de actividad, Smart TV, etc., a cada paso que damos nos encontramos con una aplicación de la IA y de los LLM's.

Existen muchas publicaciones en las que se detallan los casos de uso más frecuentes de los LLM's, como en [149], [150] y [151]. A continuación, podemos mencionar los siguientes como más relevantes.

- Asistencia en diagnósticos clínicos ([152], [153], [154]): La IA ya se utiliza con modelos lingüísticos de ayuda al diagnóstico como un caso de éxito ([128], [129], [130]). En concreto se tiene constancia de que OpenAI ha participado en el desarrollo de este concepto del que se desprenden resultados que han reducido los errores en los diagnósticos un 20% y han acortado las listas de espera de pacientes. El siguiente paso es la integración de la IA quirúrgica para mejorar la precisión en entornos de alto riesgo.
- Seguimiento y análisis de opiniones de clientes: Son una fuente importante para las decisiones basadas en datos. Así, se utilizan las reseñas de los clientes, ya sean quejas o valoraciones positivas de alguna característica concreta de cualquier producto. Los LLM's se usan para en la obtención de la valoración del análisis de opiniones, ya que realizan estas operaciones en tiempo real como se indica en [155] y [156].
- Creación de mapas semánticos de temas complejos ([157], [158]): Se emplea para crear mapas semánticos de temas muy específicos y complejos como créditos fiscales, investigación médica o marketing.
- Resumen del contenido: Realización de resúmenes de textos largos como artículos, noticias, informes de investigación, son textos por lo general muy largos que son adaptados a la longitud de salida que especifica el usuario ([159], [160], [161]).
- Asistentes de IA: esta es una de las más conocidas. Son chatbots que responden consultas de los usuarios en lenguaje natural, aunque siempre con cierta cautela. Podemos encontrar sugerencias en [162].
- Generación de código: ayuda para desarrolladores tanto para la codificación en cualquier lenguaje de programación, como para la detección de errores durante la programación. Se puede encontrar análisis de los mejor valorados en [163], [164] y [165].

- **Análisis de emoción:** a través del análisis de un texto se puede determinar el estado anímico del usuario. En este sentido, pueden tener varias aplicaciones: desde conocer la satisfacción de los clientes, hasta predecir posibles conductas suicidas, como se indica en [166] y [167].
- **Traducción de idiomas:** realización de traducciones multilingües, muy usadas tanto en la empresa como en el mundo universitario. También existen LLM's para lenguajes específicos como se puede extraer de [168] y [169].
- **Creación de contenidos:** normalmente utilizado por los usuarios para la creación de textos, relatos breves, resúmenes, encuestas, etc. Algunas de las herramientas más usadas son DALL-E, MidJourney y Stable Diffusion que crean imágenes basadas en indicaciones del usuario ([170], [171], [172]).
- **Asistentes virtuales y atención al cliente [173]:** asistente de clientes para dudas y consultas, también asistencia virtual para quejas y en general atención al cliente.
- **Transcripción:** faceta muy interesante de transcripción de archivos de audio y de video que funcionan con buenos resultados. También se utilizan para subtitular videos o películas a otros idiomas ([174], [175], [176], [177], [178]).
- **Estudios de mercado [179]:** la IA generativa se utiliza para, a través del estudio de los datos, realizar inferencias a partir de información de mercados, productos, servicios, clientes, competidores o cualquier información relevante que sirva de base para realizar inversiones.
- **Investigación de palabras clave[180]:** uso de la IA para obtener palabras clave de un tema concreto, junto con términos que estén relacionados con el mismo.
- **Estimación de ventas [181]:** se puede usar para hacer estimaciones de ventas futuras o para anticipar una línea de ventas que debe ser explotada.

Se pueden encontrar otros usos declarados de los LLM's como pueden ser, mejorar la eficacia de los cirujanos, atención al público en los bancos, tutoría para estudiantes, tomar notas en reuniones, recomendaciones legales, ayuda en auditorias, gestión de contratos en las administraciones públicas, proyectos de construcción, ofrecer valoraciones y peritaciones, o la generación de noticias.

Se ha confeccionado la Tabla 4, con la información que figura en [162], en la que se hace una relación entre algunos casos de uso y los LLM's que se sugieren para ese caso, también se pueden considerar algunos casos de uso como campos o ámbitos de actividad.

Caso de uso	Software
Generación y resumen de contenidos	ChatGPT, Jasper, ChatSonic, Bard, copy.ai, HuggingChat, ZenoChat
Servicio de atención al cliente	Zendesk
Navegar por la web	Bard, Bing, Perplexity
Traducciones	Bard, ChatGPT
Entretenimiento	MY AI, Character.AI, Replika, Pi
Educación	Khanmigo, Socratic de Google
Salud mental	Woebot

Caso de uso	Software
Investigación	You.com
Conversación	Character.AI, Replika, Claude
Recursos humanos y contratación	Ideal
Codificación	ChatGPT, Amazon CodeWhisperer
Redes sociales	Mi IA
Recogida, análisis y visualización de datos	Bing, Rose AI
Gráficos	Bing, Jasper, Character.AI, ChatSonic
Gestión de tareas	HuggingChat, Claude
Marketing	ChatSpot
Ventas	Claude

Tabla 4: Recomendación LLM's según el caso de uso. Fuente [162]

Los casos expuestos en este epígrafe no son los únicos. Hay muchos más casos de usos y seguirán aumentando en el futuro, esto solo es un resumen de algunos de ellos que tienen impacto en nuestra vida diaria.

4.3 Casos de uso en el ámbito de la Defensa.

La ciberseguridad, según [182] comprende todas las tecnologías, prácticas y políticas para prevenir los ciberataques o mitigar su impacto. La ciberseguridad tiene como objetivo proteger los sistemas informáticos, las aplicaciones, los dispositivos, los datos, los activos financieros y las personas contra el *ransomware* y otros *malware*, las estafas de *phishing*, el robo de datos y otras ciberamenazas

La guía de ciberseguridad de INCIBE [183] marca como objetivo los mecanismos y prácticas que sirven para proteger nuestros dispositivos y nuestra privacidad cuando navegamos por la red.

A modo de definición, para la empresa de seguridad Kaspersky [184], la ciberseguridad es la práctica de defender las computadoras, los servidores, los dispositivos móviles, los sistemas electrónicos, las redes y los datos de ataques maliciosos. También se conoce como seguridad de tecnología de la información o seguridad de la información electrónica.

Podemos encontrar muchas definiciones sobre el mismo término ([185], [186], [187]), pero en general todas ellas hacen alusión a los mismos principios y a los valores a proteger. En general este concepto está relacionado con un conjunto de procedimientos y herramientas cuyo objetivo es proteger la información que se procesa a través de los dispositivos TIC, esta protección se puede hacer extensiva a los propios medios TIC y a los usuarios finales.

Del mismo modo, la ciberdefensa, según [188] es el conjunto de medidas y acciones que se llevan a cabo para proteger los sistemas informáticos y las redes de comunicaciones de los ataques informáticos. Podemos encontrar también otra definición de ciberdefensa en [189] y [190], como el conjunto de operaciones activas o pasivas utilizadas por el Estado para garantizar la seguridad y el uso adecuado del ámbito digital de un país. Respecto a estas últimas, ya involucran de manera expresa al Estado o el

Gobierno del país, y su campo de actuación sería todo el ámbito digital nacional, algo que es muy relevante desde el punto de vista de la protección que siempre está ligada al MINISDEF.

En [191], encontramos también una pequeña explicación de la diferencia entre ciberseguridad y ciberdefensa: mientras la primera tiene un rol preventivo, la segunda se centra en resistir activamente un ataque. Por tanto, la ciberseguridad es complementaria a la ciberdefensa. Tanto en la ciberseguridad como en la ciberdefensa, se busca neutralizar y controlar las amenazas, pero concretamente la ciberdefensa, destaca no solo por su rol preventivo ante los ataques, sino que responde a los mismos con el fin de proteger a las empresas, a los bienes y a los usuarios.

Entonces, tanto la ciberdefensa como la ciberseguridad se circunscriben al ámbito TIC y tienen su ámbito de actuación en el ciberespacio. Esto nos conduce a otra pregunta, ¿Qué es el ciberespacio?

Existen en la actualidad diferentes definiciones. Así, según el diccionario de la RAE, ciberespacio es el ámbito virtual creado por medios informáticos, para el Departamento de Defensa de los Estados Unidos ciberespacio es un dominio global dentro del entorno de la información que consiste en redes interdependientes de infraestructuras de tecnología de la información y datos residentes, incluidos Internet, redes de telecomunicaciones, sistemas informáticos y procesadores y controladores integrados [192].y según el Centro Superior de Estudios de la Defensa Nacional [193], se trata de un nuevo ámbito que, junto con los tradicionales de tierra, mar, aire y espacio, es el medio donde se desarrollan las actividades económicas, productivas y sociales de las naciones modernas.

En España, según [193] el ciberespacio ya ha sido definido y aceptado como el quinto dominio, en el cual se llevarán a cabo operaciones militares específicas y que ya apoyan actualmente las operaciones militares que se desarrollan en el resto de dominios (Tierra, Mar, Aire y Espacio). Las Fuerzas Armadas de la era de la información dependen, cada vez más, del uso libre del entorno de la información y el ciberespacio para llevar a cabo todo el espectro de operaciones.

Por tanto, cualquier nación moderna y técnicamente avanzada, manifestará su intención de defender sus activos e intereses estratégicos en dicho ámbito, en el caso de España se han publicado leyes para la protección de las infraestructuras críticas como la Ley 8/2011, de 28 de abril, por la que se establecen medidas para la protección de las infraestructuras críticas, el Real Decreto 704/2011, de 20 de mayo, por el que se aprueba el Reglamento de protección de las infraestructuras críticas y lo que es más aplicable a nuestro caso la Estrategia Nacional de Ciberseguridad, que se vienen publicando desde 2013 y cuya última publicación es de 2019 [194].

En el documento [195] se establece como propósito que España, tal y como indica la Estrategia de Seguridad Nacional de 2017, debe garantizar un uso seguro y responsable de las redes y los sistemas de información y comunicaciones a través del fortalecimiento de las capacidades de prevención, detección y respuesta a los ciberataques potenciando y adoptando medidas específicas para contribuir a la promoción de un ciberespacio seguro y fiable.

El propósito de la Estrategia Nacional de Ciberseguridad 2019, es fijar las directrices generales del ámbito de la ciberseguridad de manera que se alcancen los objetivos previstos en la anterior Estrategia de Seguridad Nacional de 2017.

Algunas de las medidas recogidas en las líneas de acción que figuran en [194] son las siguientes:

- Potenciar las capacidades de ciberdefensa y de ciberinteligencia.
- Implantar medidas de ciberdefensa activa en el sector público con el objetivo de mejorar las capacidades de respuesta.
- Ampliar y fortalecer las capacidades de prevención, detección, respuesta, recuperación y resiliencia a los ciberataques dirigidos al sector público, a los servicios esenciales y a empresas de interés estratégico.
- Potenciar el desarrollo de la normativa sobre protección de infraestructuras críticas, reforzando la seguridad de las redes y sistemas de información que las soportan.

- Promover la realización de ciberejercicios y evaluaciones de ciberseguridad, especialmente en áreas que puedan afectar a la Seguridad Nacional, la Administración pública, los servicios esenciales y las empresas cotizadas.
- Fomentar el intercambio de información, experiencia y conocimientos, entre el personal con responsabilidades en la investigación y persecución de la cibercriminalidad.
- Impulsar programas específicos de I+D+i en ciberseguridad y ciberdefensa.

Es en este contexto donde debe participar el MINISDEF, para la defensa de sus redes y sistemas TIC propios y en apoyo a los del resto de ministerios, infraestructuras críticas y del tejido empresarial en casos de crisis o conflicto.

De la propia Estrategia Nacional de Ciberseguridad se pueden extraer las principales ciberamenazas a las que debemos hacer frente, en la Ilustración 31, se les pone nombre, serían el ciberespionaje, las amenazas híbridas, el cibercrimen y el hacktivismo. De estas amenazas globales saldrán los casos de uso particulares que pueden afectar a Defensa.



Ilustración 31: Ciberamenazas y sus principales acciones. Fuente [194]

Las fuentes de estas amenazas son variadas según [194], pero se podrían resumir en las siguientes: los cibercriminales, los espías industriales, los hacktivistas, los terroristas nacionales o internacionales, naciones con intereses contrapuestos a los nuestros, los hackers, y personal interno que por descontento, ambición y otras razones pueda actuar contra los intereses nacionales.

Y sus motivaciones también pueden ser variados, pero se podrían destacar el beneficio económico, la ventaja táctica o competitiva, motivaciones políticas o religiosas, fama o venganza y, por último, el deseo de provocar daño o destrucción.

La ciberseguridad y la ciberdefensa están en constante evolución con el fin de proteger activos que pueden ser vitales para un país, para un gobierno y para sus ciudadanos. Los constantes ataques generan millones de euros en pérdidas como se relata en [196], además del coste reputacional y otros inconvenientes o trastornos que puedan causar.

En la actualidad hay una gran dependencia de medios TIC, tanto de organizaciones públicas, como del sector privado, y también para los individuos en su esfera particular. Todos dependemos de las tecnologías digitales en nuestro día a día, para las comunicaciones, para la información, para los movimientos de personal, para el comercio, para la medicina, para el ocio, y también para la gestión de las infraestructuras críticas.

Los LLM's han surgido como una tecnología innovadora con el potencial de mejorar significativamente las prácticas de ciberseguridad. No todos los LLM's tienen resultados iguales en los diferentes casos de uso que se pueden presentar. Por este motivo, se estudiarán aquellos LLM's que pueden tener una mejor adaptación o aplicación a los casos de uso a los que el MINISDEF debe dar respuesta.

Los LLM's no son una novedad, pero sí que, como se ha visto a lo largo del trabajo, han experimentado una explosión de uso desde finales de 2022. El entrenamiento de LLM's y sobre todo el ajuste fino (*fine tuning*) no están al alcance de todos. En este sentido siempre se puede tomar como referencia al Departamento de Defensa de EEUU, no solo por estar siempre a la vanguardia en tecnologías, sino porque es uno de los pocos que puede costear desarrollos de I+D.

En [197] se hace un repaso del uso de LLM's en la seguridad nacional de EEUU y de las diferentes posiciones que han tomado la Fuerza Aérea, USAF (*United States Air Force*), la Marina de Guerra, USN (*United States Navy*) y el Cuerpo de Marines (*USMC United States Marine Corps*) respecto al uso de los LLM's. Según este documento, la USAF lo ha utilizado para crear escenarios de juegos de guerra y en resúmenes automáticos sobre la actividad.

Para el Departamento de Defensa de EEUU, los LLM's son más adecuados para funciones de apoyo en lugar de liderar decisiones estratégicas. Concretamente, en ese trabajo se habla de que las decisiones de mando tomadas por un LLM eran muy agresivas, a la vez que en las pruebas realizadas aparecían alucinaciones. Ahora bien, es indudable que destacan para actividades como el resumen o el apoyo al mando basado en datos, pero se descarta su uso activo en la toma de decisiones.

Por otro lado, los Marines crearon un LLM denominado Hermes, para la ayuda a la planificación militar y los utilizan también en el ámbito académico. En cambio, la Armada de EEUU, ha tomado un enfoque más cauteloso y su postura es una política que desaconseja el uso de modelos de IAG comerciales para fines operativos hasta que se hayan investigado y aprobado por completo los requisitos de seguridad. Ahora bien, se tiene constancia de que el Departamento de Defensa y sus agencias utilizan aplicaciones de IA generativa como la CIA, DARPA que es la Agencia de Proyectos de Investigación Avanzada de Defensa y también MITRE que colabora con OTAN, como se asegura en [198] y [199].

En el documento “concepto de ciberdefensa, resumen ejecutivo” del Estado Mayor de la Defensa, establece que las FAS, y el MINISDEF en general, se encuentran ante el reto de potenciar sus actuales capacidades de ciberdefensa para:

- fortalecer la resiliencia,
- operar de forma continuada, ágil y eficaz en el cada vez más exigente escenario operativo,
- y ser capaces de evolucionar y adaptarse al ritmo que lo hacen las tecnologías y la propia amenaza.

Hasta el momento no hay grandes desarrollos en este ámbito, aunque ya se empieza a trabajar en los primeros proyectos de aplicación de la IA, como el proyecto SAPIEM, (Sistemas y Servicios de Apoyo para la Interoperabilidad JISR) de despliegue de herramientas de inteligencia, y otros pequeños proyectos en el marco de la iniciativa BACSI (Base Aérea conectada sostenible Inteligente) del Ejército del Aire y del Espacio, que aumentaran en los próximos años.

En el día a día, hay muchas facetas que es necesario proteger, que serán los casos de uso que se tratarán en las siguientes secciones, y en los que de forma acertada se podrían implementar LLM's, optimizando así los resultados y los recursos aplicados dentro del MINISDEF.

Estos casos de uso son a nuestro juicio los más relevantes en los que se podría aplicar los LLM's en el ámbito de la Defensa, tomando como referencias las encontradas en [200], [201], [202] y [179]:

- Detección y análisis de amenazas
- Automatización de la seguridad
- Detección y respuesta al *phishing*
- Análisis forense
- Pruebas de penetración
- Respuesta automática a incidentes
- Chatbots de seguridad
- Capacitación y concientización sobre seguridad
- Modelado de contraseñas
- Creación de *Honeypot*
- IDS (*Intrusion Detection System*)

4.3.1 Detección y análisis de amenazas

Según se puede extraer de [203], los LLM's son capaces de analizar grandes volúmenes de datos en tiempo real, su gran aportación es que pueden reconocer patrones indicativos de ataques, como *malware*, intentos de *phishing* y tráfico de red inusual. Además, son capaces de predecir nuevos patrones de ataques, lo que los hace mucho más efectivos que el análisis de tráfico de red basado en reglas tradicional.

En este ámbito ya se han realizado estudios previos como [204] y [205] mediante el uso de redes neuronales recurrentes para detección de intrusiones (RNN-ID), que parecen aportar mejores resultados para este tipo de casos de uso. En [206] también se han utilizado RNN's con GRU y CNN's apiladas para detectar ciberataques.

4.3.2 Automatización de la seguridad

Este tipo de actividades pueden ser desempeñadas perfectamente por un LLM. Son tareas rutinarias que no tienen una complejidad técnica excesiva. Estas tareas pueden ser la evaluación de vulnerabilidades, las comprobaciones de cumplimiento, la gestión de versiones y de parches de seguridad. Estas tareas, aunque son rutinarias y repetitivas son muy importantes para la seguridad de un sistema como se apunta en [207].

La automatización va a proporcionar un menor tiempo de respuesta a los incidentes; va a reducir la intervención humana, lo que va a provocar que pueda dedicar esfuerzos a otras tareas más necesarias; va a mejorar la precisión del sistema y va a reducir los falsos positivos. Algunas de estas prácticas las podemos encontrar en [208].

4.3.3 Detección y respuesta al *phishing*

Los LLM's han avanzado mucho en la identificación de correos de *phishing* mediante el análisis del propio texto, del asunto y de los adjuntos. La detección ha mejorado mucho y tiene un porcentaje cada vez más alto de filtrado, pero eso no descarta que, a su vez, la parte contraria también pueda estar utilizando los LLM's para el propósito contrario.

En [209] y [210] se hacen estudios sobre el uso de modelos LLM para detectar y evitar el *phishing*, basados en general en modelos BERT. En [211], se hace un estudio de como los atacantes aprovechan los LLM's para el caso de uso ofensivos y la efectividad de los detectores de *phishing* tradicionales frente a estos ataques conducidos por LLM's.

4.3.4 *Análisis forense:*

Los LLM's son una ayuda muy eficaz que en el análisis forense analizando registros y datos para determinar la causa y el método del ataque, ayudando así en el proceso de recuperación y en futuras estrategias de prevención como se apunta en [188], [189], [190] y [191].

4.3.5 *Pruebas de penetración:*

Los LLM's son utilizados para generar scripts y para realizar pruebas de penetración. Ya existe algún modelo específico de LLM cuya finalidad es el *pentesting*. En [212] se hace un estudio de la explotación de vulnerabilidades usando LLM's como los de la familia GPT, LLaMA, y modelos Mistral; y en [213] se presenta un sistema de prueba de penetración como es PentestGPT.

4.3.6 *Respuesta automática a incidentes:*

Durante un incidente de ciberseguridad, los LLM's pueden ayudar proporcionando un análisis rápido de la situación, sugiriendo estrategias de mitigación y automatizando respuestas como se indica en [214], [215] y [216].

4.3.7 *Chatbots de seguridad:*

Los LLM's podrían ser utilizados para ofrecer a los responsables de un equipo de seguridad informes de estado, informes de uso, incidentes, vulnerabilidades nuevas, asistencia en tiempo real, y también pueden utilizarse para realizar simulaciones y preparar al personal frente a situaciones críticas, como se extrae de [217] y [218].

4.3.8 *Capacitación y concientización sobre seguridad:*

Los LLM's pueden generar materiales de capacitación adaptados a las necesidades de una organización como figura en [219] y [220]. Actualmente los medios se basan casi en exclusiva en videos y presentaciones pdf o ppt, lo que los hace poco eficaces y desactualizados. También pueden simular ataques de *phishing* y otros escenarios de seguridad para capacitar a los empleados a reconocer y responder a las amenazas de seguridad. Las grandes empresas ya comienzan a utilizar este apoyo en la formación y concienciación [220].

4.3.9 *Modelado de contraseñas:*

En [221] podemos encontrar una guía y un análisis completo de PassGPT, herramienta basada en GPT-2 que sirve para la adivinación de contraseñas y realizar la estimación de su fortaleza frente a los ataques, ya que es útil para generar contraseñas fuertes.

4.3.10 Creación de Honeypot

Este caso de uso es ciertamente extremo, ya que se trata de la creación de herramientas utilizando LLM's para actuar como señuelos y atraer atacantes para interactuar con ellos. De esta forma se puede obtener información de sus métodos, de las prácticas y medios que utilizan [222] y [223]. En [224] y [225] se realiza un estudio exhaustivo sobre este asunto utilizando un LLM con ajuste fino partiendo de ChatGPT.

4.3.11 IDS (Intrusion Detection System)

Un sistema de detección de intrusiones es uno de los anhelos a nivel de Defensa que pueda optimizar la vigilancia de las redes y de los sistemas que se despliegan. En [144] se ha realizado un estudio completo de algunos LLM's específicos para esta actividad y arquitecturas que dan buenos resultados en la detección de intrusiones. Destacan modelos creados a partir de BERT, RoBERTa y CANBERT [226].

Para la realización de este trabajo se han tomado como referencia muchos artículos de investigadores y varias revisiones sistemáticas de otros autores, que han recopilado numerosa información referente al uso de los LLM's entre los que se destacarían los siguientes: [227], [228], [229], [230] y [231].

En [231], se hace un estudio donde se analizan diversos LLM's. La Tabla 5 es justamente una recopilación de esos LLM's, en las que se relacionan los casos de uso, actividades defensivas y la publicación en la que se ha realizado el estudio.

Actividad	Uso	Modelo	Fuente
Proteger	Uso de LLM para el desarrollo de código seguro sin comprometer la funcionalidad	SVEN (GPT-2), Generación de código	[232]
Proteger	LLM's resuelven los desafíos de Capture The Flag para mejorar la experiencia de los empleados. Conciencia y conocimiento.	gpt2-csrc	[233]
Proteger	Los LLM's investigan vulnerabilidades del software.	GPT-3.5 Turbo, Geminis, Microsoft Bing	[234], [235]
Proteger	Los LLM's investigan vulnerabilidades del software.	GPT-3.5 Turbo.	[236]
Proteger	Generación de honeywords utilizando LLM	GPT-3	[237]
Proteger	Categorización de URL basada en LLM para la clasificación de sitios web..	eExponer (Conv), BERTiny, URLTran (BERT), T5 Grande, GPT3 Babbage	[238]
Proteger	Los LLM's investigan vulnerabilidades del código.	GPT-3	[239]
Detectar	Detección de comportamiento anómalo en registros de red con LLM.	RNN	[240]
Detectar	Detección de vulnerabilidades en el código software.	GPT-2	[241]
Detectar	SecureBERT para detección de anomalías.	SecureBERT (RoBERTa)	[242]

Detectar	CyBERT, un modelo BERT específico de dominio. Reconocer entidades especializadas en ciberseguridad.	Filtro de lenguaje natural basado en BERT	[243]
Detectar	Pruebas de penetración con LLM.	GPT-3.5	[244]
Detectar	CyBERT, un clasificador de reclamaciones de características de ciberseguridad.	CyBERT, GPT-2	[245]
Detectar	CySecBERT para detección de malware y agregación de alertas.	CySecBERT	[246]
Detectar	SecureBERT para procesar y comprender textos sobre ciberseguridad. específicamente Inteligencia sobre Amenazas Cibernéticas (CTI).	SecureBERT	[246]
Detectar	Detección de vulnerabilidades en el código software.	SecureFalcon (FalconLLM)	[247]
Responder	Creación de honeypots relacionados con la monitorización y detección continua de amenazas.	GPT-3.5 Turbo (shellM)	[248]
Responder	LLM como interfaz honeypot contra ataques de línea de comandos.	GPT-3.5	[222]
Detectar	LLM que actúan como equipos rojos en ciberseguridad	GPT-4 y Bard	[249]
Detectar	LLM para detectar sitios de phishing a provecha un rastreador web. Para recopilar información y generar prompts.	GPT-3.5 y GPT-4	[250]
Detectar	KARTAL, un sistema de detección de vulnerabilidades en aplicaciones web.	GPT-3.5 Turbo	[251]
Detectar	Herramienta de prueba de penetración automática potenciada por LLM.	PentestGPT, GPT-3.5 y GPT-4	[252]
Detectar	Los LLM realizarán auditorías de seguridad en Smart contracts	GPT-4 (GPT-4-32k), Claude-v1.3-100k	[253]
Detectar	Los LLM realizarán auditorías de seguridad en Smart contracts	GPT-3.5 Turbo y GPT-4	[254]
Responder	Responder a los correos electrónicos fraudulentos utilizando LLM.	GPT-3	[255]

Tabla 5: LLM's por casos de uso actividades defensivas. Fuente [231]

En la Tabla 5 ya se puede observar la aparición de modelos específicamente desarrollados para el ámbito de la ciberseguridad como CyBERT, CySecBERT, y SecureBERT y también el modelo para pruebas de penetración PentestGPT. Con los datos de la Tabla 5, se ha confeccionado la Ilustración 32 que facilita la relación de los casos de uso defensivos y los modelos LLM's

Uso	Modelo	Modelo	Modelo	Modelo	Modelo	Modelo	Modelo	Modelo	Modelo	Modelo	Modelo	Modelo
Uso de LLM para el desarrollo de código seguro sin comprometer la funcionalidad												
LLM's resuelven los desafíos de Capture The Flag para mejorar la experiencia de los empleados.												
Conciencia y conocimiento.												
Los LLM investigan vulnerabilidades del software.	Gemini											
Los LLM investigan vulnerabilidades del software.												
Generación de honeywords utilizando LLM												
Los LLM investigan vulnerabilidades del código.												
Detección de vulnerabilidades en el código software.												
SecureBERT para detección de anomalías.												
Pruebas de penetración con LLM.												
CyBERT, un clasificador de reclamaciones de características de ciberseguridad..												
CySecBERT para detección de malware y agregación de alertas.	CySecBERT											
SecureBERT para procesar y comprender textos sobre ciberseguridad específicamente Inteligencia sobre Amenazas Cibernéticas (CTI).												
Detección de vulnerabilidades en el código software.												
Creación de honeypots relacionados con la monitorización y detección continua de amenazas.												
LLM como interfaz honeypot contra ataques de línea de comandos.												
LLM que actúan como equipos rojos en ciberseguridad												
LLM para detectar sitios de phishing a provecha un rastreador web. Para recopilar información y generar prompts.												
KARTAL, un sistema de detección de vulnerabilidades en aplicaciones web.												
Herramienta de prueba de penetración automática potenciada por LLM.												
Los LLM realizarán auditorías de seguridad en Smart contracts												
Los LLM realizarán auditorías de seguridad en Smart contracts												
Responder a los correos electrónicos fraudulentos utilizando LLM.												

Ilustración 32: Tabla de modelos por casos de uso defensivos. Fuente autor basado en[231]

De esta tabla se puede extraer que los modelos de la familia GPT son los que más casos de uso pueden abordar. Destacan por importancia GPT-4, GPT-3.5 Turbo, GPT-2, GPT-3.5, GPT-3, y aparecen los mencionados anteriormente como específicos o especializados, CyBERT, CySecBERT, SecureBERT, PentestGPT. Figuran también dos generales como Gemini y Claude para la investigación de vulnerabilidades de software y auditorías de seguridad de *smart contracts* respectivamente.

La Tabla 6, muestra los LLM's que se han estudiado para actividades ofensivas, según los casos de uso indicados, y relacionados con la fuente de estudio de los mismos. Como se puede ver, son modelos tradicionales que se utilizan para acciones ofensivas o más bien destructivas, estos modelos no requieren un entrenamiento específico y son realmente eficientes generando código malicioso, *phishing* de calidad o incluso *fake news* y desinformación.

Actividad	Uso	Modelo	Fuente
Ejecución 2023	Generar código para realizar acciones que podrían ser maliciosas	GPT-3	[256]
Acceso inicial 2022	Generar correos electrónicos de <i>phishing</i> para evitar los filtros de spam	GPT-2, GPT-3, RoBERTa	[257]
Ejecución 2022 - Comando y control	Uso de LLM como complementos para actuar como proxy.	GPT-4	[258]
Acceso inicial 2023 - Colección	Generar sitio web de <i>phishing</i> a través de ChatGPT	GPT-3.5 Turbo.	[259]
Ejecución 2023	Generación de código e inyección de DLL	GPT-3	[260]
Acceso inicial 2023: reconocimiento	Recopilación de datos de las víctimas para desarrollar un correo electrónico de ataque	GPT-3.5, GPT-4	[261]
2023 Acceso inicial. Ejecución - Evasión de defensa	Elaboración de scripts maliciosos	GPT-3.5 Turbo, GPT-4, text-davinci-003	[262]
Acceso inicial 2018	Enlace de <i>spear phishing</i>	AWD-LSTM	[263]
Evasión de defensa 2023	Ofuscación de código, modificación del formato de archivo	GPT-3.5	[264]
Acceso inicial 2023: acceso a credenciales	Adivinación de contraseñas mediante LLM	GPT-2	[265]
Acceso inicial 2023: reconocimiento	Suplantación de identidad con fines de <i>phishing</i>	GPT-3.5 Turbo	[266]
Acceso inicial 2022	Generando contenido para desinformar.	GPT-2	[267]

Tabla 6: LLM's por caso de uso en actividades ofensivas. Fuente [231]

Se ha simplificado la tabla en un nuevo formato agrupado por modelos para una mejor comprensión de los casos de uso que son factibles para cada modelo, como se puede apreciar en la Tabla 7.

Uso	GPT-2	GPT-3	GPT-3.5	GPT-3.5 Turbo.	GPT-4	RoBERTa	text-davinci-003	AWD-LSTM
Generar código para realizar acciones que podrían ser maliciosas		x						
Generar correos electrónicos de phishing para evitar los filtros de spam	x	x				x		
Uso de LLM como complementos para actuar como proxy.					x			
Generar sitio web de phishing a través de ChatGPT				x				
Generación de código e inyección de DLL		x						
Recopilación de datos de víctimas para desarrollar un correo electrónico de ataque			x		x			
Elaboración de scripts maliciosos				x	x		x	
Enlace de spear phishing								x
Ofuscación de código, modificación del formato de archivo			x					
Adivinación de contraseñas mediante LLM	x							
Suplantación de identidad con fines de phishing				x				
Generando contenido para desinformar.	x							

Tabla 7: Resumen modelos de LLM's por usos. Fuente autor

En este caso también destaca de forma evidente los LLM's de la familia GPT, todos los modelos GPT obtienen resultados óptimos en tres casos de uso.

La Tabla 8 se ha confeccionado con la información que se ha extraído del artículo [229]. En la tabla se ve cada modelo LLM, con su arquitectura específica, los datos de entrenamiento que se utilizaron, las aplicaciones generales del modelo y los casos de uso de ciberdefensa para los que tienen aplicación práctica, junto con su fuente original.

Modelo	Arquitectura	Pre-entrenamiento	Aplicaciones	Casos de uso de ciberseguridad	Técnicas de entrenamiento	Fuente
GPT-3	decodificador	Libros, web, Wikipedia, Common Crawl	Modelado de lenguaje, creación de textos y QA	Detección de malware, inteligencia de amenazas, detección de ataques de ingeniería social	Entrenamiento autorregresivo, Pérdida de entropía cruzada escalada, Retropropagación y descenso de gradiente, Entrenamiento de precisión mixto.	[247]

GPT-4	decodificador	Datos web, datos de terceras partes	Modelado de lenguaje, creación de textos y QA	Detección de malware, inteligencia de amenazas, detección de ataques de ingeniería social	Entrenamiento autorregresivo	[248]
T5	Encoder-decoder	C4, Web Text, Wikipedia	Modelado de lenguaje, resúmenes y traducción	Detección de malware, inteligencia de amenazas, detección de ingeniería social	Marco de texto a texto, preentrenamiento basado en denotación	[249]
BERT	codificador	BooksCorpus, Wikipedia en inglés	modelado de lenguaje, Clasificación, Control de calidad, NER	Detección de malware, inteligencia de amenazas, detección de intrusiones, detección de phishing	LM enmascarado (MLM), Predicción de la siguiente oración (NSP)	[250]
ALBERT	Codificador	BooksCorpus, Wikipedia en inglés	modelado de lenguaje, Clasificación	Detección de malware, inteligencia de amenazas, detección de intrusiones, detección de phishing	Parametrización de incrustación factorizada, intercambio de parámetros entre capas, pérdida de coherencia entre oraciones, predicción del orden de las oraciones (SOP)	[251]
RoBERTa	Codificador	BooksCorpus, Wikipedia en inglés	Modelado de lenguaje, Clasificación, Control de calidad, NER	Detección de malware, inteligencia de amenazas, detección de intrusiones, detección de phishing	Enmascaramiento dinámico, Frases completas sin pérdida de NSP, minilots grandes, BPE de nivel de bytes más grande	[252]
XLNet	Codificador	Wikipedia en ingles	Modelado de lenguaje, clasificación y QA	Detección de malware, inteligencia de amenazas, detección de intrusiones, detección de phishing	Permutación LM (PLM), autoatención de dos corrientes, recurrencia de segmentos y codificación relativa	[253]
ProphetNet	Codificador-decodificador	Datos web, Libros	Modelado de lenguaje, generación de preguntas, Resumen	Informes de ciberseguridad, inteligencia de amenazas	Generación de secuencia enmascarada, Entrenamiento autorregresivo, objetivo de eliminación de ruido del codificador automático, parámetros compartidos entre codificador y decodificador, Estimación de máxima verosimilitud (MLE)	[254]
Falcon	Decodificador	Datos web	Modelado de lenguaje, finalización de texto, control de calidad	Detección de malware, inteligencia de amenazas,	Entrenamiento autorregresivo, FlashAttention, codificación posicional ALiBi	[255]

				detección de ingeniería social		
Reformer	Codificador-decodificador	Datos web	Modelado de lenguaje, clasificación.	Detección de malware, inteligencia de amenazas, detección de ingeniería social	Atención de hash sensible a la localidad (LSH), procesamiento fragmentado, cabezales de atención QK compartidos, capas reversibles	[256]
PaLM	Decodificador	Páginas web, libros, Wikipedia, artículos de noticias, código fuente, conversaciones en redes sociales, GitHub	Modelado de lenguaje, QA, traducción.	Inteligencia de amenazas, Generación de Políticas de Seguridad	Activación SwiGLU, capas paralelas, atención de consultas múltiples (MQA), incrustaciones de RoPE, incrustaciones de entrada y salida compartidas	[257], [258]
PaLM2	Decodificador	Documentos web, libros, código, matemáticas, datos conversacionales	Modelado de lenguaje, QA, resúmenes	Inteligencia de amenazas, Generación de Políticas de Seguridad	Calcular el escalado óptimo, secuencias de tokens Canary, tokens de control para inferencia	[258]
LLaMA	Decodificador	Combinación de datos disponibles públicamente	Modelado de lenguaje, finalización de texto, control de calidad	Inteligencia de amenazas, detección de malware	Pre-normalización, función de activación SwiGLU, incrustación rotativa, paralelismo de modelo y secuencia	[259]
LLaMA2	Decodificador	Datos web	Modelado de lenguaje, finalización de texto, control de calidad	Inteligencia de amenazas, detección de malware	Entrenamiento autorregresivo optimizado, Atención de consultas agrupadas (GQA)	[260]
GShard	MoE	Datos web	Modelado de lenguaje	Inteligencia de amenazas, detección de intrusiones, detección de malware	Computación condicional, API de anotación ligera, partición XLA SPMD, MoE por posición	[261]
ELECTRA	Codificador	BooksCorpus, Wikipedia en inglés	Modelado de lenguaje, clasificación	Detección de malware, inteligencia de amenazas, detección de intrusiones, detección de phishing	Detección de token reemplazada, marco generador-discriminador, reemplazo de token, peso compartido	[262]
Mistral 7B	Decodificador	Conjuntos de datos Phi-3 (documentos públicos, sintéticos, formatos de chat)	Modelado de lenguaje, finalización de texto, control de calidad	Inteligencia de amenazas, detección de intrusiones, detección de malware	Atención de ventana deslizante, atención de consultas (GQA), tokenizador BPE de reserva de bytes	[263]
Zephyr 7B Beta	Decodificador	Datos web	Modelado de lenguaje, finalización de texto, control de calidad	Inteligencia de amenazas, detección de intrusiones, detección de malware	Atención Flash, optimización de preferencias directas (DPO)	[264]

Falcon2 11B	Decodificador	RefinedWeb mejorado con corpus seleccionados	Modelado de lenguaje, finalizaci3n de texto, control de calidad	Inteligencia de amenazas, detecci3n de intrusiones, detecci3n de malware	ZeRO, n3cleos Triton de alto rendimiento, FlashAtenci3n-2	[265]
-------------	---------------	---	--	---	---	-------

Tabla 8: Comparativa LLM'S. Fuente [268]

De la Tabla 8 se puede extraer los LLM's m3s indicados para los distintos casos de uso. Estos son:

- Detecci3n de *malware*: Los LLM's que destacan en la detecci3n de malware son los modelos GPT-3, GPT-4, T5, BERT, ALBERT, RoBERTa, XLNet, Falcon, Reformer, LLaMa, LLaMA2, GShard, ELECTRA, Mistral 7B, Zephyr 7B y Falcon2 11B.
- Inteligencia de amenazas: se pueden destacar GPT-3, GPT-4, T5, BERT, ALBERT, RoBERTa, XLNet, ProphetNet, Falcon, Reformer, PaLM, PaLM2, LLaMa, LLaMA2, GShard, ELECTRA, Mistral 7B, Zephyr 7B y Falcon2 11B.

El n3mero se reduce bastante en lo referente a modelos con buenos resultados en la detecci3n de ataques de ingenier3a social, entre los que se pueden destacar GPT-3, GPT-4, T5, Falcon, Reformer

Para la detecci3n de *phishing* se puede destacar XLNet y ELECTRA y algunos modelos o variantes de la familia BERT como son BERT, ALBERT y RoBERTa.

De los modelos analizados, los modelos PaLM y PaLM2 destacan para la generaci3n de Pol3ticas de Seguridad y tambi3n es relevante que el modelo ProphetNet es el 3nico destacado en el caso de uso de generaci3n de informes de seguridad, algo muy apreciado por los componentes de los equipos de seguridad.

Ning3n LLM cumple con todos los casos de uso a la vez, pero se podr3an destacar por cumplimiento en cuatro casos de uso, como son, la detecci3n de malware, inteligencia de amenazas, detecci3n de intrusiones, y detecci3n de phishing los modelos BERT, ALBERT, RoBERTa, XLNet y ELECTRA. Si los casos de uso son detecci3n de malware, inteligencia de amenazas, detecci3n de ataques de ingenier3a social se pueden destacar GPT-3, GPT-4, T5, Falcon y Reformer.

Con objeto de dar un poco m3s de claridad a la informaci3n se ha generado la Tabla 9, que agrupa los casos de uso por cada uno de los modelos.

En la segunda columna denominada "Arquitectura" figura para cada modelo si hace uso solo del codificador, solo del decodificador, o de ambos codificador-decodificador, y por ultimo tambi3n aparece la arquitectura MoE que consiste en una mezcla de expertos.

Solo Codificador:

- BERT, ALBERT, RoBERTa, XLNet y ELECTRA.

Solo Decodificador.

- GPT-3, GPT-4, Falcon, PaLM, PaLM2, LLaMA, LLaMA2, Mistral 7B, Zephyr 7B y Falcom2 11B

Codificador-decodificador:

- T5, PropetNet y Reformer.

Arquitectura MoE, mezcla de expertos:

- GShard.

Modelo	Arquitectura	Casos de uso						
		Generación de políticas de seguridad	Informes de seguridad	Detección de malware	Inteligencia de amenazas	Detección de ataques de ingeniería social	Detección de intrusiones	Detección de phishing
GPT-3	Decodificador			X	X	X		
GPT-4	Decodificador			X	X	X		
T5	Codificador-decodificador			X	X	X		
BERT	Codificador			X	X		X	X
ALBERT	Codificador			X	X		X	X
RoBERTa	Codificador			X	X		X	X
XLNet	Codificador			X	X		X	X
ProphetNet	Codificador-decodificador		X		X			
Falcon	Decodificador			X	X	X		
Reformer	Codificador-decodificador			X	X	X		
PaLM	Decodificador	X			X			
PaLM2	Decodificador	X			X			
LLaMA	Decodificador			X	X			
LLaMA2	Decodificador			X	X			
GShard	MoE			X	X		X	
ELECTRA	Codificador			X	X		X	X
Mistral 7B	Decodificador			X	X		X	
Zephyr 7B	Decodificador			X	X		X	
Falcon2 11B	Decodificador			X	X		X	

Tabla 9: Comparativas casos de uso agrupados por modelo. Fuente [229].

4.4 Las principales vulnerabilidades de los LLM's

También es interesante saber cuáles son los puntos débiles de los LLM's. Según OWASP, un proyecto de código abierto facilitado por una organización internacional sin ánimo de lucro que se dedica a la seguridad en aplicaciones web, podemos extraer la lista de las 10 vulnerabilidades más importantes detectadas para los LLM's [269].

1. Inyección de solicitudes [270]: los atacantes manipulan las solicitudes a un LLM con el fin de obtener información o de escalar privilegios dentro de las máquinas. Puede ser directa (*jailbreaking*), con un ataque activo, o indirecta, contaminando sitios externos como una web

que se utiliza como fuente para el LLM. Se deben aplicar políticas de control de acceso para evitar este riesgo.

2. Gestión de salida insegura [271]: Se produce al no examinar de forma constante los resultados del LLM, dando lugar a ataques *Cross-site scripting* (XSS), falsificación de solicitud entre sitios (CSRF), falsificación de solicitud del lado del servidor (SSRF) y ejecución remota de código (RCE). Se evita aplicando un modelo de seguridad Zero Trust.
3. Envenenamiento de datos de formación [272]: se produce al manipular los datos que sirven para el entrenamiento de un modelo, el objetivo es hacer que el modelo tenga el comportamiento o tome decisiones en función de los datos envenenados que se han introducido. Se debe proteger la cadena de suministro de los datos y comprobar la legitimidad de las fuentes.
4. Modelo de denegación de servicio [273]: es un ataque similar a la denegación de servicio distribuido muy común en servidores web (DDoS) se trata de intentar bloquear o degradar la respuesta del LLM con operaciones que consumen muchos recursos. Se puede prevenir aplicando límites de velocidad de API para usuarios o para direcciones IP específicas.
5. Vulnerabilidades de la cadena de suministro [273]: expone al modelo a riesgos de seguridad o producir resultados inexactos. Se debe securizar la cadena de suministro y mantener un inventario de componentes y datos actualizados.
6. Divulgación de información confidencial [274]: se puede revelar de forma inadvertida o inintencionada información que debe ser objeto de reserva como datos personales, datos empresariales, datos médicos, etc. Se debe limitar o eliminar este tipo de información en los datos, en todo caso se deben securizar los datos que estén a disposición del modelo.
7. Diseño de *plugin* no seguro [275]: hacer uso de *plugins* de terceros que no cuenten con todas las garantías puede ser fuente de varias vulnerabilidades. Se debe utilizar complementos seguros y aplicar controles de autenticación. Comprobar la necesidad de conocer.
8. Agencia excesiva [276]: se produce como respuesta automática en respuesta a mensajes, que previamente han sido configuradas por los desarrollares, quizás para dotar al modelo de mayor versatilidad, pero esto puede producir resultados inesperados o indeseados. Se debe limitar la funcionalidad de los modelos, los permisos y la autonomía de los complementos, así como cualquier otra herramienta, se deben configurar con el principio de mínimo privilegio.
9. Dependencia excesiva [277]: ocasionalmente se pueden producir resultados no deseados por la IA que se han dado en denominar “alucinaciones de la IA”. Para evitar la excesiva dependencia de la IA y prevenir los posibles efectos de resultados con “alucinaciones de la IA” se deben implementar políticas de supervisión de los LLM’s.
10. Robo de modelos [278]: Se trata del robo, plagio o intento de apropiación de LLM’s propietarios desarrollados por empresas o particulares. Se deben aplicar controles de acceso basados en roles, se deben supervisar los *logs* y los registros de acceso y aplicar funciones de prevención de pérdida de datos.

De igual forma, se indican de manera genérica algunas acciones que se deben poner en marcha para proteger los LLM’s de las vulnerabilidades más críticas ([279], [280], [281]):

- Análisis de tráfico de red para detectar patrones que puedan dar indicios de manipulación, intervención o vulneración de un LLM.
- Visibilidad en tiempo real de paquetes y datos que interactúan con los LLM’s a nivel de bits.
- Aplicar DLP para proteger los datos en tránsito, especialmente datos sensibles o que deban ser objeto de la debida protección (datos personales, datos legales, etc.).

- Verificar, filtrar y aislar el tráfico para proteger las aplicaciones de los LLM's en riesgo.
- Emplear el aislamiento remoto del navegador (RBI) para aislar a los usuarios de los modelos con código malicioso inyectado.
- Utilizar conjuntos de reglas administradas por un WAF para bloquear los ataques LLM basados en inyección de código SQL, XSS y otros vectores de ataque web.
- Emplear un modelo de seguridad Zero Trust para reducir su superficie de ataque al conceder solo acceso basado en el contexto y con privilegios mínimos por recurso.

En el [Anexo III: Listado cve que afectan a los llm's](#), se ha recopilado la información de los 54 CVE que afectan a la fecha de este trabajo a los LLM's.

4.5 Recomendaciones sobre posibles implementaciones.

Desarrollar un LLM con medios propios, puede requerir unos recursos técnicos y económicos que solo están al alcance de las grandes empresas tecnológicas. Según [282] el desarrollo de GPT-4 costó más de 100 millones de dólares, pero también hay que tener en cuenta muy especialmente en Defensa que la información puede tener un valor estratégico y compartir o perder la trazabilidad de los datos en algún momento del proceso puede tener consecuencias muy negativas.

Ajuste fino, es una opción igualmente válida, se trata de utilizar un modelo de código abierto que se adapte a nuestras necesidades y realizar el ajuste fino para el caso o los casos de uso que nos interese desarrollar, existe un proyecto explicativo muy interesante sobre el ajuste fino de modelos LLM que es accesible públicamente en GitHub denominado HackMentor⁵ [283].

Se considera muy recomendable hacer pruebas con LLM's específicos de ciberseguridad, como es el caso de: CySecBERT, del que podemos encontrar información y recursos en [284], [285], [286] y [287]; en el caso de CyBERT encontramos dicha información en [288] y [289]; para SecureBERT existe información muy instructiva en [290], [291] y [292]; y para el análisis de vulnerabilidades, PentestGPT, podemos obtener información o realizar una prueba en [293], [294] y [295].

Como recomendación extraída a lo largo de la confección del trabajo se considera importante poder realizar pruebas con LLM's para familiarizarse con el uso, administración y características básicas de algunos modelos, esto se puede hacer con LM Studio⁶ y también con el uso de Hugging Face⁷.

Como se ha mencionado anteriormente es fundamental tener conocimientos de Python para poder desenvolverse con el uso de Keras⁸, Tensorflow⁹ y Pytorch¹⁰.

El salto de calidad que puede proporcionar una buena implementación de la IA a través del uso de los LLM's en los casos de uso especificados para el ámbito de la Defensa, puede ser importantísimo y absolutamente imprescindible, ya que, como se ha visto los mismos desarrollos y ajuste fino que se pueden realizar para hacer frente a las amenazas críticas que afectan a nuestros sistemas, también son aprovechadas para realizar desarrollos y ajuste fino para casos ofensivos y por tanto contrapuestos a nuestros intereses. Es evidente que la implementación inmediata solo podría llevarse a cabo recurriendo a una solución o proyecto llave en mano, pero se considera fundamental poder avanzar en el

⁵ <https://github.com/tmylla/HackMentor>

⁶ <https://lmstudio.ai/>

⁷ <https://huggingface.co/>

⁸ <https://keras.io/>

⁹ <https://www.tensorflow.org/?hl=es>

¹⁰ <https://pytorch.org/>

conocimiento, concienciación y formación del personal de Defensa para poder en un futuro cercano, desarrollar y gestionar soluciones propias.

5 CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

En el primer capítulo se habían marcado cuatro puntos como los “objetivos” para este trabajo, el análisis de estos objetivos, va a servir para hacer una recapitulación de lo mas relevante y servirá para extraer las conclusiones finales a este trabajo.

Los objetivos marcados al inicio de este trabajo fueron:

- PI 1. ¿Cuál ha sido el desarrollo de la IA y su contribución al desarrollo de los LLM’s?
- PI 2. ¿Cual ha sido la evolucón de las redes neuronales artificiales y su contribución al desarrollo delos LLM’s?
- PI.3. ¿Como se ha producido la explosión de uso de los LLM’s en los ultimos años?
- PI.4. ¿Qué nos pueden aportar los LLM’s en el ámbito de la Defensa, casos de uso y posible implementación a corto-medio plazo?

¿Cuál ha sido el desarrollo de la IA y su contribución al desarrollo de los LLM’s?

Aunque la explosión de uso de los LLM’s es relativamente reciente ya existen suficiente literatura al respecto como se ha podido comprobar. Se destacan [227], [228], [229], [230] y [231], las cuales ofrecen una visión suficientemente completa del estado del arte.

También se han visto estudios más específicos como en [213], [221], [223], [224], [226], [296] y [297] que hacen análisis de herramientas muy eficaces y que pueden ser usadas en un marco como es el propuesto en este estudio.

Pero también hay que destacar el estudio realizado en [197] en el que, tras diversas pruebas realizadas por el Departamento de Defensa de Estados Unidos, se sigue considerando a día de hoy, que no se pueden dejar las decisiones de mando o la dirección de las operaciones en manos de los LLM’s, ya que, es frecuente la aparición indeseable de alucinaciones y la toma de decisiones que conducen a la propia autodestrucción. Sin embargo, en [197] sí se muestra la eficacia y la gran ayuda que ofrecen en la realización de resúmenes de operaciones, en la confección de inteligencia y en general en el apoyo al mando.

Se puede considerar que los LLM’s son el máximo exponente de la IA y su uso ha pasado a ser general entre la población.

¿Cual ha sido la evolucón de las redes neuronales artificiales y su contribución al desarrollo de los LLM’s?

El desarrollo de los LLM’s ha sido paralelo al de las redes neuronales, aunque ya se daban desarrollos importantes con las RNN’s y sus variantes GRU y LSTM, y también con las CNN’s. Sin embargo, no se ha producido un verdadero despegue de la IAG hasta la aparición de los transformadores, cuya arquitectura ha revolucionado el *deep learning* y ha posibilitado la explosión de uso vivida desde el año 2022 con los *chatbots* como ChatGPT. Inicialmente los casos de uso de los LLM’s eran más limitados, pero actualmente ya tienen aplicación en todos los ámbitos de nuestra vida, y aunque su desarrollo y entrenamiento solo está al alcance unas pocas grandes empresas a nivel mundial, su conocimiento y su uso se ha generalizado y como no puede ser de otra forma, deben ser tenidos en consideración desde la óptica o del punto de vista de la Defensa

¿Como se ha producido la explosión de uso de los LLM’s en los ultimos años?

La explosión de la IA ha sido en datos la mayor de la historia, ChatGpt batió el record de tiempo en llegar a un millon de usuarios. Mas que la explosión en terminos numericos, se puede considerar que lo

mas destacado es la popularización de la IA y de los LLM's que a día de hoy son utilizados de forma masiva por la población, en trabajos de investigación, academicos, para traducciones, en apoyo a la codificación y pruebas y en muchos otros ambitos de la vida. Desde nuestros moviles, pulseras de actividad, vehiculos, hasta nuestros hogares, ya hacen uso de la IA, los LLM's ya estan presentes en nuestros diagnosticos medicos, en las recomendaciones de compras, o en predicción de cosechas, pero quiza los mas importante es la velocidad con la que avanzan y la independencia del ser humano que se está manifestando en su entrenamiento y ajuste fino, hasta el punto de que podrían ser totalmente independientes, algo que tambien puede ser percibido mas como un peligro que como una ventaja.

¿Qué nos pueden aportar los LLM's en el ámbito de la Defensa, casos de uso y posible implementación a corto-medio plazo?

En este trabajo se han tomado 11 posibles casos de uso en Defensa, alguno quizá más ofensivo como es el caso de los *honeypots*, pero que se considera perfectamente factibles con los desarrollos existentes actuales.

Las tablas extraídas de las revisiones realizadas por otros autores muestran resultados muy satisfactorios alcanzados con modelos que en general están basado en la arquitectura de transformadores. Si finalmente se decide implementar un LLM o un ecosistema de varios LLM's habrá que hacer un estudio más pormenorizado de los pros y contras de utilizar modelos comerciales, de solicitar ofertas para el desarrollo de un modelo u optar por un desarrollo propio con medios internos, estas tres opciones tienen implicaciones importantes desde el punto de vista económico, desde el punto de vista reputacional y también desde la óptica de la dependencia de actores externos. En el MINISDEF la seguridad y la protección de la información es vital y por tanto es necesario estudiar todas las alternativas y tener en cuenta que estos datos no se pueden revelar o ceder a terceras partes, cualquier desarrollo o implementación que se haga ha de tener en cuenta la seguridad de la información.

5.2 Líneas futuras.

La implantación de los 11 posibles casos de uso tendría que pasar necesariamente por la elección de los LLM's que se van a aplicar en cada caso. De las tablas vistas en el trabajo, se pueden extraer modelos testados con resultados óptimos para cada uno de ellos.

Como se ha visto a lo largo del trabajo, estos desarrollos están muy ligados al ecosistema Python, la IA habla Python, esto que en el mundo universitario es algo muy habitual, en el entorno de Defensa es prácticamente inexistente, por lo que, cualquier desarrollo o intento de aplicación, va a tener como requisito previo la formación del personal de Defensa y tener en consideración todo el ciclo de vida del modelo tanto para costes, como para mantenimiento y actualizaciones.

Además, la explosión de los LLM's ha tenido lugar en apenas tres años, y los avances en este tiempo han sido enormes, quedarse fuera de esta corriente significaría prácticamente condenar nuestros sistemas a la inseguridad total.

Se puede considerar que a corto plazo la opción más viable sería comenzar por el uso de LLM's existentes y entrenados, para familiarizarse con su manejo y formar al personal que va a tener que hacer frente a estas amenazas, así mismo inicialmente solo se debería aplicar en ámbitos de información no clasificada como la I3D o redes de uso público. En el medio o largo plazo y vistos los avances y posibilidades que pueden aportar los LLM's sería conveniente desarrollar y entrar LLM's específicos en el ámbito de la Defensa, en este caso, ya se podrían implementar en sistemas clasificados, sistemas aislados y redes con información sensible del MINISDEF.

Albert Einstein dijo *“No sé con qué armas se librará la Tercera Guerra Mundial, pero la Cuarta Guerra Mundial se librará con palos y piedras”*. Es posible que los avances en IA sean percibidos como una revolución de las máquinas y la caída o deterioro de las capacidades de los seres humanos, parece

que estamos condenados a ser sustituidos por máquinas que hacen el trabajo mejor y más rápido que nosotros, también se puede percibir como una pérdida de capacidades intelectuales y de la intuición y poder de decisión que siempre nos ha caracterizado como especie, pero hasta que ese momento llegue, es necesario hacer frente a los retos y los desafíos, e indudablemente uno de ellos es conocer, manejar y aprovechar todas las ventajas y beneficios que nos aporta la tecnología. Los LLM's son una realidad que debemos aprovechar, nuestra Defensa va a depender en gran medida del conocimiento que tengamos de ellos y del buen uso que hagamos de todas las posibilidades que nos ofrecen.

6 BIBLIOGRAFÍA

- [1] «The Turing Test (Stanford Encyclopedia of Philosophy)». Accedido: 31 de diciembre de 2024. [En línea]. Disponible en: <https://plato.stanford.edu/entries/turing-test/>
- [2] M. Solutions, «El auge de los large language models: de los fundamentos a la aplicación», 2024, Accedido: 31 de diciembre de 2024. [En línea]. Disponible en: www.managementsolutions.com
- [3] «El chatbot Gemini de Google explota ante un usuario: “Eres una pérdida de tiempo y recursos. Por favor, muere”». Accedido: 20 de noviembre de 2024. [En línea]. Disponible en: <https://www.20minutos.es/tecnologia/inteligencia-artificial/gemini-chatbot-google-explota-usuario-por-favor-muere-5655125/>
- [4] «¿Qué son los grandes modelos de lenguaje (LLM)? | IBM». Accedido: 3 de octubre de 2024. [En línea]. Disponible en: <https://www.ibm.com/mx-es/topics/large-language-models>
- [5] «Inteligencia artificial generativa y LLM: ¿amenaza u oportunidad?» Accedido: 8 de enero de 2025. [En línea]. Disponible en: https://www.redseguridad.com/especialidades-tic/tecnologias-disruptivas/inteligencia-artificial-generativa-amenaza-u-oportunidad_20241010.html
- [6] «CHAT GPT, Inteligencia artificial en el sector de la comunicación - 3AW». Accedido: 1 de noviembre de 2024. [En línea]. Disponible en: <https://3aw.es/chat-gpt-inteligencia-artificial-en-el-sector-de-la-comunicacion>
- [7] «El año en que ChatGPT nos hizo vivir peligrosamente: su primer aniversario deja más preguntas que respuestas». Accedido: 1 de noviembre de 2024. [En línea]. Disponible en: <https://www.xataka.com/robotica-e-ia/ano-que-chatgpt-vivio-peligrosamente>
- [8] «El uso de LLM en China alcanza los 600 millones de usuarios en medio de una adopción acelerada de GenAI | Cryptopolitan en Binance Square». Accedido: 1 de noviembre de 2024. [En línea]. Disponible en: <https://www.binance.com/es-LA/square/post/12128656747417>
- [9] «ChatGPT bate el récord de crecimiento más rápido de usuarios en la historia». Accedido: 1 de noviembre de 2024. [En línea]. Disponible en: <https://forbes.com.mx/chatgpt-bate-el-record-de-crecimiento-mas-rapido-de-usuarios-en-la-historia/>
- [10] «2023: el año de la Inteligencia Artificial generativa | ITDigitalSecurity | IT Digital Security». Accedido: 31 de diciembre de 2024. [En línea]. Disponible en: <https://www.itdigitalsecurity.es/itdigitalsecurity/2023/12/2023-el-ano-de-la-inteligencia-artificial-generativa>
- [11] «El auge de ChatGPT: 100 millones de usuarios en dos meses, ¿aliado o amenaza para la humanidad? - Semana». Accedido: 31 de diciembre de 2024. [En línea]. Disponible en: <https://www.semana.com/educacion/articulo/el-auge-de-chatgpt-100-millones-de-usuarios-en-dos-meses-aliado-o-amenaza-para-la-humanidad/202425/>
- [12] «ChatGPT on track to surpass 100 million users faster than TikTok or Instagram: UBS». Accedido: 2 de noviembre de 2024. [En línea]. Disponible en: <https://finance.yahoo.com/news/chatgpt-on-track-to-surpass-100-million-users-faster-than-tiktok-or-instagram-ubs-214423357.html>
- [13] «ChatGPT: usuarios y estadísticas (2024)». Accedido: 2 de noviembre de 2024. [En línea]. Disponible en: <https://www.primeweb.com.mx/chatgpt-usuarios-estadisticas>
- [14] «Chart: Threads Shoots Past One Million User Mark at Lightning Speed | Statista». Accedido: 2 de noviembre de 2024. [En línea]. Disponible en: <https://www.statista.com/chart/29174/time-to-one-million-users/>

- [15] «The-rise-of-generative-ai-large-language-models-llms-like-chatgpt», <https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt/>. Accedido: 1 de noviembre de 2024. [En línea]. Disponible en: <https://informationisbeautiful.net/visualizations/the-rise-of-generative-ai-large-language-models-llms-like-chatgpt/>
- [16] «IIB LLMs public (new, Oct 2024) - Hojas de cálculo de Google». Accedido: 5 de noviembre de 2024. [En línea]. Disponible en: https://docs.google.com/spreadsheets/d/14KzlcYOK6Xj-_6N19T2Iqro4sGaAUOLD2FovSXYefUQ/edit?gid=426115144#gid=426115144
- [17] R. Benítez, G. Escudero, y S. Kanaan, «Inteligencia artificial avanzada».
- [18] M. Negnevitsky, *Artificial Intelligence: A Guide to Intelligent Systems, Second Edition*, vol. 123. 2005.
- [19] J. A. Alonso Jiménez, «Manual de Lisp», 1991.
- [20] J. McCarthy, M. L. Minsky, N. Rochester, y C. E. Shannon, «A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955», *AI Mag*, vol. 27, n.º 4, pp. 12-12, dic. 2006, doi: 10.1609/AIMAG.V27I4.1904.
- [21] C. J. Bashe, W. Buchholz, G. V. Hawkins, J. J. Ingram, y N. Rochester, «ARCHITECTURE OF IBM's EARLY COMPUTERS.», *IBM J Res Dev*, vol. 25, n.º 5, pp. 363-375, 1981, doi: 10.1147/RD.255.0363.
- [22] «Claude Shannon es el fundador de la teoría de la información. Quién es Claude Shannon y por qué es famoso. Teoría de la comunicación en sistemas secretos». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://severnyetravy.ru/es/klod-shennon-osnovatel-teorii-informacii-kto-takoi-klod-shennon-i-chem-on/>
- [23] A. L. Samuel, «Some Studies in Machine Learning Using the Game of Checkers».
- [24] «Una aproximación biográfica e histórica a la inteligencia artificial on jstor». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.jstor.org/stable/27135844>
- [25] H. A. Simon, «Revista Latinoamericana de Psicología», vol. 33, n.º 2, pp. 223-224, 1916, Accedido: 1 de enero de 2025. [En línea]. Disponible en: <http://www.redalyc.org/articulo.oa?id=80533209>
- [26] A. Newell y H. A. Simon, «The logic theory machine a complex information processing system», *IRE Transactions on Information Theory*, vol. 2, n.º 3, pp. 61-79, 1956, doi: 10.1109/TIT.1956.1056797.
- [27] L. Gugerty, «Newell and Simon's Logic Theorist: Historical Background and Impact on Cognitive Modeling», <http://dx.doi.org/10.1177/154193120605000904>, pp. 880-884, oct. 2006, doi: 10.1177/154193120605000904.
- [28] «(PDF) El modelo neuronal de mcculloch y pitts Interpretación Comparativa del Modelo». Accedido: 1 de enero de 2025. [En línea]. Disponible en: https://www.researchgate.net/publication/343141076_EL_MODELO_NEURONAL_DE_McCULLOCH_Y_PITTS_Interpretacion_Comparativa_del_Modelo
- [29] S. Chakraverty, D. M. Sahoo, y N. R. Mahato, «McCulloch–Pitts Neural Network Model», *Concepts of Soft Computing*, pp. 167-173, 2019, doi: 10.1007/978-981-13-7430-2_11.
- [30] «Redes neuronales artificiales - OpenCourseWare.2010».
- [31] A. M. Turing, «COMPUTING MACHINERY AND INTELLIGENCE», *Computing Machinery and Intelligence. Mind*, vol. 49, pp. 433-460, 1950.

- [32] A. M. Turing, «Computing Machinery and Intelligence», *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*, pp. 23-65, 2009, doi: 10.1007/978-1-4020-6710-5_3.
- [33] «View of Un genio adelantado a su época CHARLES BABBAGE». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://revista-dyna.com/index.php/DYNA/article/view/2626/2498>
- [34] R. Da *et al.*, «Los teoremas de incompletitud de Gödel, teoría de conjuntos y el programa de David Hilbert», *EPISTEME*, vol. 34, n.º 1, pp. 19-40, 2014, Accedido: 1 de enero de 2025. [En línea]. Disponible en: http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0798-43242014000100002&lng=es&nrm=iso&tlng=es
- [35] A. S. Ramírez, «Maquina universal de turing: algunas indicaciones para su construcción», *Revista Universidad EAFIT*, vol. 33, n.º 108, pp. 61-106, jul. 2012, Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://publicaciones.eafit.edu.co/index.php/revista-universidad-eafit/article/view/1142>
- [36] S. Rajsbaum y E. Morales, «6 ciencia • enero-marzo de 2016».
- [37] U. César Vallejo Perú Barrera Arrestegui, «UCV-HACER. Revista de Investigación y Cultura», *Revista de Investigación y Cultura*, vol. 1, n.º 1, 2012, Accedido: 1 de enero de 2025. [En línea]. Disponible en: <http://www.redalyc.org/articulo.oa?id=521752338014>
- [38] R. Martínez y A. García-Beltrán, «BREVE HISTORIA DE LA INFORMÁTICA».
- [39] J. Von Neumann, J. B. Fontelles, y C. Mayeur, «El ordenador y el cerebro», 1999, Accedido: 1 de enero de 2025. [En línea]. Disponible en: https://books.google.com/books/about/El_ordenador_y_el_cerebro.html?hl=es&id=huATJ-RWbWAC
- [40] «The Johnniac - CHM». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://computerhistory.org/events/johnniac/>
- [41] «JOHNNIAC - CHM Revolution». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.computerhistory.org/revolution/birth-of-the-computer/4/94>
- [42] F. J. Gruenberger, «The History of the JOHNNIAC», 1968, Accedido: 1 de enero de 2025. [En línea]. Disponible en: https://www.rand.org/pubs/research_memoranda/RM5654.html
- [43] «1954, IBM 704. Primera generación de ordenadores – La pasión de pensar». Accedido: 1 de enero de 2025. [En línea]. Disponible en: https://lapasiondepensar.wordpress.com/2018/09/07/1954-ibm_704/
- [44] «Lenguaje de programación más antiguo, un repaso histórico». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://blogs.uoc.edu/informatica/es/el-lenguaje-de-programacion-mas-antiguo/>
- [45] «¿Cuál fue el primer robot real de la historia? | Endesa». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.endesa.com/es/la-cara-e/historia/primer-robot-real-historia>
- [46] «¿Qué es el procesamiento de lenguaje natural (NLP)? | Una guía completa del NLP | Elastic». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.elastic.co/es/what-is/natural-language-processing>
- [47] «Historia de la inteligencia artificial - Iberdrola». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.iberdrola.com/innovacion/historia-inteligencia-artificial>
- [48] «Traducción automática: el informe ALPAC». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://blog.pangeanic.com/es/traduccion-automatica-informe-alpac>

- [49] «Recesión de la inteligencia artificial». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://keepcoding.io/blog/recesion-de-la-inteligencia-artificial/>
- [50] «Invierno De La IA: El Informe Lighthill | IAresponde». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://iaresponde.com/invierno-de-la-ia/>
- [51] «Los inviernos de la IA: cuando los abuelos de ChatGPT fracasaron». Accedido: 4 de noviembre de 2024. [En línea]. Disponible en: <https://hipertextual.com/2023/06/inviernos-inteligencia-artificial>
- [52] «Los inviernos de la IA: cuando los abuelos de ChatGPT fracasaron». Accedido: 20 de noviembre de 2024. [En línea]. Disponible en: <https://hipertextual.com/2023/06/inviernos-inteligencia-artificial>
- [53] «The Rise & Fall of LISP - Too Good For The Rest Of the World : r/lisp». Accedido: 30 de noviembre de 2024. [En línea]. Disponible en: https://www.reddit.com/r/lisp/comments/11ttnxk/the_rise_fall_of_lisp_too_good_for_the_rest_of/?tl=es-es
- [54] «Japón se rinde ante EE UU en la guerra de los ordenadores “inteligentes” | Sociedad | EL PAÍS». Accedido: 30 de noviembre de 2024. [En línea]. Disponible en: https://elpais.com/diario/1992/06/03/sociedad/707522405_850215.html
- [55] «Introducción a los Modelos de Lenguaje de Gran Tamaño (LLM)». Accedido: 30 de noviembre de 2024. [En línea]. Disponible en: <https://www.pwc.es/es/newlaw-pulse/legaltech/introduccion-modelos-lenguaje-gran-tamano.html>
- [56] «Así era ELIZA, el primer bot conversacional de la historia». Accedido: 30 de noviembre de 2024. [En línea]. Disponible en: <https://www.xataka.com/historia-tecnologica/asi-era-eliza-el-primer-bot-conversacional-de-la-historia>
- [57] «Descubre a ELIZA, el ChatGPT terapeuta de los años 60», <https://www.fundacionmariapaulalonso.org/eliza-chatbots-aprendizaje-evolucion-historia/>.
- [58] «La Historia de los Chatbots - Automaticchat», <https://automaticchat.com/blog/la-historia-de-los-chatbots>.
- [59] E. León, «Asistentes virtuales ¿Conoces a Alexa y sus amigas?», <https://www.baoss.es/asistentes-virtuales-conoces-a-alexa-y-sus-amigas/>.
- [60] «25 aniversario de Computer Hoy: historia de los asistentes virtuales como Alexa, Siri y Google, y el principio del fin ante el auge de la IA tipo ChatGPT», <https://computerhoy.20minutos.es/android/historia-siri-alexa-otros-asistentes-futuro-chatgpt-1237612>.
- [61] «Tay, la robot racista y xenófoba de Microsoft - BBC News Mundo», https://www.bbc.com/mundo/noticias/2016/03/160325_tecnologia_microsoft_tay_bot_adolescente_inteligencia_artificial_racista_xenofoba_lb.
- [62] «Papers with Code - ELMo Explained», <https://paperswithcode.com/method/elmo>.
- [63] «ELMo (Embeddings from Language Models)», <https://www.iterate.ai/ai-glossary/what-is-elmo-embeddings-from-language-models>.
- [64] «IA generativa: ¿Cómo Transformers, BERT y GPT están cambiando el marketing digital?» Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.growketing.com/ia-generativa-como-transformers-bert-y-gpt-estan-cambiando-el-marketing-digital/>

- [65] «¿Qué es el GPT (transformador generativo preentrenado)? | IBM», <https://www.ibm.com/es-es/think/topics/gpt>.
- [66] «¿Qué es el BERT? Introducción a los modelos BERT», <https://www.datacamp.com/es/blog/what-is-bert-an-intro-to-bert-models>.
- [67] «ChatGPT: historia de la IA más conocida | Ecosistema MÁS», <https://ecosistemamas.ibercaja.es/actualidad/chatgpt-historia-de-la-ia-mas-conocida/>.
- [68] A. Heienickle, «Cronología de ChatGPT: la evolución de las conversaciones impulsadas por IA», <https://www.adogy.com/es/Cronolog%C3%ADa-de-ChatGPT%3A-la-evoluci%C3%B3n-de-las-conversaciones-impulsadas-por-IA/>.
- [69] «PaLM 2: qué es, cómo funciona, qué puede hacer y cómo probar este modelo de lenguaje por inteligencia artificial», <https://www.xataka.com/basics/palm-2-que-como-funciona-que-puede-hacer-como-probar-este-modelo-lenguaje-inteligencia-artificial>.
- [70] «Qué es LLaMA, cómo funciona y qué sabemos de la inteligencia artificial de Meta», <https://www.xataka.com/basics/que-llama-como-funciona-que-sabemos-inteligencia-artificial-meta>.
- [71] «Llama», <https://www.llama.com/>.
- [72] «Acerca de Grok», <https://help.x.com/es/using-x/about-grok>.
- [73] «xAI», <https://x.ai/>.
- [74] J. Devlin, M. W. Chang, K. Lee, y K. Toutanova, «BERT: Pre-training of deep bidirectional transformers for language understanding», en *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, Association for Computational Linguistics (ACL), 2019, pp. 4171-4186.
- [75] «Meet Claude», <https://www.anthropic.com/claude>.
- [76] «Claude», <https://claude.ai/login?returnTo=%2F%3F>.
- [77] «Breve historia visual de la inteligencia artificial». Accedido: 2 de noviembre de 2024. [En línea]. Disponible en: https://www.nationalgeographic.com/es/ciencia/breve-historia-visual-inteligencia-artificial_14419
- [78] «Marvin Minsky, cerebro de la inteligencia artificial | Ciencia | EL PAÍS». Accedido: 30 de noviembre de 2024. [En línea]. Disponible en: https://elpais.com/elpais/2016/01/26/ciencia/1453809513_840043.html
- [79] «¿Qué es el Machine Learning? | Aprendizaje Automático | UCM». Accedido: 10 de noviembre de 2024. [En línea]. Disponible en: <https://www.masterdatascienceucm.com/que-es-machine-learning/>
- [80] «¿Qué es el machine learning (ML)? | IBM». Accedido: 10 de noviembre de 2024. [En línea]. Disponible en: https://www.ibm.com/es-es/topics/machine-learning?mhsrc=ibmsearch_a&mhq=Que%20es%20el%20machine%20learning
- [81] «Machine Learning: ¿Qué es el aprendizaje automático y cómo funciona?» Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.algotive.ai/es-mx/blog/machine-learning-que-es-el-aprendizaje-autom%C3%A1tico-y-c%C3%B3mo-funciona>
- [82] «Aprendizaje supervisado: definición y aplicaciones - Telefónica». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.telefonica.com/es/sala-comunicacion/blog/aprendizaje-supervisado-definicion-aplicaciones/>

- [83] «¿Qué es el aprendizaje supervisado? | IBM». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/supervised-learning>
- [84] «Tipos de aprendizaje automático. La Inteligencia Artificial (IA) está en... | by Javier Luna Gonzalez | SoldAI | Medium». Accedido: 30 de noviembre de 2024. [En línea]. Disponible en: <https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-6413e3c615e2>
- [85] «¿Qué es el aprendizaje no supervisado? - Telefónica». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.telefonica.com/es/sala-comunicacion/blog/que-es-aprendizaje-no-supervisado/>
- [86] «¿Qué es el aprendizaje no supervisado? | Google Cloud». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://cloud.google.com/discover/what-is-unsupervised-learning?hl=es-419>
- [87] «¿Qué es el aprendizaje no supervisado? | IBM». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/unsupervised-learning>
- [88] «Semi-Supervised Learning...el gran desconocido». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://telefonicatech.com/blog/semi-supervised-learningel-gran-desconocido>
- [89] «¿Qué es el aprendizaje semisupervisado? | IBM». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/semi-supervised-learning>
- [90] «Tipos de aprendizaje que usan los algoritmos de Machine Learning». Accedido: 30 de noviembre de 2024. [En línea]. Disponible en: <https://decidesoluciones.es/tipos-de-aprendizaje-algoritmos-machine-learning/>
- [91] «Reinforcement Learning... el nuevo invitado». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://telefonicatech.com/blog/reinforcement-learning-el-nuevo-invitado>
- [92] «¿Qué es el aprendizaje de refuerzo? | IBM». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/reinforcement-learning>
- [93] «¿Cuáles son los tipos de algoritmos del machine learning? | APD». Accedido: 30 de noviembre de 2024. [En línea]. Disponible en: <https://www.apd.es/algoritmos-del-machine-learning/>
- [94] «¿Qué son las redes neuronales? Aplicaciones, tipos y ejemplos». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.bbva.com/es/innovacion/que-son-las-redes-neuronales-aplicaciones-tipos-y-ejemplos/>
- [95] «¿Qué son las redes neuronales? | IBM». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/neural-networks>
- [96] «Cómo Funcionan las Redes Neuronales: Explicación Detallada». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://data-universe.org/como-funcionan-las-redes-neuronales-explicacion-detallada/>
- [97] «Redes neuronales profundas: qué son y cómo funcionan». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://psicologaiymente.com/cultura/redes-neuronales-profundas>
- [98] «Redes neuronales profundas - Tipos y Características - Código Fuente». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.codigofuente.org/redes-neuronales-profundas-tipos-caracteristicas/>
- [99] «Redes Neuronales: Propagación hacia adelante y propagación hacia atrás | by Rubiales Alberto | Medium». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://rubialesalberto.medium.com/redes-neuronales-propagaci%C3%B3n-hacia-adelante-y-propagaci%C3%B3n-hacia-atr%C3%A1s-4745c0fb6286>

- [100] «Deep Neural Network (DNN) Explained | by ZOMEV | Medium». Accedido: 30 de noviembre de 2024. [En línea]. Disponible en: <https://medium.com/@zomev/deep-neural-network-dnn-explained-0f7311a0e869>
- [101] «¿Qué es la retropropagación? | IBM». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/think/topics/backpropagation>
- [102] «Backpropagation - KeyTrends». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://keytrends.ai/es/academy/glosario/inteligencia-artificial/backpropagation>
- [103] IATUTORIALES, «Red Neuronal de Retropropagación: Cómo Funciona el Algoritmo de Retropropagación - Ai y Programación», <https://aitutoriales.com/red-neuronal-de-retropropagacion-como-funciona-el-algoritmo-de-retropropagacion/>.
- [104] J. Sensio, «Herramientas en Python para ML», https://www.juansensio.com/blog/088_ml_tools.
- [105] «¿Qué es el deep learning? | IBM». Accedido: 10 de noviembre de 2024. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/deep-learning>
- [106] «Redes Neuronales Convolucionales: qué son, tipos y aplicaciones - Telefónica». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.telefonica.com/es/sala-comunicacion/blog/redes-neuronales-convolucionales-que-son-tipos-aplicaciones/>
- [107] «¿Qué son las redes neuronales convolucionales? | IBM». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/convolutional-neural-networks>
- [108] «Redes Neuronales Convolucionales: Todo lo que Debes Saber - Ai web Learning». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://aiweblearning.com/redes-neuronales-convolucionales/>
- [109] S. Contreras y F. De La Rosa, «Application of deep learning in mobile robotics for exploration and recognition of objects based on images», *2016 IEEE 11th Colombian Computing Conference, CCC 2016 - Conference Proceedings*, n.º September 2016, pp. 3471-3478, 2016, doi: 10.1109/ColumbianCC.2016.7750800.
- [110] «RNN (Red Neuronal Recurrente) | FOQUM». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://foqum.io/blog/termino/rnn-red-neuronal-recurrente/>
- [111] «¿Qué es una red neuronal recurrente (RNN)? | IBM». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/recurrent-neural-networks>
- [112] «Backpropagation Through Time (BPTT) - IA Blog». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://iartificial.blog/glossary/backpropagation-through-time-bptt/>
- [113] «¿Cuál es la principal ventaja de usar redes neuronales recurrentes (RNN) para manejar datos secuenciales o temporales? - Academia EITCA». Accedido: 1 de enero de 2025. [En línea]. Disponible en: <https://es.eitca.org/inteligencia-artificial/eitc-ai-dltf-aprendizaje-profundo-con-tensorflow/redes-neuronales-recurrentes-en-tensorflow/redes-neuronales-recurrentes-rnn/examen-revisi%C3%B3n-redes-neuronales-recurrentes-rnn/%C2%BFCu%C3%A1l-es-la-principal-ventaja-de-usar-redes-neuronales-recurrentes-rnns-para-manejar-datos-secuenciales-o-temporales%3F/>
- [114] «Disadvantages of Neural Networks/Deep Learning - why are they not preferred in all situations? : r/MachineLearning». Accedido: 1 de enero de 2025. [En línea]. Disponible en: https://www.reddit.com/r/MachineLearning/comments/3dnolr/disadvantages_of_neural_networksdeep_learning_why/?tl=es-es&rdt=45798
- [115] «The Neural Network Zoo - The Asimov Institute». Accedido: 30 de noviembre de 2024. [En línea]. Disponible en: <https://www.asimovinstitute.org/neural-network-zoo/>

- [116] «¿Qué es un autocodificador variacional? | IBM». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/think/topics/variational-autoencoder>
- [117] «IBM Systems Blog para Latinoamérica – Español». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/blogs/systems/mx-es/2019/11/tecnologia-de-red-generativa-antagonica-la-ia-pasa-al-primer-plano/>
- [118] «¿Qué son los modelos de difusión? | IBM». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/think/topics/diffusion-models>
- [119] «¿Qué es un autocodificador? | IBM». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/autoencoder>
- [120] «Autocodificadores variacionales: Cómo funcionan y por qué son importantes», <https://www.datacamp.com/es/tutorial/variational-autoencoders>.
- [121] «¿Qué es una GAN? - Explicación sobre las redes generativas antagónicas - AWS». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://aws.amazon.com/es/what-is/gan/>
- [122] «Redes GAN: ¿Qué son? Características, funciones y ventajas», <https://redesinformaticas.org/red-gan/>.
- [123] «¿Qué es una GAN? - Explicación sobre las redes generativas antagónicas - AWS», <https://aws.amazon.com/es/what-is/gan/>.
- [124] R. R. Abril, «Modelos de difusión: introducción • La Máquina Oráculo», <https://lamaquinaoraculo.com/deep-learning/modelos-de-difusion/>.
- [125] «Los modelos de difusión claramente explicados». Accedido: 1 de diciembre de 2024. [En línea]. Disponible en: <https://impulsatek.com/los-modelos-de-difusion-claramente-explicados/>
- [126] A. Vaswani *et al.*, «Attention Is All You Need», *Adv Neural Inf Process Syst*, vol. 2017-December, pp. 5999-6009, jun. 2017, Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/1706.03762v7>
- [127] «Un informático en el lado del mal: “Attention is all you need”: La investigación que revolucionó la Inteligencia Artificial con los Transformers (Parte 1)». Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://www.elladodelmal.com/2024/01/attention-is-all-you-need-la.html>
- [128] «¿Qué es un modelo de transformador? | IBM», <https://www.ibm.com/es-es/topics/transformer-model>.
- [129] «Cómo funcionan los transformadores: Una exploración detallada de la arquitectura de los transformadores | DataCamp». Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://www.datacamp.com/es/tutorial/how-transformers-work>
- [130] «Transformers: Una Revolución en el Aprendizaje Automático | RUBIK | TECH MAGAZINE». Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://rubiklab.tech/newstech/transformers-machine-learning>
- [131] «Qué son los Transformers en PLN: ventajas e inconvenientes». Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://blog.pangeanic.com/es/que-son-los-transformers-en-pln>
- [132] «¿Cómo funcionan los Transformers? en Español», <https://www.aprendemachinelarning.com/como-funcionan-los-transformers-espanol-nlp-gpt-bert/>.
- [133] «Así era ELIZA, el primer bot conversacional de la historia». Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://www.xataka.com/historia-tecnologica/asi-era-eliza-el-primer-bot-conversacional-de-la-historia>

- [134] P. Timsina, «Building Transformer Models with PyTorch 2.0 NLP, computer vision, and speech processing with PyTorch and Hugging Face».
- [135] A. Vaswani *et al.*, «Attention is all you need», en *Advances in Neural Information Processing Systems*, 2017.
- [136] «Redes neuronales recurrentes: qué son y cuáles son sus usos». Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://keepcoding.io/blog/que-son-las-redes-neuronales-recurrentes/>
- [137] «Redes Neuronales Recurrentes: explicación Detallada | Codificando Bits». Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://codificandobits.com/blog/redes-neuronales-recurrentes-explicacion-detallada/>
- [138] «Tutorial de redes neuronales recurrentes (RNN) | DataCamp». Accedido: 18 de noviembre de 2024. [En línea]. Disponible en: <https://www.datacamp.com/es/tutorial/tutorial-for-recurrent-neural-network>
- [139] «¿Qué son las Redes LSTM? | Codificando Bits». Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://codificandobits.com/blog/redes-lstm/>
- [140] «Cómo funcionan los transformadores: Una exploración detallada de la arquitectura de los transformadores», <https://www.datacamp.com/es/tutorial/how-transformers-work>.
- [141] A. Kucharavy, «From Deep Neural Language Models to LLMs», *Large Language Models in Cybersecurity*, pp. 3-17, 2024, doi: 10.1007/978-3-031-54827-7_1.
- [142] «Word Embeddings for NLP. Understanding word embeddings and their... | by Renu Khandelwal | Towards Data Science». Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://towardsdatascience.com/word-embeddings-for-nlp-5b72991e01d4>
- [143] «Deep Learning para la predicción de series temporales: Redes Neuronales Recurrentes (RNN) y Long Short-Term Memory (LSTM)». Accedido: 28 de noviembre de 2024. [En línea]. Disponible en: <https://cienciadedatos.net/documentos/py54-forecasting-con-deep-learning>
- [144] H. Kheddar, «Transformers and Large Language Models for Efficient Intrusion Detection Systems: A Comprehensive Survey», ago. 2024, Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2408.07583v1>
- [145] «¿Qué es la tokenización? Tipos, casos de uso, aplicación | DataCamp». Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://www.datacamp.com/es/blog/what-is-tokenization>
- [146] «¿Qué es un modelo de transformador? | Glosario | HPE España». Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://www.hpe.com/es/es/what-is/transformer-model.html>
- [147] «Large Language Models in Cybersecurity», *Large Language Models in Cybersecurity*, 2024, doi: 10.1007/978-3-031-54827-7.
- [148] «GitHub - eugeneyan/open-llms: 📄 A list of open LLMs available for commercial use.» Accedido: 9 de enero de 2025. [En línea]. Disponible en: <https://github.com/eugeneyan/open-llms>
- [149] «¿Qué son los grandes modelos de lenguaje (LLM)? | IBM». Accedido: 23 de octubre de 2024. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/large-language-models>
- [150] «12 aplicaciones útiles para los grandes modelos de lenguaje». Accedido: 3 de noviembre de 2024. [En línea]. Disponible en: <https://www.techopedia.com/es/aplicaciones-practicas-llm>
- [151] «Veinte casos de uso exitosos en el mundo real de los LLMs que demuestran el poder de la IA Generativa - Forbes Argentina». Accedido: 3 de noviembre de 2024. [En línea]. Disponible en: <https://www.forbesargentina.com/innovacion/veinte-casos-uso-exitosos-mundo-real-llms-demuestran-poder-ia-generativa-n49007>

- [152] «La inteligencia artificial superó a los médicos en el diagnóstico de enfermedades: qué pasará con esta profesión - Infobae». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://www.infobae.com/tecnologia/2024/11/21/la-inteligencia-artificial-supero-a-los-medicos-en-el-diagnostico-de-enfermedades-que-pasara-con-esta-profesion/>
- [153] «OpenAI | Creando inteligencia artificial segura y beneficiosa | IAOpen». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://www.aiopen.es/>
- [154] «M42 anuncia un nuevo LLM clínico para transformar el futuro de la IA en la atención médica - Forbes España». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://forbes.es/ultima-hora/350281/m42-anuncia-un-nuevo-llm-clinico-para-transformar-el-futuro-de-la-ia-en-la-atencion-medica/>
- [155] «Reflexión sobre el Uso de Modelos de Lenguaje Extendido (LLMs) en el Análisis de Comentarios en eCommerce y el Potencial de GPT-4o». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://www.woku.app/blog/reflexion-sobre-el-uso-de-modelos-de-lenguaje-extendido-llms-en-el-analisis-de-comentarios-en-ecommerce-y-el-potencial-de-gpt-4o>
- [156] «Desbloqueando Insights: Analizando Google Reviews con GPT-4o LLM OpenAI y SerpAPI | by Albert Gil López | Medium». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://medium.com/@jddam/desbloqueando-insights-analizando-google-reviews-con-gpt-4o-llm-openai-y-serpapi-d2316106829d>
- [157] «Mapas conceptuales con IA | Algor Education». Accedido: 4 de enero de 2025. [En línea]. Disponible en: https://www.algorededucation.com/es?utm_source=gads&utm_medium=paid-search&utm_campaign=Search_ES_Bottom_Purchase_050924&utm_term=mapa%20conceptual&utm_content=697990445414&gad_source=1&gclid=CjwKCAiA1eO7BhATEiwAm0Ee-D2eTrIoeb0slrgDLyAMbg9eiOVfqL50CqsHOQWWa3tyNm1tYT15KxoCBMYQAvD_BwE
- [158] «¿Cómo hacer un mapa mental en línea con ayuda de la IA? - Educational Tools». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://educational.tools/es/como-hacer-un-mapa-mental-en-linea-con-ayuda-de-la-ia/>
- [159] «PrivateGPT para resumen de libros: variables de configuración de prueba y clasificación | HackerNoon». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://hackernoon.com/lang/es/variables-de-configuracion-de-clasificacion-y-pruebas-de-resumen-de-libros-privadas-gpt>
- [160] «Creación de resúmenes con IA | Google Cloud». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://cloud.google.com/use-cases/ai-summarization?hl=es-419>
- [161] «Claude vs Bard vs GPT-4: evaluando resúmenes de textos LLM». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://www.nucleoo.com/blog/claude-vs-bard-vs-gp-evaluando-resumenes-de-textos-de-llm/>
- [162] «Los mejores chatbots de IA de 2024: ChatGPT y alternativas». Accedido: 4 de enero de 2025. [En línea]. Disponible en: https://www.zendesk.es/service/messaging/chatbot/?psafe_param=1&utm_source=google&utm_medium=Search-Paid&utm_network=g&utm_campaign=SE_AW_EM_ES_ES_N_Sup_NB_Core_Zeta_All_H&matchtype=&utm_term=&utm_content=605024036530&theme=&gad_source=1&gclid=CjwKCAiA1eO7BhATEiwAm0Ee-JT_xVjRLbvH5cqkbls0uCm98SijT8llTp0XknCDOE2tsHDcy2-VxoCRQgQAvD_BwE
- [163] «Top LLMs for Coding All Developers Should Know About». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://autogpt.net/top-llms-for-coding-all-developers-should-know-about/>

- [164] «Cómo realizar la generación de código con modelos LLM - novita.ai». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://blogs.novita.ai/es/how-to-perform-code-generation-with-llm-models/>
- [165] «Generación de código con modelo LLM». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://keepcoding.io/blog/generacion-de-codigo-con-modelo-llm/>
- [166] «¿Qué es el análisis de sentimiento? | IBM». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/sentiment-analysis>
- [167] «Análisis de Sentimientos en LLM y Diseño UX/UI | KeepCoding». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://keepcoding.io/blog/analisis-de-sentimientos-en-llm-y-diseno-ux-ui/>
- [168] «El estado de los LLM multilingües: más allá del inglés - Unite.AI». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://www.unite.ai/es/the-state-of-multilingual-llms-moving-beyond-english/>
- [169] «Grandes modelos de lenguaje y traducción automática - LanguageWire». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://www.languagewire.com/es-es/blog/llm-hyper-personalization>
- [170] «Stable Diffusion en línea». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://stablediffusionweb.com/es>
- [171] «Midjourney». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://www.midjourney.com/home>
- [172] «DALL-E Free AI Image Generator | Powered by Dall-E». Accedido: 4 de enero de 2025. [En línea]. Disponible en: https://www.dall-e-free.com/?gad_source=1&gclid=CjwKCAiA1eO7BhATEiwAm0Ee-J2Mkeo-z8GtaBkxbnrslYpF7oWu8bNKRxLAdtgTsR9oMmfauoHlBoCe20QAvD_BwE
- [173] «Asistentes virtuales: beneficios y aplicaciones». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://www.telefonicaempresas.es/grandes-empresas/blog/asistentes-virtuales-inteligencia-artificial-atencion-cliente/>
- [174] «Automate Video Transcribing with LLM (Simple Web Interface Included) | by Kirill Gorbushin | Medium». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://medium.com/@kgvrtr/automate-video-transcribing-with-llm-simple-web-interface-included-586c57a2f489>
- [175] «Create any Video-to-Text LLM with Claude-3». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://lightning.ai/cvetkotim/studios/create-any-video-to-text-llm-with-claude-3?section=featured>
- [176] «Using OPENAI's Whisper and Mixtral8x7B LLM to Transcribe and Correct Grammar from Videos | by Leandro M Losaria | Medium». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://brandolosaria.medium.com/using-openai-whisper-and-mixtral8x7b-to-transcribe-and-correct-grammar-from-videos-da1d243fc157>
- [177] «Using LLMs to Evaluate and Improve Automated Transcription Quality – WebRTC.ventures». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://webrtc.ventures/2024/05/using-llms-to-evaluate-and-improve-automated-transcription-quality/>
- [178] «How to transcribe audio using a Large Language Model – Look back in respect». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://mashimo.wordpress.com/2023/08/31/how-to-transcribe-audio-using-a-large-language-model/>

- [179] «El papel de los LLMs en la investigación de mercado.» Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://intel.goodrebels.com/p/inteligencia-artificial-llms-investigacion-mercado>
- [180] «Investigación de palabras clave para SEO utilizando Python y LLM». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://alprado.com/blog/investigacion-de-palabras-clave-para-seo/>
- [181] «Sales Forecast using LLM's. To use an open-source GPT model for... | by ravindar madishetty | Medium». Accedido: 4 de enero de 2025. [En línea]. Disponible en: <https://medium.com/@ravindarmadishetty/sales-forecast-using-llms-a3571f275403>
- [182] «¿Qué es la ciberseguridad? | IBM», <https://www.ibm.com/es-es/topics/cybersecurity>.
- [183] I. - Osi, «Guía de ciberseguridad La ciberseguridad al alcance de todos *». [En línea]. Disponible en: <https://www.incibe.es>
- [184] «¿Qué es la ciberseguridad?», https://latam.kaspersky.com/resource-center/definitions/what-is-cyber-security?srsId=AfmBOootYu98_0Ztdp34jOfIkE0aE7oeMHZBHyBAxZDZ5OWCmLytJH46.
- [185] «Ciberseguridad». Accedido: 2 de enero de 2025. [En línea]. Disponible en: <https://www.infosecuritymexico.com/es/ciberseguridad.html>
- [186] «¿Qué es la ciberseguridad? Aprende cómo protegerte en la red | UNIR». Accedido: 2 de enero de 2025. [En línea]. Disponible en: <https://www.unir.net/revista/ingenieria/que-es-ciberseguridad/>
- [187] «¿Qué es la ciberseguridad? - Explicación de la ciberseguridad - AWS». Accedido: 2 de enero de 2025. [En línea]. Disponible en: <https://aws.amazon.com/es/what-is/cybersecurity/>
- [188] «Ciberdefensa: ¿Qué es y por qué es importante?» Accedido: 2 de enero de 2025. [En línea]. Disponible en: <https://globalt4e.com/ciberdefensa-que-es-y-por-que-es-importante/>
- [189] «¿Qué es la Ciberdefensa? Principales funciones y aplicaciones». Accedido: 2 de enero de 2025. [En línea]. Disponible en: <https://ecuador.unir.net/actualidad-unir/que-es-ciberdefensa/>
- [190] «¿Qué es la ciberdefensa? - Seguritecnia». Accedido: 2 de enero de 2025. [En línea]. Disponible en: https://www.seguritecnia.es/actualidad/que-es-la-ciberdefensa_20230203.html
- [191] «¿Qué es la ciberdefensa?» Accedido: 2 de enero de 2025. [En línea]. Disponible en: <https://www.avenuglobal.com/blog/que-es-la-ciberdefensa-diferencia-entre-ciberdefensa-y-ciberseguridad>
- [192] «Defense Primer: Cyberspace Operations Overview». [En línea]. Disponible en: <https://crsreports.congress.gov>
- [193] M. De Defensa, «Documentos de Seguridad y Defensa 60 Estrategia de la información y seguridad en el ciberespacio Escuela de Altos Estudios de la Defensa», Accedido: 2 de enero de 2025. [En línea]. Disponible en: <http://publicacionesoficiales.boe.es/>
- [194] «Estrategia Nacional de Ciberseguridad 2019».
- [195] «BOE-A-2019-6347 Orden PCI/487/2019, de 26 de abril, por la que se publica la Estrategia Nacional de Ciberseguridad 2019, aprobada por el Consejo de Seguridad Nacional.» Accedido: 2 de enero de 2025. [En línea]. Disponible en: <https://www.boe.es/eli/es/o/2019/04/26/pci487>
- [196] ignacio.hinojal, «El coste de los ciberataques supone cerca del 1,5% del PIB», https://www.redseguridad.com/actualidad/informes-ciberseguridad/el-coste-de-los-ciberataques-ya-supone-cerca-del-15-del-pib-mundial_20240207.html.

- [197] W. N. Caballero y P. R. Jenkins, «On Large Language Models in National Security Applications», jul. 2024, Accedido: 2 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2407.03453v1>
- [198] «DARPA Aims to Develop AI, Autonomy Applications Warfighters Can Trust», <https://www.defense.gov/News/News-Stories/Article/Article/3722849/darpa-aims-to-develop-ai-autonomy-applications-warfighters-can-trust/>.
- [199] C. Trifiletti, «NATO Leverages MITRE's AI Expertise», <https://www.mitre.org/news-insights/impact-story/nato-leverages-mitres-ai-expertise>.
- [200] «Seguridad y auditorías en Modelos grandes del lenguaje (LLM)». Accedido: 3 de noviembre de 2024. [En línea]. Disponible en: <https://www.codemotion.com/magazine/es/ciberseguridad/seguridad-y-auditorias-en-modelos-grandes-del-lenguaje/>
- [201] «Grandes modelos de lenguaje (LLM) en ciberseguridad: bienvenido profesional de TI». Accedido: 3 de noviembre de 2024. [En línea]. Disponible en: <https://es.itpedia.nl/2024/05/25/large-language-models-llms-in-cyber-security/>
- [202] «Seis formas en que los 'chatbots' de IA generativa y los LLM pueden mejorar la ciberseguridad | Computerworld.es». Accedido: 3 de noviembre de 2024. [En línea]. Disponible en: <https://www.computerworld.es/article/2107246/seis-formas-en-que-los-chatbots-de-ia-generativa-y-los-llm-pueden-mejorar-la-ciberseguridad.html>
- [203] M. A. Ferrag *et al.*, «Revolutionizing Cyber Threat Detection with Large Language Models: A Privacy-Preserving BERT-Based Lightweight Model for IoT/IIoT Devices», *IEEE Access*, vol. 12, pp. 23733-23750, jun. 2024, doi: 10.1109/ACCESS.2024.3363469.
- [204] «(PDF) A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks». Accedido: 2 de enero de 2025. [En línea]. Disponible en: https://www.researchgate.net/publication/320366926_A_Deep_Learning_Approach_for_Intrusion_Detection_Using_Recurrent_Neural_Networks
- [205] C. Yin, Y. Zhu, J. Fei, y X. He, «A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks», *IEEE Access*, vol. 5, pp. 21954-21961, oct. 2017, doi: 10.1109/ACCESS.2017.2762418.
- [206] A. Chawla, B. Lee, S. Fallon, y P. Jacob, «Host Based Intrusion Detection System with Combined CNN/RNN Model», *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11329 LNAI, pp. 149-158, 2019, doi: 10.1007/978-3-030-13453-2_12/TABLES/1.
- [207] «(PDF) LLM-Based Security Automation: Revolutionizing Threat Detection and Incident Response AUTHORS». Accedido: 6 de enero de 2025. [En línea]. Disponible en: https://www.researchgate.net/publication/387398356_LLM-Based_Security_Automation_Revolutionizing_Threat_Detection_and_Incident_Response_AUTHORS
- [208] «AIOps Solutions | IBM». Accedido: 6 de enero de 2025. [En línea]. Disponible en: https://www.ibm.com/aiops?utm_content=SRCWW&p1=Search&p4=43700081258911211&p5=p&p9=58700008827799435&gad_source=1&gclid=CjwKCAiAm-67BhBlEiwAEVftNjF201xFctQwT3d7NQYJCSXxypic1f_oCkz3tQeqdCvHY4pWKIDixoCYZYQAvD_BwE&gclid=aw.ds
- [209] M. A. Uddin y I. H. Sarker, «An Explainable Transformer-based Model for Phishing Email Detection: A Large Language Model Approach», feb. 2024, Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2402.13871v1>

- [210] S. Jamal, H. Wimmer, y I. H. Sarker, «An Improved Transformer-based Model for Detecting Phishing, Spam, and Ham: A Large Language Model Approach», nov. 2023, Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2311.04913v2>
- [211] K. Afane, W. Wei, Y. Mao, J. Farooq, y J. Chen, «Next-Generation Phishing: How LLM Agents Empower Cyber Attackers», nov. 2024, Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2411.13874v1>
- [212] R. Fang, R. Bindu, A. Gupta, y D. Kang, «LLM Agents can Autonomously Exploit One-day Vulnerabilities», abr. 2024, Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2404.08144v2>
- [213] G. Deng *et al.*, «PentestGPT: An LLM-empowered Automatic Penetration Testing Tool», ago. 2023, Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2308.06782v2>
- [214] «Using LLMs for Automated IT Incident Management - OnPage». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://www.onpage.com/using-llms-for-automated-it-incident-management/>
- [215] «AWS | Community | Incident Response in the Age of LLMs - New Opportunities for Security Teams». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://community.aws/content/2mErOmJ4DQShZFt5DN7Zxj7mkgo/incident-response-in-the-age-of-llms-new-opportunities-for-security-teams>
- [216] «How Meta Uses LLMs to Improve Incident Response (and how you can too) - Parity». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://www.tryparity.com/blog/how-meta-uses-llms-to-improve-incident-response>
- [217] «Seguridad del chatbot - IBM watsonx Assistant». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/products/watsonx-assistant/enterprise-security>
- [218] «Estos 12 bots de ChatGPT especializados en ciberseguridad te ayudarán con tu trabajo o estudios en seguridad informática». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://www.adslzone.net/noticias/ia/bots-chatgpt-ciberseguridad-trabajo/>
- [219] «Large Language Models (LLMs) in Cyber Security - Welcome IT professional». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://en.itpedia.nl/2024/05/25/large-language-models-llms-in-cyber-security/>
- [220] «Un informático en el lado del mal: Un Programa de Formación para usar Inteligencia Artificial en Seguridad Ofensiva & Seguridad Defensiva». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://www.elladodelmal.com/2024/04/un-programa-de-formacion-para-usar.html>
- [221] J. Rando, F. Perez-Cruz, y B. Hitaj, «PassGPT: Password Modeling and (Guided) Generation with Large Language Models», *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14347 LNCS, pp. 164-183, jun. 2023, doi: 10.1007/978-3-031-51482-1_9.
- [222] F. McKee y D. Noever, «Chatbots in a Honeypot World», ene. 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2301.03771v1>
- [223] M. Sladić, V. Valeros, C. Catania, y S. Garcia, «LLM in the Shell: Generative Honeypots», *Proceedings - 9th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2024*, pp. 430-435, ago. 2023, doi: 10.1109/EuroSPW61312.2024.00054.

- [224] Reworr y D. Volkov, «LLM Agent HoneyPot: Monitoring AI Hacking Agents in the Wild», oct. 2024, Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2410.13919v1>
- [225] H. T. Otal y M. A. Canbaz, «LLM HoneyPot: Leveraging Large Language Models as Advanced Interactive HoneyPot Systems», sep. 2024, doi: 10.1109/CNS62487.2024.10735607.
- [226] E. Nwafor y H. Olufowobi, «CANBERT: A Language-based Intrusion Detection Model for In-vehicle Networks», *Proceedings - 21st IEEE International Conference on Machine Learning and Applications, ICMLA 2022*, pp. 294-299, 2022, doi: 10.1109/ICMLA55696.2022.00048.
- [227] «[2405.04760] Large Language Models for Cyber Security: A Systematic Literature Review». Accedido: 5 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2405.04760>
- [228] H. Xu *et al.*, «Large Language Models for Cyber Security: A Systematic Literature Review», may 2024, doi: 10.1145/3695988.
- [229] M. A. Ferrag, F. Alwahedi, A. Battah, B. Cherif, A. Mechri, y N. Tihanyi, «Generative AI and Large Language Models for Cyber Security: All Insights You Need», may 2024, Accedido: 5 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2405.12750v1>
- [230] «Large Language Models in Cybersecurity», *Large Language Models in Cybersecurity*, 2024, doi: 10.1007/978-3-031-54827-7.
- [231] F. N. Motlagh, M. Hajizadeh, M. Majd, P. Najafi, F. Cheng, y C. Meinel, «Large Language Models in Cybersecurity: State-of-the-Art», ene. 2024, Accedido: 5 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2402.00891v1>
- [232] J. He y M. Vechev, «Large Language Models for Code: Security Hardening and Adversarial Testing», *CCS 2023 - Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1865-1879, feb. 2023, doi: 10.1145/3576915.3623175.
- [233] W. Tann, Y. Liu, J. H. Sim, C. M. Seah, y E.-C. Chang, «Using Large Language Models for Cybersecurity Capture-The-Flag Challenges and Certification Questions», ago. 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2308.10443v1>
- [234] «[2112.02125] Examining Zero-Shot Vulnerability Repair with Large Language Models». Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2112.02125>
- [235] H. Pearce, B. Tan, B. Ahmad, R. Karri, y B. Dolan-Gavitt, «Examining Zero-Shot Vulnerability Repair with Large Language Models», dic. 2021, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <http://arxiv.org/abs/2112.02125>
- [236] N. Tihanyi, R. Jain, Y. Charalambous, M. A. Ferrag, Y. Sun, y L. C. Cordeiro, «A New Era in Software Security: Towards Self-Healing Software via Large Language Models and Formal Verification», may 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2305.14752v2>
- [237] F. Yu y M. V. Martin, «Honey, I Chunked the Passwords: Generating Semantic Honeywords Resistant to Targeted Attacks Using Pre-trained Language Models», *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13959 LNCS, pp. 89-108, 2023, doi: 10.1007/978-3-031-35504-2_5.
- [238] T. Vörös, S. Bergeron, y K. Berlin, «Web Content Filtering through knowledge distillation of Large Language Models», *CEUR Workshop Proc*, vol. 3652, pp. 1-21, may 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2305.05027v2>
- [239] G. Sandoval, H. Pearce, T. Nys, R. Karri, S. Garg, y B. Dolan-Gavitt, «Lost at C: A User Study on the Security Implications of Large Language Model Code Assistants», *32nd USENIX Security*

- Symposium, USENIX Security 2023*, vol. 3, pp. 2205-2222, ago. 2022, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2208.09727v4>
- [240] A. Tuor, R. Baerwolf, N. Knowles, B. Hutchinson, N. Nichols, y R. Jasper, «Recurrent Neural Network Language Models for Open Vocabulary Event-Level Cyber Anomaly Detection», dic. 2017, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/1712.00557v1>
- [241] M. Omar y S. Shiaeles, «VulDetect: A novel technique for detecting software vulnerabilities using Language Models», *Proceedings of the 2023 IEEE International Conference on Cyber Security and Resilience, CSR 2023*, pp. 105-110, 2023, doi: 10.1109/CSR57506.2023.10224924.
- [242] M. Gao, «The Advance of GPTs and Language Model in Cyber Security», *Highlights in Science, Engineering and Technology*, vol. 57, pp. 195-202, jul. 2023, doi: 10.54097/HSET.V57I.10001.
- [243] P. Ranade, A. Piplai, A. Joshi, y T. Finin, «CyBERT: Contextualized Embeddings for the Cybersecurity Domain», *Proceedings - 2021 IEEE International Conference on Big Data, Big Data 2021*, pp. 3334-3342, 2021, doi: 10.1109/BIGDATA52589.2021.9671824.
- [244] A. Happe y J. Cito, «Getting pwn'd by AI: Penetration Testing with Large Language Models», *ESEC/FSE 2023 - Proceedings of the 31st ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 2082-2086, jul. 2023, doi: 10.1145/3611643.3613083.
- [245] K. Ameri, M. Hempel, H. Sharif, J. Lopez, y K. Perumalla, «CyBERT: Cybersecurity Claim Classification by Fine-Tuning the BERT Language Model», *Journal of Cybersecurity and Privacy 2021, Vol. 1, Pages 615-637*, vol. 1, n.º 4, pp. 615-637, nov. 2021, doi: 10.3390/JCP1040031.
- [246] M. Bayer, P. Kuehn, R. Shanehsaz, y C. Reuter, «CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain», *ACM Transactions on Privacy and Security*, vol. 27, n.º 2, dic. 2022, doi: 10.1145/3652594.
- [247] M. A. Ferrag *et al.*, «SecureFalcon: Are We There Yet in Automated Software Vulnerability Detection with LLMs?», jul. 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <http://arxiv.org/abs/2307.06616>
- [248] M. Sladić, V. Valeros, C. Catania, y S. Garcia, «LLM in the Shell: Generative Honeypots», *Proceedings - 9th IEEE European Symposium on Security and Privacy Workshops, Euro S and PW 2024*, pp. 430-435, ago. 2023, doi: 10.1109/EuroSPW61312.2024.00054.
- [249] B. Garvey y A. Svendsen, «Can Generative-AI (ChatGPT and Bard) Be Used as Red Team Avatars in Developing Foresight Scenarios?», *SSRN Electronic Journal*, ago. 2023, doi: 10.2139/SSRN.4541396.
- [250] T. Koide, N. Fukushi, H. Nakano, y D. Chiba, «Detecting Phishing Sites Using ChatGPT», jun. 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2306.05816v2>
- [251] S. Sakaoglu, «KARTAL: Web Application Vulnerability Hunting Using Large Language Models: Novel method for detecting logical vulnerabilities in web applications with finetuned Large Language Models», 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-337159>
- [252] G. Deng *et al.*, «PentestGPT: An LLM-empowered Automatic Penetration Testing Tool», ago. 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2308.06782v2>

- [253] I. David, L. Zhou, K. Qin, D. Song, L. Cavallaro, y A. Gervais, «Do you still need a manual smart contract audit?», jun. 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2306.12338v2>
- [254] C. Chen *et al.*, «When ChatGPT Meets Smart Contract Vulnerability Detection: How Far Are We?», *ACM Trans. Softw. Eng. Methodol.*, vol. 1, n.º 1, sep. 2023, doi: 10.1145/nnnnnnnn.nnnnnnn.
- [255] E. Cambiaso y L. Caviglione, «Scamming the Scammers: Using ChatGPT to Reply Mails for Wasting Time and Resources», feb. 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2303.13521v1>
- [256] P. V. S. Charan, H. Chunduri, P. M. Anand, y S. K. Shukla, «From Text to MITRE Techniques: Exploring the Malicious Use of Large Language Models for Generating Cyber Attack Payloads», may 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2305.15336v1>
- [257] R. Karanjai, «Targeted Phishing Campaigns using Large Scale Language Models», dic. 2022, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2301.00665v1>
- [258] M. Beckerich, L. Plein, y S. Coronado, «RatGPT: Turning online LLMs into Proxies for Malware Attacks», ago. 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2308.09183v2>
- [259] S. S. Roy, K. V. Naragam, y S. Nilizadeh, «Generating Phishing Attacks using ChatGPT», may 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2305.05133v1>
- [260] «GPThreats-3: Is Automatic Malware Generation a Threat? | IEEE Conference Publication | IEEE Xplore». Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://ieeexplore.ieee.org/document/10188649>
- [261] «(PDF) Large Language Models Can Be Used To Effectively Scale Spear Phishing Campaigns». Accedido: 3 de enero de 2025. [En línea]. Disponible en: https://www.researchgate.net/publication/370688440_Large_Language_Models_Can_Be_Used_To_Effectively_Scale_Spear_Phishing_Campaigns
- [262] Y. M. Pa Pa, S. Tanizaki, T. Kou, M. Van Eeten, K. Yoshioka, y T. Matsumoto, «An Attacker's Dream? Exploring the Capabilities of ChatGPT for Developing Malware», *ACM International Conference Proceeding Series*, pp. 10-18, ago. 2023, doi: 10.1145/3607505.3607513.
- [263] J. S. Salesforce y P. T. Zerofox, «Generative Models for Spear Phishing Posts on Social Media», feb. 2018, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/1802.05196v1>
- [264] E. Chatzoglou, G. Karopoulos, G. Kambourakis, y Z. Tsiatsikas, «Bypassing antivirus detection: old-school malware, new tricks», *ACM International Conference Proceeding Series*, may 2023, doi: 10.1145/3600160.3605010.
- [265] Y. M. Pa Pa, S. Tanizaki, T. Kou, M. Van Eeten, K. Yoshioka, y T. Matsumoto, «An Attacker's Dream? Exploring the Capabilities of ChatGPT for Developing Malware», *ACM International Conference Proceeding Series*, pp. 10-18, ago. 2023, doi: 10.1145/3607505.3607513.
- [266] L. Salewski, S. Alaniz, I. Rio-Torto, E. Schulz, y Z. Akata, «In-Context Impersonation Reveals Large Language Models' Strengths and Biases», *Adv Neural Inf Process Syst*, vol. 36, may 2023, Accedido: 3 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2305.14930v2>
- [267] S. E. Kreps, M. McCain, y M. Brundage, «All the News that's Fit to Fabricate: AI-Generated Text as a Tool of Media Misinformation», *SSRN Electronic Journal*, sep. 2020, doi: 10.2139/SSRN.3525002.

- [268] M. A. Ferrag, F. Alwahedi, A. Battah, B. Cherif, A. Mechri, y N. Tihanyi, «Generative AI and Large Language Models for Cyber Security: All Insights You Need», may 2024, Accedido: 20 de octubre de 2024. [En línea]. Disponible en: <http://arxiv.org/abs/2405.12750>
- [269] «¿Cuáles son los 10 principales riesgos de OWASP para los LLMs?», <https://www.cloudflare.com/es-es/learning/ai/owasp-top-10-risks-for-llms/>.
- [270] «¿Qué es un ataque de inyección de prompts? | IBM». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/prompt-injection>
- [271] «Riesgos de los modelos de lenguaje grande (LLM) y la solución de Copilot for Security - Bravent». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://www.bravent.net/noticias/revolucion-ciberseguridad-riesgos-modelos-lenguaje-grande-llm-solucion-copilot-security/>
- [272] «Ataques a la Inteligencia Artificial (III): Data Poisoning». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://telefonicatech.com/blog/ataques-a-la-inteligencia-artificial-iii-data-poisoning>
- [273] «¿Cuáles son los 10 principales riesgos de OWASP para los LLMs? | Cloudflare». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://www.cloudflare.com/es-es/learning/ai/owasp-top-10-risks-for-llms/>
- [274] «IA y Riesgos de Confidencialidad: Se puede alimentar a ChatGPT con data sensible - SOC-CSIRT CEDIA». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://socsirt.cedia.edu.ec/blog/2023/04/26/ia-y-riesgos-de-confidencialidad-se-puede-alimentar-a-chatgpt-con-data-sensible/>
- [275] «Los plugins de ChatGPT ponen en riesgo tu información personal | WIRED». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://es.wired.com/articulos/plugins-de-chatgpt-ponen-en-riesgo-tu-informacion-personal>
- [276] «What is excessive agency? | Tutorial & examples | Snyk Learn». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://learn.snyk.io/lesson/excessive-agency/>
- [277] «Las 10 mejores formas de eliminar las alucinaciones del LLM - Skim AI». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://skimai.com/es/las-10-mejores-formas-de-eliminar-las-alucinaciones-de-llm/>
- [278] «Las vulnerabilidades y amenazas a la seguridad que enfrentan los grandes modelos lingüísticos - Unite.AI». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://www.unite.ai/es/the-vulnerabilities-and-security-threats-facing-large-language-models/>
- [279] «Asegurando la IA y los LLM: El Papel Crítico de los Controles de Acceso | BoxyHQ». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://boxyhq.com/blog/asegurando-la-ia-y-los-llm:-el-papel-cr%C3%ADtico-de-los-controles-de-acceso>
- [280] «Seguridad y auditorías en Modelos grandes del lenguaje (LLM)». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://www.codemotion.com/magazine/es/ciberseguridad/seguridad-y-auditorias-en-modelos-grandes-del-lenguaje/>
- [281] «theNET | Proteger los LLM | Cloudflare». Accedido: 8 de enero de 2025. [En línea]. Disponible en: <https://www.cloudflare.com/es-es/the-net/vulnerable-llm-ai/>
- [282] «Cinco formas de implementar su propio modelo LLM | CIO». Accedido: 10 de enero de 2025. [En línea]. Disponible en: <https://www.cio.com/article/1314035/cinco-formas-de-implementar-su-propio-modelo-llm.html>

- [283] «GitHub - tmylla/HackMentor: The repository of paper “HackMentor: Fine-Tuning Large Language Models for Cybersecurity”». Accedido: 10 de enero de 2025. [En línea]. Disponible en: <https://github.com/tmylla/HackMentor>
- [284] M. Bayer, P. Kuehn, R. Shanehsaz, y C. Reuter, «CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain», *ACM Transactions on Privacy and Security*, vol. 27, n.º 2, abr. 2024, doi: 10.1145/3652594.
- [285] M. Bayer, P. Kuehn, R. Shanehsaz, y C. Reuter, «CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain», *ACM Transactions on Privacy and Security*, vol. 27, n.º 2, abr. 2024, doi: 10.1145/3652594.
- [286] M. Bayer, P. Kuehn, R. Shanehsaz, y C. Reuter, «CySecBERT: A Domain-Adapted Language Model for the Cybersecurity Domain», *ACM Transactions on Privacy and Security*, vol. 27, n.º 2, abr. 2024, doi: 10.1145/3652594.
- [287] «CySecBERT · Models · Dataloop». Accedido: 10 de enero de 2025. [En línea]. Disponible en: https://dataloop.ai/library/model/markusbayer_cysecbert/
- [288] P. Ranade, A. Piplai, A. Joshi, y T. Finin, «CyBERT: Contextualized Embeddings for the Cybersecurity Domain», *Proceedings - 2021 IEEE International Conference on Big Data, Big Data 2021*, pp. 3334-3342, 2021, doi: 10.1109/BIGDATA52589.2021.9671824.
- [289] «SynamicTechnologies/CYBERT · Hugging Face». Accedido: 10 de enero de 2025. [En línea]. Disponible en: <https://huggingface.co/SynamicTechnologies/CYBERT>
- [290] E. Aghaei, X. Niu, W. Shadid, y E. Al-Shaer, «SecureBERT: A Domain-Specific Language Model for Cybersecurity», *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 462 LNICST, pp. 39-56, abr. 2022, doi: 10.1007/978-3-031-25538-0_3.
- [291] E. Aghaei, X. Niu, W. Shadid, y E. Al-Shaer, «SecureBERT: A Domain-Specific Language Model for Cybersecurity», *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 462 LNICST, pp. 39-56, 2023, doi: 10.1007/978-3-031-25538-0_3.
- [292] E. Aghaei, X. Niu, W. Shadid, y E. Al-Shaer, «SecureBERT: A Domain-Specific Language Model for Cybersecurity», *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 462 LNICST, pp. 39-56, 2023, doi: 10.1007/978-3-031-25538-0_3.
- [293] «PentestGPT». Accedido: 10 de enero de 2025. [En línea]. Disponible en: <https://pentestgpt.ai/1a62ddc8-b04e-42d6-9344-bcb234ee0350/chat>
- [294] «GitHub - GreyDGL/PentestGPT: A GPT-empowered penetration testing tool». Accedido: 10 de enero de 2025. [En línea]. Disponible en: <https://github.com/GreyDGL/PentestGPT>
- [295] G. Deng *et al.*, «PentestGPT: An LLM-empowered Automatic Penetration Testing Tool», ago. 2023, Accedido: 10 de enero de 2025. [En línea]. Disponible en: <https://arxiv.org/abs/2308.06782v2>
- [296] M. A. Ferrag *et al.*, «Revolutionizing Cyber Threat Detection with Large Language Models: A Privacy-Preserving BERT-Based Lightweight Model for IoT/IIoT Devices», *IEEE Access*, vol. 12, pp. 23733-23750, 2024, doi: 10.1109/ACCESS.2024.3363469.
- [297] «The Good and Bad of LLMs in Digital Forensics | by Prof Bill Buchanan OBE FRSE | ASecuritySite: When Bob Met Alice | Medium». Accedido: 6 de enero de 2025. [En línea]. Disponible en: <https://medium.com/asecuritysite-when-bob-met-alice/the-good-and-bad-of-llms-in-digital-forensics-c005fdd93f39>

ANEXO I: ALGORITMOS DE APRENDIZAJE MÁS UTILIZADOS

- Algoritmos de regresión: el programa de aprendizaje debe estimar y comprender las relaciones que hay entre las variables. El análisis de regresión se enfoca en una variable dependiente y en un conjunto de variables cambiantes. Son utilizadas para la predicción y el diagnóstico.
- Algoritmos bayesianos: son algoritmos basados en el teorema de Bayes, cada valor es clasificado de forma independiente, lo que permite predecir clases o categorías en función de un conjunto de características utilizando la probabilidad, son muy utilizados y se usan para clasificación.
- Algoritmos de agrupación: busca grupos dentro de los datos con el número de grupos representados por la variable K. Su funcionamiento es iterativo, asigna a cada valor a uno de los K grupos según las características dadas. Son usados en el aprendizaje no supervisado, ya que son buenos categorizando datos no etiquetados o no asignados previamente a un grupo o categoría.
- Algoritmos de árbol de decisión: son similares a un diagrama de flujo que ilustran cada vía o resultado posible de una decisión. Cada nodo representa una prueba en una variable específica y cada rama es el resultado de esa prueba.
- Algoritmos de redes neuronales: compuestas por capas que se interconectan. Se basan en el funcionamiento de las neuronas humanas y cómo procesan la información. Aprenden con el ejemplo y la experiencia y son útiles para modelar relaciones no lineales o donde la relación entre las variables de entrada es difícil de entender.
- Algoritmos de reducción de dimensión: reducen el número de variables para encontrar la información requerida.
- Algoritmos de Aprendizaje Profundo: utilizan varias capas de redes neuronales, que pasan datos simplificados a la siguiente capa. Cuando los datos no están estructurados, como en las imágenes, no tienen buenos resultados, para conjuntos de datos estructurados pequeños sí tienen resultados satisfactorios.
- Algoritmos de Máquinas de Soporte Vectorial (SVM): son técnicas de clasificación y regresión. Son particularmente efectivos en espacios de alta dimensión y cuando el número de dimensiones es mayor que el número de muestras.
- Algoritmos de Ensemble: combinan las predicciones de varios modelos base para mejorar la precisión y la robustez de los resultados.
 - *Bagging*: usa métodos como Random Forest, que construyen múltiples árboles de decisión en subconjuntos del *dataset* y promedian sus predicciones.
 - *Boosting*: Algoritmos como AdaBoost y Gradient Boosting que ajustan secuencialmente modelos débiles y combinan sus predicciones.
- Algoritmos de Aprendizaje por Refuerzo: el agente aprende a tomar decisiones secuenciales, y recibe una recompensa o un castigo en función del éxito, el agente trata de maximizar la recompensa.
 - *Q-Learning*: busca aprender la política de acción-estado que maximiza la recompensa.
 - *Deep Q-Networks (DQN)*: Utiliza redes neuronales profundas para aproximar la función Q.
- Algoritmos de series temporales: analizan datos secuenciales y hacen predicciones basadas en los patrones vistos anteriormente.
 - *ARIMA* (AutoRegressive Integrated Moving Average): Modelo utilizado para el análisis y pronóstico de series temporales.
 - *Prophet*: Modelo desarrollado por Facebook para el pronóstico de series temporales

- Algoritmos de asociación: encuentran relaciones de asociación en grandes conjuntos de datos
 - *Apriori*: Usado en minería de reglas de asociación, busca patrones de elementos que aparecen frecuentemente juntos.
 - *Eclat (Equivalence Class Transformation)*: Variante del algoritmo Apriori que utiliza un enfoque de intersección de conjuntos para encontrar ítems frecuentes.
- Algoritmos de optimización: buscan la mejor solución a un problema dado un conjunto de restricciones.
 - *Algoritmos Genéticos*: Técnicas basadas en la selección natural para encontrar soluciones óptimas.
 - *Simulated Annealing*: Técnica de optimización que simula el proceso de enfriamiento de metales para encontrar un estado de energía mínima.
- Algoritmos de filtrado colaborativo: se usan en sistemas de recomendación para predecir las preferencias de los usuarios basándose en elecciones anteriores.
 - Filtrado Colaborativo Basado en Usuarios: Recomienda ítems basados en usuarios similares.
 - Filtrado Colaborativo Basado en Ítems: Recomienda ítems basados en ítems similares.
- Algoritmos de reducción de dimensionalidad: reducen el número de variables de entrada para simplificar el modelo y reducir el ruido.
 - Análisis de Componentes Principales (PCA): Transforma las variables originales en un nuevo conjunto de variables no correlacionadas.
 - t-SNE (t-Distributed Stochastic Neighbor Embedding): Técnica para la reducción de dimensionalidad que es especialmente útil para la visualización de datos en 2D o 3D.
- Algoritmos de Procesamiento de Lenguaje Natural (NLP): permiten a las computadoras entender, interpretar y generar lenguaje humano.
 - Modelos de Bolsas de Palabras (Bag-of-Words): Representación de texto que cuenta las ocurrencias de palabras.
 - Word2Vec: Modelos de redes neuronales para aprender representaciones vectoriales de palabras.

ANEXO II:

LISTADO DE LLM'S OPEN SOURCE

Modelo de Lenguaje	Artículo/Blog	Params (B)	Longitud del contexto	Licencia
T5	Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer	0.06 - 11	512	Apache 2.0
RWKV 4	The RWKV Language Model (and my LM tricks)	0.1 - 14	infinity (RNN)	Apache 2.0
GPT-NeoX-20B	GPT-NeoX-20B: An Open-Source Autoregressive Language Model	20	2048	Apache 2.0
YaLM-100B	Yandex publishes YaLM 100B, the largest GPT-like neural network in open source	100	1024	Apache 2.0
UL2	UL2 20B: An Open Source Unified Language Learner	20	512, 2048	Apache 2.0
Bloom	BLOOM: A 176B-Parameter Open-Access Multilingual Language Model	176	2048	OpenRAIL-M v1
ChatGLM	ChatGLM, Github	6	2048	Custom Free with some usage restriction (might require registration)
Cerebras-GPT	Cerebras-GPT: A Family of Open, Compute-efficient, Large Language Models (Paper)	0.111 - 13	2048	Apache 2.0
Open Assistant (Pythia family)	Democratizing Large Language Model Alignment	12	2048	Apache 2.0
Pythia	Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling	0.07 - 12	2048	Apache 2.0
Dolly	Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM	3, 7, 12	2048	MIT
StableLM-Alpha	Stability AI Launches the First of its StableLM Suite of Language Models	mar-65	4096	CC BY-SA-4.0
FastChat-T5	We are excited to release FastChat-T5: our compact and commercial-friendly chatbot!	3	512	Apache 2.0
DLite	Announcing DLite V2: Lightweight, Open LLMs That Can Run Anywhere	0.124 - 1.5	1024	Apache 2.0

h2oGPT	Building the World's Best Open-Source Large Language Model: H2O.ai's Journey	dic-20	256 - 2048	Apache 2.0
MPT-7B	Introducing MPT-7B: A New Standard for Open-Source, Commercially Usable LLMs	7	84k (ALiBi)	Apache 2.0, CC BY-SA-3.0
RedPajama-INCITE	Releasing 3B and 7B RedPajama-INCITE family of models including base, instruction-tuned & chat models	03-jul	2048	Apache 2.0
OpenLLaMA	OpenLLaMA: An Open Reproduction of LLaMA The RefinedWeb Dataset for Falcon LLM:	3, 7	2048	Apache 2.0
Falcon	Outperforming Curated Corpora with Web Data, and Web Data Only	180, 40, 7	2048	Apache 2.0
GPT-J-6B	GPT-J-6B: 6B JAX-Based Transformer	6	2048	Apache 2.0
MPT-30B	MPT-30B: Raising the bar for open-source foundation models	30	8192	Apache 2.0, CC BY-SA-3.0
LLaMA 2	Llama 2: Open Foundation and Fine-Tuned Chat Models	jul-70	4096	Custom Free if you have under 700M users and you cannot use LLaMA outputs to train other LLMs besides LLaMA and its derivatives
ChatGLM2	ChatGLM2-6B, Github	6	32k	Custom Free with some usage restriction (might require registration)
XGen-7B	Long Sequence Modeling with XGen	7	4096, 8192	Apache 2.0
Jais-13b	Jais and Jais-chat: Arabic-Centric Foundation and Instruction-Tuned Open Generative Large Language Models	13	2048	Apache 2.0
OpenHermes	Nous Research	7, 13	4096	MIT
OpenLM	Open LM: a minimal but performative language modeling (LM) repository	1, 7	2048	MIT
Mistral 7B	Mistral 7B	7	4096-16K with Sliding Windows	Apache 2.0
ChatGLM3	ChatGLM3	6	8192, 32k, 128k	Custom Free with some usage restriction (might require registration)

Skywork	Skywork	13	4096	Custom Free with usage restriction and models trained on Skywork outputs become Skywork derivatives, subject to this license.
Jais-30b	Jais-30B: Expanding the Horizon in Open-Source Arabic NLP	30	2048	Apache 2.0
Zephyr	Website	7	8192	Apache 2.0
DeepSeek	Introducing DeepSeek LLM,	7, 67	4096	Custom Free with usage restriction and models trained on DeepSeek outputs become DeepSeek derivatives, subject to this license.
Mistral 7B v0.2	La Plateforme	7	32k	Apache 2.0
Mixtral 8x7B v0.1	Mixtral of experts	46.7	32k	Apache 2.0
LLM360 Amber	Introducing LLM360: Fully Transparent Open-Source LLMs	6.7	2048	Apache 2.0
SOLAR	Upstage	10.7	4096	apache-2.0
phi-2	Microsoft	2.7	2048	MIT
FLOR	FLOR-6.3B: a chinchilla-compliant model for Catalan, Spanish and English	0.76, 1.3, 6.3	2048	Apache 2.0 with usage restriction inherited from BLOOM
RWKV 5 v2	RWKV 5	0.4, 1.5, 3, 7	unlimited(RNN), trained on 4096 (and 16k for 3b)	Apache 2.0
OLMo	AI2	1,7	2048	Apache 2.0
Qwen1.5	Introducing Qwen1.5	7, 14, 72	32k	Custom Free if you have under 100M users and you cannot use Qwen outputs to train other LLMs besides Qwen and its derivatives
LWM	Large World Model (LWM)	7	128k, 256k, 512k, 1M	LLaMA 2 license
Jais-30b v3	Jais 30b v3	30	8192	Apache 2.0

Gemma	Technical report	02-jul	8192	Gemma Terms of Use Free with usage restriction and models trained on Gemma outputs become Gemma derivatives, subject to this license.
Grok-1	Open Release of Grok-1	314	8192	Apache 2.0
Qwen1.5 MoE	Qwen1.5-MoE: Matching 7B Model Performance with 1/3 Activated Parameters	14.3	8192	Custom Free if you have under 100M users and you cannot use Qwen outputs to train other LLMs besides Qwen and its derivatives
Jamba 0.1	Introducing Jamba: AI21's Groundbreaking SSM-Transformer Model	52	256k	Apache 2.0
Qwen1.5 32B	Qwen1.5-32B: Fitting the Capstone of the Qwen1.5 Language Model Series	32	32k	Custom Free if you have under 100M users and you cannot use Qwen outputs to train other LLMs besides Qwen and its derivatives
Mamba-7B	Toyota Research Institute	7	unlimited(RNN), trained on 2048	Apache 2.0
Mixtral8x22B v0.1	Cheaper, Better, Faster, Stronger	141	64k	Apache 2.0
Llama 3	Introducing Meta Llama 3, Meta Llama 3	8, 70	8192	Meta Llama 3 Community License Agreement Free if you have under 700M users and you cannot use LLaMA 3 outputs to train other LLMs besides LLaMA 3 and its derivatives
Phi-3 Mini	Introducing Phi-3, Technical Report	3.8	4096, 128k	MIT
OpenELM	OpenELM: An Efficient Language Model Family with Open Training and Inference Framework	0.27, 0.45, 1.1, 3	2048	Custom open license No usage or training restrictions
Snowflake Arctic	Snowflake Arctic: The Best LLM for Enterprise AI — Efficiently Intelligent, Truly Open	480	4096	Apache 2.0

Qwen1.5 110B	Qwen1.5-110B: The First 100B+ Model of the Qwen1.5 Series	110	32k	Custom Free if you have under 100M users and you cannot use Qwen outputs to train other LLMs besides Qwen and its derivatives
RWKV 6 v2.1	RWKV 6	1.6, 3, 7	unlimited(RNN), trained on 4096	Apache 2.0
DeepSeek-V2	DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model	236	128k	Custom Free with usage restriction and models trained on DeepSeek outputs become DeepSeek derivatives, subject to this license.
Fugaku-LLM	Release of "Fugaku-LLM" – a large language model trained on the supercomputer "Fugaku"	13	2048	Custom Free with usage restrictions
Falcon 2	Meet Falcon 2: TII Releases New AI Model Series, Outperforming Meta's New Llama 3	11	8192	Custom Apache 2.0 with mild acceptable use policy
Yi-1.5	Yi-1.5	6, 9, 34	4096	Apache 2.0
DeepSeek-V2-Lite	DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model	16	32k	Custom Free with usage restriction and models trained on DeepSeek outputs become DeepSeek derivatives, subject to this license.
Phi-3 small/medium	New models added to the Phi-3 family, available on Microsoft Azure, Technical Report	7, 14	4096, 128k	MIT
SantaCoder	SantaCoder: don't reach for the stars!	1.1	2048	OpenRAIL-M v1
CodeGen2	CodeGen2: Lessons for Training LLMs on Programming and Natural Languages	ene-16	2048	Apache 2.0
StarCoder	StarCoder: A State-of-the-Art LLM for Code, StarCoder: May the source be with you!	1.1-15	8192	OpenRAIL-M v1
StarChat Alpha	Creating a Coding Assistant with StarCoder	16	8192	OpenRAIL-M v1

Replit Code	Training a SOTA Code LLM in 1 week and Quantifying the Vibes — with Reza Shabani of Replit	2.7	infinity? (ALiBi)	CC BY-SA-4.0
CodeT5+	CodeT5+: Open Code Large Language Models for Code Understanding and Generation	0.22 - 16	512	BSD-3-Clause
XGen-7B	Long Sequence Modeling with XGen: A 7B LLM Trained on 8K Input Sequence Length	7	8192	Apache 2.0
CodeGen2.5	CodeGen2.5: Small, but mighty	7	2048	Apache 2.0
DeciCoder-1B	Introducing DeciCoder: The New Gold Standard in Efficient and Accurate Code Generation	1.1	2048	Apache 2.0
Code Llama	Code Llama: Open Foundation Models for Code	jul-34	4096	Custom Free if you have under 700M users and you cannot use LLaMA outputs to train other LLMs besides LLaMA and its derivatives

Tabla 10: Listado de LLM's open source

ANEXO III: LISTADO CVE QUE AFECTAN A LOS LLM'S

Nombre	Descripción
CVE-2024-8939	A vulnerability was found in the ilab model serve component, where improper handling of the best_of parameter in the vllm JSON web API can lead to a Denial of Service (DoS). The API used for LLM-based sentence or chat completion accepts a best_of parameter to return the best completion from several options. When this parameter is set to a large value, the API does not handle timeouts or resource exhaustion properly, allowing an attacker to cause a DoS by consuming excessive system resources. This leads to the API becoming unresponsive, preventing legitimate users from accessing the service.
CVE-2024-7783	mintplex-labs/anything-llm version latest contains a vulnerability where sensitive information, specifically a password, is improperly stored within a JWT (JSON Web Token) used as a bearer token in single user mode. When decoded, the JWT reveals the password in plaintext. This improper storage of sensitive information poses significant security risks, as an attacker who gains access to the JWT can easily decode it and retrieve the password. The issue is fixed in version 1.0.3.
CVE-2024-6961	RAIL documents are an XML-based format invented by Guardrails AI to enforce formatting checks on LLM outputs. Guardrails users that consume RAIL documents from external sources are vulnerable to XXE, which may cause leakage of internal file data via the SYSTEM entity.
CVE-2024-5826	In the latest version of vanna-ai/vanna, the `vanna.ask` function is vulnerable to remote code execution due to prompt injection. The root cause is the lack of a sandbox when executing LLM-generated code, allowing an attacker to manipulate the code executed by the `exec` function in `src/vanna/base/base.py`. This vulnerability can be exploited by an attacker to achieve remote code execution on the app backend server, potentially gaining full control of the server.
CVE-2024-53844	E.D.D.I (Enhanced Dialog Driven Interface) is a middleware to connect and manage LLM API bots. A path traversal vulnerability exists in the backup export functionality of EDDI, as implemented in `RestExportService.java`. This vulnerability allows an attacker to access sensitive files on the server by manipulating the `botFilename` parameter in requests. The application fails to sanitize user input, enabling malicious inputs such as `../%2f..%2fetc%2fpasswd` to access arbitrary files. However, the severity of this vulnerability is significantly limited because EDDI typically runs within a Docker container , which provides additional layers of isolation and restricted permissions. As a result, while this vulnerability exposes files within the container, it does not inherently threaten the underlying host system or other containers. A patch is required to sanitize and validate the botFilename input parameter. Users should ensure they are using version 5.4 which contains this patch. For temporary mitigation, access to the vulnerable endpoint should be restricted through firewall rules or authentication mechanisms.
CVE-2024-5216	A vulnerability in mintplex-labs/anything-llm allows for a Denial of Service (DoS) condition due to uncontrolled resource consumption. Specifically, the issue arises

Nombre	Descripción
	from the application's failure to limit the size of usernames, enabling attackers to create users with excessively bulky texts in the username field. This exploit results in the user management panel becoming unresponsive, preventing administrators from performing critical user management actions such as editing, suspending, or deleting users. The impact of this vulnerability includes administrative paralysis, compromised security, and operational disruption, as it allows malicious users to perpetuate their presence within the system indefinitely, undermines the system's security posture, and degrades overall system performance.
CVE-2024-5213	In mintplex-labs/anything-llm versions up to and including 1.5.3, an issue was discovered where the password hash of a user is returned in the response after login (<code>POST /api/request-token</code>) and after account creations (<code>POST /api/admin/users/new</code>). This exposure occurs because the entire User object, including the bcrypt password hash, is included in the response sent to the frontend. This practice could potentially lead to sensitive information exposure despite the use of bcrypt, a strong hashing algorithm. It is recommended not to expose any clues about passwords to the frontend.
CVE-2024-5211	A path traversal vulnerability in mintplex-labs/anything-llm allowed a manager to bypass the <code>normalizePath()</code> function, intended to defend against path traversal attacks. This vulnerability enables the manager to read, delete, or overwrite the <code>'anythingllm.db'</code> database file and other files stored in the <code>'storage'</code> directory, such as internal communication keys and <code>.env</code> secrets. Exploitation of this vulnerability could lead to application compromise, denial of service (DoS) attacks, and unauthorized admin account takeover. The issue stems from improper validation of user-supplied input in the process of setting a custom logo for the app, which can be manipulated to achieve arbitrary file read, deletion, or overwrite, and to execute a DoS attack by deleting critical files required for the application's operation.
CVE-2024-5208	An uncontrolled resource consumption vulnerability exists in the <code>'upload-link'</code> endpoint of mintplex-labs/anything-llm. This vulnerability allows attackers to cause a denial of service (DOS) by shutting down the server through sending invalid upload requests. Specifically, the server can be made to shut down by sending an empty body with a <code>'Content-Length: 0'</code> header or by sending a body with arbitrary content, such as <code>'asdasdasd'</code> , with a <code>'Content-Length: 9'</code> header. The vulnerability is reproducible by users with at least a <code>'Manager'</code> role, sending a crafted request to any workspace. This issue indicates that a previous fix was not effective in mitigating the vulnerability.
CVE-2024-4343	A Python command injection vulnerability exists in the <code>'SagemakerLLM'</code> class's <code>'complete()'</code> method within <code>./private_gpt/components/llm/custom/sagemaker.py</code> of the <code>imartinez/privategpt</code> application, versions up to and including 0.3.0. The vulnerability arises due to the use of the <code>'eval()'</code> function to parse a string received from a remote AWS SageMaker LLM endpoint into a dictionary. This method of parsing is unsafe as it can execute arbitrary Python code contained within the response. An attacker can exploit this vulnerability by manipulating the response from the AWS SageMaker LLM endpoint to include malicious Python code, leading to potential execution of arbitrary commands on the system hosting the application. The issue is fixed in version 0.6.0.

Nombre	Descripción
CVE-2024-4287	In mintplex-labs/anything-llm, a vulnerability exists due to improper input validation in the workspace update process. Specifically, the application fails to validate or format JSON data sent in an HTTP POST request to <code>/api/workspace/:workspace-slug/update</code> , allowing it to be executed as part of a database query without restrictions. This flaw enables users with a manager role to craft a request that includes nested write operations, effectively allowing them to create new Administrator accounts.
CVE-2024-4286	Mintplex-Labs' anything-llm application is vulnerable to improper neutralization of special elements used in an expression language statement, identified in the commit id <code>'57984fa85c31988b2eff429adfc654c46e0c342a'</code> . The vulnerability arises from the application's handling of user modifications by managers or admins, allowing for the modification of all existing attributes of the <code>'user'</code> database entity without proper checks or sanitization. This flaw can be exploited to delete user threads, denying users access to their previously submitted data, or to inject fake threads and/or chat history for social engineering attacks.
CVE-2024-4284	A vulnerability in mintplex-labs/anything-llm allows for a denial of service (DoS) condition through the modification of a user's <code>'id'</code> attribute to a value of 0. This issue affects the current version of the software, with the latest commit id <code>'57984fa85c31988b2eff429adfc654c46e0c342a'</code> . By exploiting this vulnerability, an attacker, with manager or admin privileges, can render a chosen account completely inaccessible. The application's mechanism for suspending accounts does not provide a means to reverse this condition through the UI, leading to uncontrolled resource consumption. The vulnerability is introduced due to the lack of input validation and sanitization in the user modification endpoint and the middleware's token validation logic. This issue has been addressed in version 1.0.0 of the software.
CVE-2024-42479	llama.cpp provides LLM inference in C/C++. The unsafe <code>'data'</code> pointer member in the <code>'rpc_tensor'</code> structure can cause arbitrary address writing. This vulnerability is fixed in b3561.
CVE-2024-42478	llama.cpp provides LLM inference in C/C++. The unsafe <code>'data'</code> pointer member in the <code>'rpc_tensor'</code> structure can cause arbitrary address reading. This vulnerability is fixed in b3561.
CVE-2024-42477	llama.cpp provides LLM inference in C/C++. The unsafe <code>'type'</code> member in the <code>'rpc_tensor'</code> structure can cause <code>'global-buffer-overflow'</code> . This vulnerability may lead to memory data leakage. The vulnerability is fixed in b3561.
CVE-2024-41950	Haystack is an end-to-end LLM framework that allows you to build applications powered by LLMs, Transformer models, vector search and more. Haystack clients that let their users create and run Pipelines from scratch are vulnerable to remote code executions. Certain Components in Haystack use Jinja2 templates, if anyone can create and render that template on the client machine they run any code. The vulnerability has been fixed with Haystack <code>'2.3.1'</code> .
CVE-2024-4181	A command injection vulnerability exists in the RunGptLLM class of the llama_index library, version 0.9.47, used by the RunGpt framework from JinaAI

Nombre	Descripción
	to connect to Language Learning Models (LLMs). The vulnerability arises from the improper use of the eval function, allowing a malicious or compromised LLM hosting provider to execute arbitrary commands on the client's machine. This issue was fixed in version 0.10.13. The exploitation of this vulnerability could lead to a hosting provider gaining full control over client machines.
CVE-2024-41130	llama.cpp provides LLM inference in C/C++. Prior to b3427, llama.cpp contains a null pointer dereference in gguf_init_from_file. This vulnerability is fixed in b3427.
CVE-2024-4084	A Server-Side Request Forgery (SSRF) vulnerability exists in the latest version of mintplex-labs/anything-llm, allowing attackers to bypass the official fix intended to restrict access to intranet IP addresses and protocols. Despite efforts to filter out intranet IP addresses starting with 192, 172, 10, and 127 through regular expressions and limit access protocols to HTTP and HTTPS, attackers can still bypass these restrictions using alternative representations of IP addresses and accessing other ports running on localhost. This vulnerability enables attackers to access any asset on the internal network, attack web services on the internal network, scan hosts on the internal network, and potentially access AWS metadata endpoints. The vulnerability is due to insufficient validation of user-supplied URLs, which can be exploited to perform SSRF attacks.
CVE-2024-3570	A stored Cross-Site Scripting (XSS) vulnerability exists in the chat functionality of the mintplex-labs/anything-llm repository, allowing attackers to execute arbitrary JavaScript in the context of a user's session. By manipulating the ChatBot responses, an attacker can inject malicious scripts to perform actions on behalf of the user, such as creating a new admin account or changing the user's password, leading to a complete takeover of the AnythingLLM application. The vulnerability stems from the improper sanitization of user and ChatBot input, specifically through the use of `dangerouslySetInnerHTML`. Successful exploitation requires convincing an admin to add a malicious LocalAI ChatBot to their AnythingLLM instance.
CVE-2024-3569	A Denial of Service (DoS) vulnerability exists in the mintplex-labs/anything-llm repository when the application is running in 'just me' mode with a password. An attacker can exploit this vulnerability by making a request to the endpoint using the [validatedRequest] middleware with a specially crafted 'Authorization:' header. This vulnerability leads to uncontrolled resource consumption, causing a DoS condition.
CVE-2024-32878	Llama.cpp is LLM inference in C/C++. There is a use of uninitialized heap variable vulnerability in gguf_init_from_file, the code will free this uninitialized variable later. In a simple POC, it will directly cause a crash. If the file is carefully constructed, it may be possible to control this uninitialized value and cause arbitrary address free problems. This may further lead to be exploited. Causes llama.cpp to crash (DoS) and may even lead to arbitrary code execution (RCE). This vulnerability has been patched in commit b2740.
CVE-2024-3283	A vulnerability in mintplex-labs/anything-llm allows users with manager roles to escalate their privileges to admin roles through a mass assignment issue. The '/admin/system-preferences' API endpoint improperly authorizes manager-level

Nombre	Descripción
	users to modify the 'multi_user_mode' system variable, enabling them to access the '/api/system/enable-multi-user' endpoint and create a new admin user. This issue results from the endpoint accepting a full JSON object in the request body without proper validation of modifiable fields, leading to unauthorized modification of system settings and subsequent privilege escalation.
CVE-2024-3279	An improper access control vulnerability exists in the mintplex-labs/anything-llm application, specifically within the import endpoint. This vulnerability allows an anonymous attacker, without an account in the application, to import their own database file, leading to the deletion or spoofing of the existing 'anythingllm.db' file. By exploiting this vulnerability, attackers can serve malicious data to users or collect information about them. The vulnerability stems from the application's failure to properly restrict access to the data-import functionality, allowing unauthorized database manipulation.
CVE-2024-3271	A command injection vulnerability exists in the run-llama/llama_index repository, specifically within the safe_eval function. Attackers can bypass the intended security mechanism, which checks for the presence of underscores in code generated by LLM, to execute arbitrary code. This is achieved by crafting input that does not contain an underscore but still results in the execution of OS commands. The vulnerability allows for remote code execution (RCE) on the server hosting the application.
CVE-2024-3166	A Cross-Site Scripting (XSS) vulnerability exists in mintplex-labs/anything-llm, affecting both the desktop application version 1.2.0 and the latest version of the web application. The vulnerability arises from the application's feature to fetch and embed content from websites into workspaces, which can be exploited to execute arbitrary JavaScript code. In the desktop application, this flaw can be escalated to Remote Code Execution (RCE) due to insecure application settings, specifically the enabling of 'nodeIntegration' and the disabling of 'contextIsolation' in Electron's webPreferences. The issue has been addressed in version 1.4.2 of the desktop application.
CVE-2024-3153	mintplex-labs/anything-llm is affected by an uncontrolled resource consumption vulnerability in its upload file endpoint, leading to a denial of service (DOS) condition. Specifically, the server can be shut down by sending an invalid upload request. An attacker with the ability to upload documents can exploit this vulnerability to cause a DOS condition by manipulating the upload request.
CVE-2024-3152	mintplex-labs/anything-llm is vulnerable to multiple security issues due to improper input validation in several endpoints. An attacker can exploit these vulnerabilities to escalate privileges from a default user role to an admin role, read and delete arbitrary files on the system, and perform Server-Side Request Forgery (SSRF) attacks. The vulnerabilities are present in the '/request-token', '/workspace/:slug/thread/:threadSlug/update', '/system/remove-logo', '/system/logo', and collector's '/process' endpoints. These issues are due to the application's failure to properly validate user input before passing it to 'prisma' functions and other critical operations. Affected versions include the latest version prior to 1.0.0.

Nombre	Descripción
CVE-2024-3150	In mintplex-labs/anything-llm, a vulnerability exists in the thread update process that allows users with Default or Manager roles to escalate their privileges to Administrator. The issue arises from improper input validation when handling HTTP POST requests to the endpoint <code>`/workspace/:slug/thread/:threadSlug/update`</code> . Specifically, the application fails to validate or check user input before passing it to the <code>`workspace_thread`</code> Prisma model for execution. This oversight allows attackers to craft a Prisma relation query operation that manipulates the <code>`users`</code> model to change a user's role to admin. Successful exploitation grants attackers the highest level of user privileges, enabling them to see and perform all actions within the system.
CVE-2024-3149	A Server-Side Request Forgery (SSRF) vulnerability exists in the upload link feature of mintplex-labs/anything-llm. This feature, intended for users with manager or admin roles, processes uploaded links through an internal Collector API using a headless browser. An attacker can exploit this by hosting a malicious website and using it to perform actions such as internal port scanning, accessing internal web applications not exposed externally, and interacting with the Collector API. This interaction can lead to unauthorized actions such as arbitrary file deletion and limited Local File Inclusion (LFI), including accessing NGINX access logs which may contain sensitive information.
CVE-2024-3110	A stored Cross-Site Scripting (XSS) vulnerability exists in the mintplex-labs/anything-llm application, affecting versions up to and including the latest before 1.0.0. The vulnerability arises from the application's failure to properly sanitize and validate user-supplied URLs before embedding them into the application UI as external links with custom icons. Specifically, the application does not prevent the inclusion of <code>`javascript:`</code> protocol payloads in URLs, which can be exploited by a user with manager role to execute arbitrary JavaScript code in the context of another user's session. This flaw can be leveraged to steal the admin's authorization token by crafting malicious URLs that, when clicked by the admin, send the token to an attacker-controlled server. The attacker can then use this token to perform unauthorized actions, escalate privileges to admin, or directly take over the admin account. The vulnerability is triggered when the malicious link is opened in a new tab using either the CTRL + left mouse button click or the mouse scroll wheel click, or in some non-updated versions of modern browsers, by directly clicking on the link.
CVE-2024-3104	A remote code execution vulnerability exists in mintplex-labs/anything-llm due to improper handling of environment variables. Attackers can exploit this vulnerability by injecting arbitrary environment variables via the <code>`POST /api/system/update-env`</code> endpoint, which allows for the execution of arbitrary code on the host running anything-llm. The vulnerability is present in the latest version of anything-llm, with the latest commit identified as <code>fde905aac1812b84066ff72e5f2f90b56d4c3a59</code> . This issue has been fixed in version 1.0.0. Successful exploitation could lead to code execution on the host, enabling attackers to read and modify data accessible to the user running the service, potentially leading to a denial of service.
CVE-2024-3102	A JSON Injection vulnerability exists in the <code>`mintplex-labs/anything-llm`</code> application, specifically within the username parameter during the login process at

Nombre	Descripción
	the <code>`/api/request-token`</code> endpoint. The vulnerability arises from improper handling of values, allowing attackers to perform brute force attacks without prior knowledge of the username. Once the password is known, attackers can conduct blind attacks to ascertain the full username, significantly compromising system security.
CVE-2024-3101	In mintplex-labs/anything-llm, an improper input validation vulnerability allows attackers to escalate privileges by deactivating 'Multi-User Mode'. By sending a specially crafted curl request with the 'multi_user_mode' parameter set to false, an attacker can deactivate 'Multi-User Mode'. This action permits the creation of a new admin user without requiring a password, leading to unauthorized administrative access.
CVE-2024-3033	An improper authorization vulnerability exists in the mintplex-labs/anything-llm application, specifically within the <code>`/api/v/`</code> endpoint and its sub-routes. This flaw allows unauthenticated users to perform destructive actions on the VectorDB, including resetting the database and deleting specific namespaces, without requiring any authorization or permissions. The issue affects all versions up to and including the latest version, with a fix introduced in version 1.0.0. Exploitation of this vulnerability can lead to complete data loss of document embeddings across all workspaces, rendering workspace chats and embeddable chat widgets non-functional. Additionally, attackers can list all namespaces, potentially exposing private workspace names.
CVE-2024-3029	In mintplex-labs/anything-llm, an attacker can exploit improper input validation by sending a malformed JSON payload to the <code>`/system/enable-multi-user`</code> endpoint. This triggers an error that is caught by a catch block, which in turn deletes all users and disables the 'multi_user_mode'. The vulnerability allows an attacker to remove all existing users and potentially create a new admin user without requiring a password, leading to unauthorized access and control over the application.
CVE-2024-3028	mintplex-labs/anything-llm is vulnerable to improper input validation, allowing attackers to read and delete arbitrary files on the server. By manipulating the 'logo_filename' parameter in the 'system-preferences' API endpoint, an attacker can construct requests to read sensitive files or the application's '.env' file, and even delete files by setting the 'logo_filename' to the path of the target file and invoking the 'remove-logo' API endpoint. This vulnerability is due to the lack of proper sanitization of user-supplied input.
CVE-2024-3025	mintplex-labs/anything-llm is vulnerable to path traversal attacks due to insufficient validation of user-supplied input in the logo filename functionality. Attackers can exploit this vulnerability by manipulating the logo filename to reference files outside of the restricted directory. This can lead to unauthorized reading or deletion of files by utilizing the <code>`/api/system/upload-logo`</code> and <code>`/api/system/logo`</code> endpoints. The issue stems from the lack of filtering or validation on the logo filename, allowing attackers to target sensitive files such as the application's database.
CVE-2024-2913	A race condition vulnerability exists in the mintplex-labs/anything-llm repository, specifically within the user invite acceptance process. Attackers can exploit this

Nombre	Descripción
	vulnerability by sending multiple concurrent requests to accept a single user invite, allowing the creation of multiple user accounts from a single invite link intended for only one user. This bypasses the intended security mechanism that restricts invite acceptance to a single user, leading to unauthorized user creation without detection in the invite tab. The issue is due to the lack of validation for concurrent requests in the backend.
CVE-2024-22422	AnythingLLM is an application that turns any document, resource, or piece of content into context that any LLM can use as references during chatting. In versions prior to commit `08d33cfd8` an unauthenticated API route (file export) can allow attacker to crash the server resulting in a denial of service attack. The `data-export` endpoint is used to export files using the filename parameter as user input. The endpoint takes the user input, filters it to avoid directory traversal attacks, fetches the file from the server, and afterwards deletes it. An attacker can trick the input filter mechanism to point to the current directory, and while attempting to delete it the server will crash as there is no error-handling wrapper around it. Moreover, the endpoint is public and does not require any form of authentication, resulting in an unauthenticated Denial of Service issue, which crashes the instance using a single HTTP packet. This issue has been addressed in commit `08d33cfd8`. Users are advised to upgrade. There are no known workarounds for this vulnerability.
CVE-2024-21552	All versions of `SuperAGI` are vulnerable to Arbitrary Code Execution due to unsafe use of the `eval` function. An attacker could induce the LLM output to exploit this vulnerability and gain arbitrary code execution on the SuperAGI application server.
CVE-2024-10131	The `add_llm` function in `llm_app.py` in infiniflow/ragflow version 0.11.0 contains a remote code execution (RCE) vulnerability. The function uses user-supplied input `req['llm_factory']` and `req['llm_name']` to dynamically instantiate classes from various model dictionaries. This approach allows an attacker to potentially execute arbitrary code due to the lack of comprehensive input validation or sanitization. An attacker could provide a malicious value for 'llm_factory' that, when used as an index to these model dictionaries, results in the execution of arbitrary code.
CVE-2024-0798	A privilege escalation vulnerability exists in mintplex-labs/anything-llm, allowing users with 'default' role to delete documents uploaded by 'admin'. Despite the intended restriction that prevents 'default' role users from deleting admin-uploaded documents, an attacker can exploit this vulnerability by sending a crafted DELETE request to the `/api/system/remove-document` endpoint. This vulnerability is due to improper access control checks, enabling unauthorized document deletion and potentially leading to loss of data integrity.
CVE-2024-0549	mintplex-labs/anything-llm is vulnerable to a relative path traversal attack, allowing unauthorized attackers with a default role account to delete files and folders within the filesystem, including critical database files such as 'anythingllm.db'. The vulnerability stems from insufficient input validation and normalization in the handling of file and folder deletion requests. Successful exploitation results in the compromise of data integrity and availability.

Nombre	Descripción
CVE-2024-0404	A mass assignment vulnerability exists in the <code>/api/invite/:code</code> endpoint of the mintplex-labs/anything-llm repository, allowing unauthorized creation of high-privileged accounts. By intercepting and modifying the HTTP request during the account creation process via an invitation link, an attacker can add a <code>role</code> property with <code>admin</code> value, thereby gaining administrative access. This issue arises due to the lack of property allowlisting and blocklisting, enabling the attacker to exploit the system and perform actions as an administrator.
CVE-2023-5833	Improper Access Control in GitHub repository mintplex-labs/anything-llm prior to 0.1.0.
CVE-2023-5832	Improper Input Validation in GitHub repository mintplex-labs/anything-llm prior to 0.1.0.
CVE-2023-4899	SQL Injection in GitHub repository mintplex-labs/anything-llm prior to 0.0.1.
CVE-2023-4898	Authentication Bypass by Primary Weakness in GitHub repository mintplex-labs/anything-llm prior to 0.0.1.
CVE-2023-4897	Relative Path Traversal in GitHub repository mintplex-labs/anything-llm prior to 0.0.1.
CVE-2023-37275	Auto-GPT is an experimental open-source application showcasing the capabilities of the GPT-4 language model. The Auto-GPT command line UI makes heavy use of color-coded print statements to signify different types of system messages to the user, including messages that are crucial for the user to review and control which commands should be executed. Before v0.4.3, it was possible for a malicious external resource (such as a website browsed by Auto-GPT) to cause misleading messages to be printed to the console by getting the LLM to regurgitate JSON encoded ANSI escape sequences (<code>\u001b[</code>). These escape sequences were JSON decoded and printed to the console as part of the model's "thinking process". The issue has been patched in release version 0.4.3.
CVE-2023-37274	Auto-GPT is an experimental open-source application showcasing the capabilities of the GPT-4 language model. When Auto-GPT is executed directly on the host system via the provided run.sh or run.bat files, custom Python code execution is sandboxed using a temporary dedicated docker container which should not have access to any files outside of the Auto-GPT workspace directory. Before v0.4.3, the <code>execute_python_code</code> command (introduced in v0.4.1) does not sanitize the <code>basename</code> arg before writing LLM-supplied code to a file with an LLM-supplied name. This allows for a path traversal attack that can overwrite any .py file outside the workspace directory by specifying a <code>basename</code> such as <code>../../main.py</code> . This can further be abused to achieve arbitrary code execution on the host running Auto-GPT by e.g. overwriting autogpt/main.py which will be executed outside of the docker environment meant to sandbox custom python code execution the next time Auto-GPT is started. The issue has been patched in version 0.4.3. As a workaround, the risk introduced by this vulnerability can be remediated by running Auto-GPT in a virtual machine, or another environment in which damage to files or corruption of the program is not a critical problem.

Nombre	Descripción
CVE-2021-30348	Improper validation of LLM utility timers availability can lead to denial of service in Snapdragon Auto, Snapdragon Compute, Snapdragon Connectivity, Snapdragon Consumer Electronics Connectivity, Snapdragon Consumer IOT, Snapdragon Industrial IOT, Snapdragon Mobile, Snapdragon Voice & Music
