Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

Aplicación móvil para evitar fraudes en criptomonedas mediante la detección de logos

Trabajo Fin de Máster

Presentado por: Sánchez Alonso, David

Director: Pedraza Gomara, Luis

Ciudad: Mondragón
Fecha: 02/03/2022

Resumen

La detección y clasificación de logotipos de criptomonedas puede ayudar a reducir algunas de las estafas asociadas a la falsificación de *Smart-Contracts*. La gran variedad de criptomonedas que existen, puede ser usada para investigar la creación de un modelo, que detecte y clasifique las criptomonedas más importantes. Este proyecto presenta el desarrollo de una aplicación móvil que converge en un análisis comparativo de algunas de las mejores arquitecturas de *Transfer Learning* (VGG16, VGG19, *InceptionResnetV2* e *InceptionV3*) del estado del arte. Se hacen pruebas con diferentes conjuntos de datos, variando el número de logotipos por clase y con variedades de criptomonedas que van desde las 90, hasta las 250. Se obtuvieron los mejores resultados con la arquitectura VGG16 al clasificar 90 clases, obteniendo un *Accuracy* del 77%. Lo que indica que aún queda mucho por investigar, en el ámbito del *Transfer Learning,* cuando el número de clases comienza a ser significativo.

Palabras Clave: Cryptocurrency, Logo-Detection, Pump-and-Dump, Shitcoin, Transfer-Learning

Abstract

Detecting and classifying cryptocurrency logos can help to reduce some of the scams associated with Smart-Contracts. The wide variety of cryptocurrencies that exist can be used to investigate the creation of a model that detects and classifies the most important cryptocurrencies. This project presents the development of a mobile application that converges on a comparative analysis of some of the best *Transfer Learning* architectures (VGG16, VGG19, *InceptionResnetV2* and *InceptionV3*) in the state of the art. Tests are done with different balanced datasets, varying the number of logos per class and with varieties of cryptocurrencies ranging from 90, to 250. After the experiments, the best results were obtained with the VGG16 architecture by classifying 90 different classes, obtaining an Accuracy of 77%. This indicates that there is still a lot of research to be done in the field of Transfer Learning, when the number of classes begins to be significant.

Keywords: Cryptocurrency, Logo-Detection, Pump-and-Dump, Shitcoin, Transfer-Learning

Índice de contenidos

1. Introducción 1
1.1. Motivación 1
1.2. Planteamiento del trabajo 1
1.3. Estructura de la memoria 2
2. Análisis del contexto 3
2.1. La tecnología *blockchain* 3
2.2. Criptomonedas y *tokens* 5
2.2.1. Contratos inteligentes (*Smart Contracts*) 8
2.2.1.1. Initial Coin Offering (ICO) 8
2.2.2. Monedas estables 9
2.2.3. Rasgos identificativos de las criptomonedas 9
2.2.3.1. Nombre y Sigla 9
2.2.3.2. Smart Contract 10
2.2.3.3. Logotipo 11
2.3. Análisis del mercado de las criptomonedas 14
2.3.1. CoinMarketCap 14
2.3.2. Tarifas en redes *Blockchain* 16
2.4. Plataformas de intercambio 17
2.4.1. Centralizadas (CEX) 17
2.4.1.1. Modo de intercambio 17
2.4.2. Descentralizadas (DEX) 19
2.4.2.1. Modo de intercambio 19
2.5. Fraudes (Scams) 23
2.5.1. Token Sniffer 24
2.5.2. *Pump-and-Dump* en grupos de *Telegram* 29
2.5.2.1. Suplantación de identidad 30

2.5.2.2. Fear Of Missing Out (FOMO) 31
2.6. Resumen del contexto 33
3. Estado del arte 35
3.1. Hitos destacados de la Inteligencia Artificial para la detección de logotipos 35
3.2. Conjuntos de datos para el entrenamiento de modelos 37
3.3. Métricas destacadas para la evaluación de los modelos 38
3.3.1. En la clasificación de imágenes 38
3.3.1.1. Exactitud (Accuracy) 38
3.3.1.2. Función de pérdida (Loss) 38
3.3.2. En la detección de objetos en imágenes 39
3.3.2.1. Average Precision (AP) 39
3.3.2.2. Mean Average Precision (mAP) 39
3.4. Soluciones actuales de la Inteligencia Artificial para el reconocimiento de logos 39
3.4.1. VGG 40
3.4.1.1. VGG-16 41
3.4.1.2. VGG-19 41
3.4.2. GoogleNet 42
3.4.3. Inception 42
3.4.3.1. InceptionV3 43
3.4.4. InceptionResnet 43
3.4.4.1. InceptionResnetV2 44
3.4.5. DenseNet 44
3.4.6. YOLO 45
3.5. FIrmcla 45
3.6. Problemas a resolver en la Inteligencia Artificial del reconocimiento de logos 45
3.6.1. Escasez en el número de conjuntos de datos 46
3.6.1.1. Limitación en el número y balanceado de instancias 46
3.6.2. Limitada cantidad de clases a predecir por conjunto de datos 46

3.6.3. Dificultad en el etiquetado de los conjuntos de datos 46
3.6.3.1. RobotFlow 47
3.6.3.2. Programas de código abierto para el etiquetado 47
3.6.3.3. Preparado manual del Dataset 48
3.6.4. Elevados tiempos de ejecución 48
3.7. Soluciones a algunas de las limitaciones 48
3.7.1. *Data Augmentation* para el aumento de conjuntos de datos 48
3.7.2. *Transfer Learning* para el elevado tiempo de entrenamiento 50
3.8. Tecnologías para la implementación 51
3.8.1. Entrenamiento de los modelos 51
3.8.1.1. Tensorflow 52
3.8.1.2. Keras 52
3.8.1.3. Pythorch 52
3.8.2. Despliegue de los modelos 53
3.8.2.1. Online: desplegados en la nube 53
3.8.2.2. Offline: desplegados en el dispositivo 54
3.8.2.2.1. PyTorch Mobile 55
3.8.2.2.2. TensorFlow Lite 55
3.9. Problemas a resolver en las tecnologías para la implementación 55
3.9.1. Altos recursos computacionales 55
3.9.2. Falta de documentación 55
3.9.3. Limitaciones en la nube 56
3.9.3.1. Dificultades al utilizar datasets de gran tamaño 56
3.9.3.2. Cortes de conexión que interrumpen el trabajo 56
3.9.3.3. Coste elevado 56
3.10. Resumen del estado del arte 57
4. Objetivos y metodología de trabajo 58
4.1. Objetivo general 58

4.2. Objetivos específicos....58
4.3. Metodología de trabajo....59
4.3.1. EDT....59
4.3.1.1. Descripción de tareas....59
4.3.1.2. Diagrama de Gantt....60
4.4. Herramientas, bibliotecas y librerías utilizadas....61
4.4.1. Herramientas utilizadas....61
4.4.1.1. Firebase....61
4.4.2. Bibliotecas y librerías....61
4.4.2.1. Keras....61
4.4.2.2. TensorFlow Lite....61
5. Identificación de requisitos....62
5.1. Requisitos Funcionales....62
5.2. Requisitos No-Funcionales....63
6. Descripción de la herramienta software desarrollada....64
6.1. Generar un *dataset* adecuado....64
6.1.1. *Data Augmentation* para aumentar la variedad del conjunto de datos....69
6.1.1.1. Data Augmentation dividida en 3 iteraciones....69
6.1.1.2. Limpieza de los conjuntos de datos....70
6.2. Evaluar las alternativas del estado del arte....70
6.2.1. Elección de los algoritmos para el *Transfer Learning*....70
6.2.1.1. Entrenamiento del modelo mediante el uso de GPU....71
6.2.2. Almacenamiento del modelo óptimo....71
6.3. Obtener un modelo para la clasificación....71
6.3.1. Clasificando 250 criptomonedas....72
6.3.1.1. Resultados utilizando el Dataset 2 (250 Clases)....72
6.3.1.2. Resultados utilizando el Dataset 3 (250 Clases)....72
6.3.2. Clasificando 100 criptomonedas....73

6.3.2.1. Actualización de los conjuntos de datos 73
6.3.2.2. Resultados utilizando el Dataset 6 (100 clases) 74
6.3.2.3. Resultados utilizando el Dataset 7 (100 clases) 74
6.3.3. Clasificando 90 criptomonedas 74
6.3.3.1. Actualización de los conjuntos de datos 75
6.3.3.2. Resultados utilizando el Dataset 8 (90 clases) 75
6.4. Desarrollar una aplicación móvil 75
6.4.1. Desarrollar las bases de datos 76
6.4.2. Conexión con Firebase para almacenar el modelo en la nube 77
6.4.3. Prueba de funcionamiento 77
6.5. Desarrollar una página web con los términos y condiciones legales 78
7. Evaluación de la aplicación móvil 79
8. Conclusiones y trabajo futuro 81
8.1. Conclusiones 81
8.2. Líneas de trabajo futuro 82
9. Bibliografía 84
1. Anexos 1
1.1. Anexo I. Mejores resultados de los entrenamientos 1
1.2. Anexo II. Capturas de pantalla de la aplicación 1
1.3. Anexo. Artículo de investigación 1

Índice de tablas

Tabla 1. Conjuntos de datos existentes para LD....37
Tabla 2. Algoritmos y soluciones del estado del arte de LD....40
Tabla 3. Arquitectura de GoogleNet....42
Tabla 4. Listado de tareas del proyecto....60
Tabla 5. Distribución de los logotipos del Dataset 1....67
Tabla 6. Distribución de los logotipos del Dataset 1 tras revisarlo....68
Tabla 7. Distribución de los 3 datasets de entrenamiento resultantes....70
Tabla 8. Resultados utilizando el Dataset 2 clasificando 250 criptomonedas....72
Tabla 9. Resultados utilizando el Dataset 3 clasificando 250 criptomonedas....72
Tabla 10. Distribución de los nuevos 3 datasets de entrenamiento resultantes....73
Tabla 11. Resultados utilizando el Dataset 6 clasificando 100 criptomonedas....74
Tabla 12. Resultados utilizando el Dataset 7 clasificando 100 criptomonedas....74
Tabla 13. Distribución de los datos de entrenamiento del nuevo Dataset 8....75
Tabla 14. Resultados utilizando el Dataset 8 clasificando 90 criptomonedas....75

Índice de figuras

Figura 1. Diseño de redes centralizadas, descentralizas y distribuidas....4
Figura 2. Arquitectura blockchain....5
Figura 3. Resultado sobre la frecuencia en la búsqueda de <<cryptocurrency>> 6
Figura 4. Comparación de frecuencia en la búsqueda <<cryptocurrency,blockchain>> 7
Figura 5. Contratos inteligentes asociados a Ethereum 10
Figura 6. Estadio del equipo de la NBA Los Angeles Lakers 11
Figura 7. Comparación de los logos de la moneda de Cardano en diferentes redes 12
Figura 8. Marketing mediante el uso del logo de Bitcoin en el metro de Londres....13
Figura 9. El logo de Bitcoin en el cartel publicitario de eToro....13
Figura 10. Página principal de CoinMarketCap....15
Figura 11. Información exhaustiva de la criptomoneda Solana 16
Figura 12. Diagrama de flujo que representa el intercambio de USDT por ETH en CEX 18
Figura 13. Interfaz de intercambio en la aplicación móvil de Binance 18
Figura 14. Descargando Monedero virtual Trust Wallet 20
Figura 15. Comprando la criptomoneda BUSD desde Trust Wallet....21
Figura 16. Conectando la cartera Trust Wallet en PancakeSwap 22
Figura 17. Intercambiando activos desde PancakeSwap (V2) 23
Figura 18. Advertencia sobre la precisión en la detección de Token Sniffer 25
Figura 19. Comparación de la frecuencia en la búsqueda de <<Token Sniffer, Shitcoin>>....25
Figura 20. Lista de los últimos supuestos nuevos fraudes en Token Sniffer 26
Figura 21. Auditoría realizada a uno de los supuestos tokens fraudulentos....27
Figura 22. Analizando un token de riesgo bajo en Token Sniffer 28
Figura 23. Posibles ejemplos de manipulación de mercado usando Telegram 29
Figura 24. Algunos ejemplos de grupos que realizan supuestamente Pumping....30
Figura 25. Grupos oficiales con sus logos y otros de dudosa procedencia: 31
Figura 26. Tweet de Samsung en el que incluye el logo de la criptomoneda MANA 32

Figura 27. Conversaciones sobre criptomonedas en grupos aleatorios de Telegram 33
Figura 28. Arquitectura de una red CNN.....36
Figura 29. Arquitectura de VGG-16 41
Figura 30. Arquitectura de VGG-19 41
Figura 31. Arquitectura de InceptionV3.....43
Figura 32. Arquitectura de InceptionResnetV2.....44
Figura 33. Arquitectura densa de DenseNet 44
Figura 34. Data Augmentation aplicado a diferentes imágenes 49
Figura 35. Ruido excesivo al aplicar Data Augmentation 49
Figura 36. Diferencias entre el enfoque tradicional y Transfer Learning 50
Figura 37. Diagrama de flujo que representa la clasificación de logos en la nube 53
Figura 38. Diagrama de flujo que representa la clasificación de logos offline 54
Figura 39. EDT del proyecto 59
Figura 40. Gantt del proyecto 60
Figura 41. Diagrama de flujo correspondiente a la aplicación propuesta 64
Figura 42. Logo de Ethereum en la web de Cryptologos 65
Figura 43. Ordenando el dataset descargado de Kaggle en Jupyter Notebook.....66
Figura 44. Creando lista con las palabras clave de cada criptomoneda.....66
Figura 45. Descargando 40 imágenes por cada criptomoneda en Jupyter Notebook.....67
Figura 46. Muestra de algunas imágenes descargadas incorrectamente.....68
Figura 47. Interfaz gráfica de la aplicación desarrollada 76
Figura 48. Prueba de funcionamiento con la criptomoneda Solana 77
Figura 49. Página web del proyecto.....78
Figura 50. Resultados de la evaluación 79

1. Introducción

En el transcurso de este primer capítulo, se mostrará la motivación que ha inspirado este proyecto y, además, el planteamiento asociado para llevarlo a cabo. Tras ello, se explicará, brevemente, cada uno de los apartados que forman la estructura de esta memoria.

1.1. Motivación

Miles de transacciones comienzan por *Twitter* o por grupos de *Telegram*, intercambiando información sobre cuáles serán las criptomonedas que les van a permitir obtener un mejor rendimiento económico. Todo ello da lugar a que la gente se lance compulsivamente a comprar cripto activos, basándose únicamente en su logotipo, teniendo como consecuencia el fraude mediante la suplantación de identidad. Algunos estudios han identificado más de 7.000 víctimas, únicamente en una de las redes *Blockchain* más conocidas, en *Ethereum*. Las pérdidas ascienden a más de 17 millones de dólares. Mientras que la adopción del dinero digital sube, los fraudes asociados aumentan a una velocidad mayor (Gao et al., 2020).

Existe la necesidad real de añadir formas de verificación para las criptomonedas, el logotipo, dada su importancia, podría ser una de ellas. Además, en lo referente a la *Inteligencia Artificial*, existe un nicho de investigación que engloba la problemática asociada al reconocimiento y clasificación de logotipos. Una gran diversidad de clases podría afectar el rendimiento de las mejores arquitecturas de *Transfer Learning* del estado del arte. Por lo tanto, las criptomonedas, teniendo en cuenta la gran diversidad de ellas que existen y sus diferentes representaciones, pueden ser la opción ideal para investigar, analizar y probar este tipo de soluciones.

1.2. Planteamiento del trabajo

Las transacciones con criptomonedas están cambiando la manera de invertir de muchas personas en todo el mundo. Tras comprobar que los métodos de verificación actuales no son suficientes para reducir los fraudes, se ha decidido implementar una nueva forma de verificación para las criptomonedas y sus *Smart Contracts* asociados. Gracias a esta nueva forma de verificación, mediante el logotipo y el uso de la *Inteligencia Artificial*, se le ofrecerá una seguridad adicional a este tipo de inversores. La aplicación móvil resultante de este proyecto permitirá, a partir del

logotipo de las criptomonedas capturadas mediante *Twitter* o grupos *Telegram*, obtener información relevante y oficial de las criptomonedas más demandadas.

Entre las arquitecturas para *Transfer Learning* analizadas, algunas de las que mejores rendimientos ofrecen son VGG16, VGG19, *InceptionResnetV2* e *InceptionV3*. Los experimentos documentados se han realizado en un grupo reducido de *datasets* y, en la mayoría de los casos, con una escasa diversidad de clases. Por lo tanto, se propone realizar diferentes pruebas con grandes conjuntos de datos, con el fin de poner realmente a prueba este tipo de arquitecturas.

1.3. Estructura de la memoria

En este **primer capítulo**, se ha presentado un resumen de lo que engloba la aplicación móvil que se ha desarrollado, la motivación, el planteamiento y la estructura que tiene el trabajo realizado.

En el **capítulo 2**, se analiza el contexto que abarca la información más actual y relevante de las criptomonedas, mostrando los fraudes y resaltando la importancia que tienen sus logotipos en este tipo de perjuicios.

En el **capítulo 3**, se analiza el estado del arte de la *Inteligencia Artificial* asociado al reconocimiento de logotipos. Se presentan las soluciones destacadas y los problemas a resolver.

En el **capítulo 4**, se muestran los objetivos y la metodología seguida para delimitar bien la posible solución al problema tratado.

En el **capítulo 5**, se identifican los requisitos necesarios para la elaboración de la aplicación, de manera que se puedan cumplir los objetivos previamente establecidos.

En el **capítulo 6**, se muestra el desarrollo de la aplicación móvil propuesta, resaltando la *Inteligencia Artificial* implementada dentro de la misma.

En el **capítulo 7**, se evalúa la solución propuesta mediante un cuestionario a un grupo de especialistas del sector.

En el **capítulo 8**, se muestran las conclusiones obtenidas de todos y cada uno de los apartados anteriores.

En el **anexo 1**, se adjuntan gráficamente los mejores resultados del entrenamiento de los modelos.

En el **anexo 2**, se adjuntan capturas de pantalla de la aplicación desarrollada.

En el **anexo 3** se adjunta el artículo de investigación correspondiente al trabajo realizado.

2. Análisis del contexto

En este apartado se va a abordar el éxito de las criptomonedas, el cual ha conseguido alzar a *Bitcoin* al tercer puesto en el ranking de divisas más grandes del mundo ordenadas por capitalización de mercado, detrás del *dólar estadounidense (USD)* y el *euro (EUR)* respectivamente, según señala *Deutsche Bank* (Arthur Herman, 2021). También se abordará la otra cara, el fraude, mostrando la relevancia que tienen los logotipos para este indeseado cometido.

Las criptomonedas han conseguido el apoyo de algunas de las personas más influyentes del mundo como *Elon Musk,* con más de 65 millones de seguidores en la red social *Twitter*, además de ser uno de los fundadores de *OpenAI*, una de las organizaciones más famosas en el ámbito de la investigación en Inteligencia Artificial (Bambrough, 2021).

Las últimas noticias oficiales, analizan la relación que existe entre el metaverso y las criptomonedas, fomentando su uso en el recién declarado metaverso de *Adidas*. (Adidas, 2021). Esta marca mundial, ha establecido la imagen de un cripto-activo, valorado en más de 300.000$, como foto de perfil en la red social *Twitter*. Todo ello ante más de 4 millones de seguidores, demostrando la importancia que tienen las imágenes y los logotipos en el mundo de las criptomonedas (Sun, 2021). Por otra parte, la exsecretaria de estado de los Estados Unidos, *Hillary Clinton,* alega que *Bitcoin* es una grave amenaza para el dólar, pudiendo destruir la misma como reserva de valor (Bambrough, 2021).

Se explicará la tecnología *blockchain* asociada a las criptodivisas, el estado actual y las amenazas que emergen entorno a ellas. Todo ello destacando la importancia de los logos asociados a las mismas. La imagen de algunos cripto-activos se está comenzando a mostrar en ciudades y en algunos de los lugares más emblemáticos del mundo, como, por ejemplo, en el propio *Time Square* en Nueva York, según informa *The New York Times* (Roose, 2021).

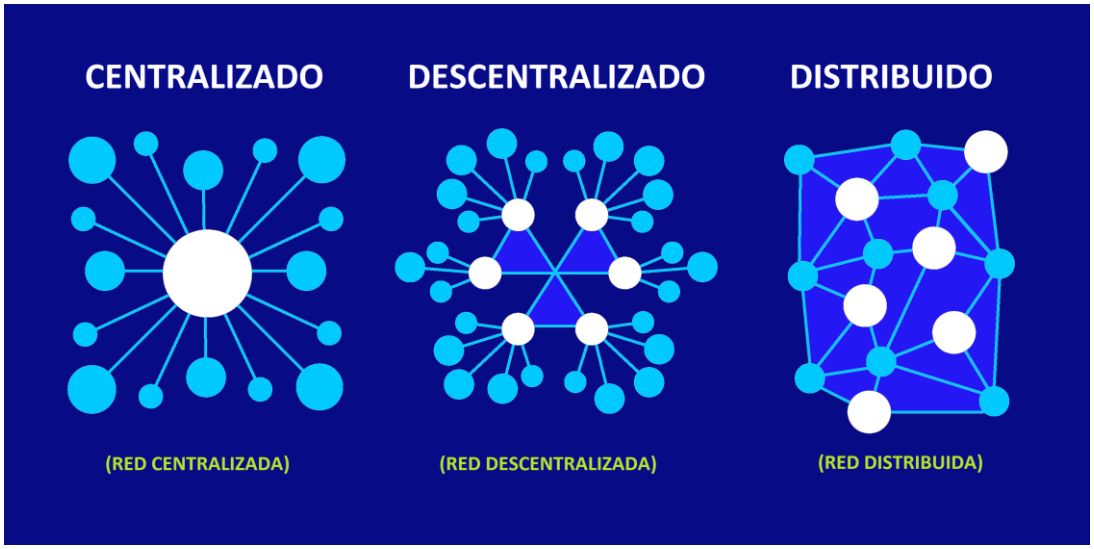
2.1. La tecnología *blockchain*

Blockchain es un tipo de tecnología capaz de aportar seguridad en su implementación y uso. Dispone de amplias aplicaciones en diferentes dominios, es una tecnología emergente para generar confianza y mejorar transparencia y trazabilidad. Este sistema está revolucionando el sector del lujo ya que permite verificar la autenticidad, reclamar la propiedad, beneficiarse de la escasez y la exclusividad de los productos (Roshanzamirª et al., 2020).

El sector alimentario es otro de los ejemplos de aplicación por cuestiones de seguridad. La constante falsificación de alimentos ha provocado un aumento en las pérdidas económicas y se ha convertido en un tema urgente para productores, investigadores y gobiernos. El seguimiento y la autentificación de la cadena de suministro es fundamental para abordar este tipo de problemas, pudiéndose solucionar mediante la autenticidad y trazabilidad que ofrece la *blockchain* (Galvez et al., 2018).

Técnicamente, la estructura es compleja, pero se podría comparar con una base de datos robusta, segura y descentralizada. La estructura de la *blockchain* almacena bloques idénticos de información a través de nodos y, cuanto más se comparten, más segura se vuelve la red. Cuando se verifican los bloques de la red, se logran ciertas características como distribución y descentralización (Cano, 2020). En la Figura 1, se observan las diferencias principales en el diseño de las redes centralizadas, descentralizadas y distribuidas. Es importante distinguir este tipo de arquitecturas cuando se habla de redes *blockchain*, ya que tienden a ser más valoradas cuanto mayor sea la descentralización y la distribución de los datos, como es el caso de las redes descentralizadas y distribuidas, las que se muestran en la segunda y tercera posición de la Figura 1 (Cano, 2020).

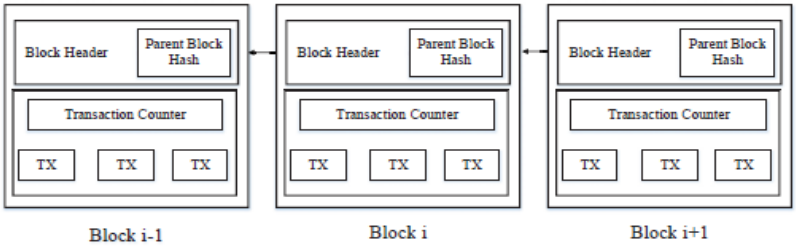
Figura 1. Diseño de redes centralizadas, descentralizas y distribuidas



Uno de los mayores beneficios de la tecnología *blockchain* se basa en que la implementación es pública. Cada participante de la red posee una completa actualización de cada transacción y de su bloque asociado (Miraz & Ali, 2018).

Cada bloque de datos lleva asociada la referencia del bloque anterior, comúnmente denominada como **función hash**, también denominada **Parent Block Hash**, tal y como se muestra en la Figura 2.

Figura 2. Arquitectura blockchain



Fuente: (Kitsantas et al., Jul 2, 2019)

Las transacciones no están asociadas a ninguna autoridad ni a reglas específicas, lo que implica una gran confianza para llegar a un consenso entre los miembros de la red, esto permite incrementar la transparencia y mitigar la manipulación de los datos. La seguridad, la transparencia y la trazabilidad son factores clave para poder establecer las criptomonedas bajo este ecosistema (Miraz & Ali, 2018).

2.2. Criptomonedas y *tokens*

Bitcoin fue creado por Satoshi Nakamoto en 2008 para construir una red de almacenamiento y transferencia descentralizada. Se creó un ecosistema que permite transacciones rápidas y seguras entre usuarios desconocidos (Kurtulmus & Daniel, 2018).

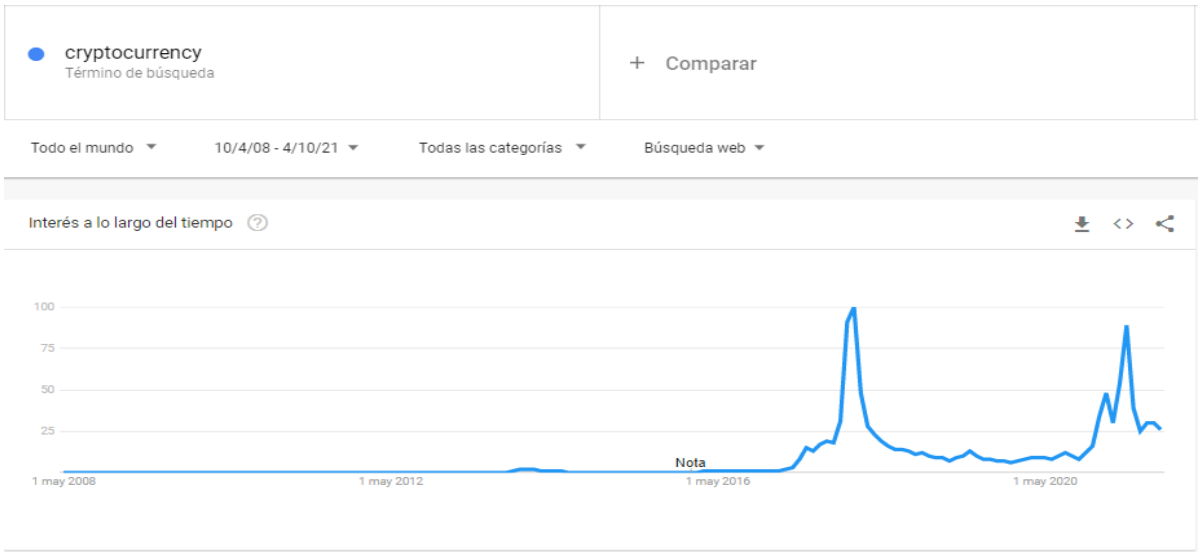
Tras la invención del *Bitcoin* se crearon más de dos mil criptomonedas denominadas **altcoins** (Aysan et al., 2021). Todas ellas se comercializan ininterrumpidamente, 24/7 en el mercado de criptomonedas (CoinMarketCap, 2021). Cada una de ellas tiene su a, su misión, su visión y sus valores. Por ejemplo, *Ethereum* es una plataforma global y de código abierto sobre la que sus inversores pueden elaborar sus contratos inteligentes para construir aplicaciones descentralizadas (Ethereum, 2021).

Cuando el precio de *Bitcoin* comenzó a incrementarse repentinamente, en el año 2017, la gente empezó a mostrar un gran interés por el mundo cripto. Se llevaron a cabo muchos estudios para

interpretar los ciclos y las posibles burbujas en las gráficas de las criptomonedas (Aysan et al., 2021).

Analizando las búsquedas relacionadas con las criptomonedas, en todo el mundo y desde el año 2008, se observan oleadas de interés durante los últimos 5 años. El crecimiento por su interés puede verse reflejado en los resultados que ofrece *Google Trends*.(Google Trends, 2021). En la Figura 3 se observan los dos grandes momentos de interés que han tenido las criptomonedas en los último 13 años. La primera gran ola hace referencia a la subida de *Bitcoin* en 2017 y, la segunda, al impacto generado por la pandemia del COVID-19 en la economía (Aysan et al., 2021).

Figura 3. Resultado sobre la frecuencia en la búsqueda de <<cryptocurrency>>



Fuente: (Google Trends, 2021)

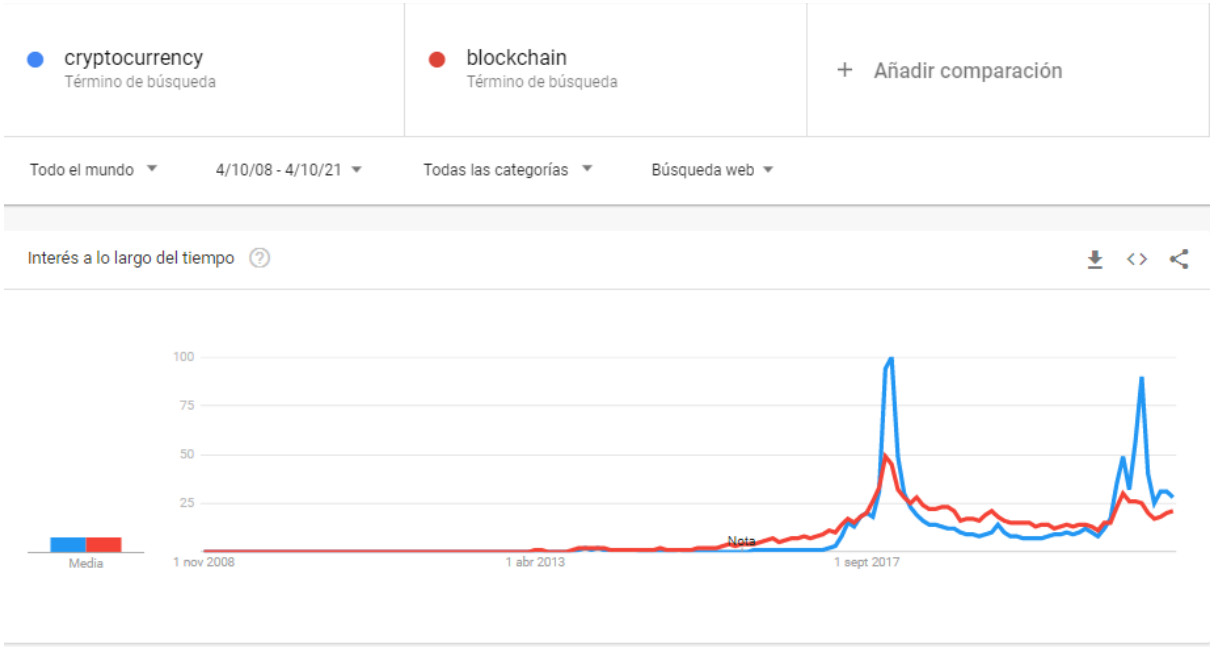
Bitcoin, Ethereum, *Ripple*, *Litecoin*, *Eos*, *BitcoinCash*, *BinanceCoin*, *Stellar* y *Tron* han sido declaradas como las criptomonedas que más tiempo han estado en el top 20 teniendo en cuenta la capitalización de mercado. A octubre de 2021 existen más de doce mil criptomonedas declaradas en el mercado, todas ellas construidas mediante la tecnología *blockchain* (CoinMarketCap, 2021).

Los *Security Tokens* (STOs) son una inversión basada en contratos *blockchain* y sujetos a la ley de valores. Interoperabilidad, liquidez de mercado y la rápida liquidación son las principales razones por las cuales, este tipo de *tokens,* son un catalizador principal para la digitalización de las finanzas (Momtaz, 2021).

Otro de los ejemplos destacados, durante estos últimos años, son los *Non-Fungible Tokens* (NFTs) que son archivos digitales únicos, relacionados directamente con el arte digital (Ante, 2021). Es un campo amplio y novedoso en el que no se va a profundizar en este trabajo.

Como las criptomonedas están implementadas mediante el sistema *blockchain*, el interés global por este tipo de tecnología comenzó antes de la subida de *Bitcoin* en 2017. Aun así, el interés por ambos términos está directamente relacionado (Google Trends, 2021). En la Figura 4 se puede observar la relación, en la frecuencia de búsqueda, de ambos términos, aun así, el término **cryptocurrency** supera en frecuencia al de *blockchain* debido al gran incremento del número de **altcoins** en los últimos años (Aysan et al., 2021).

Figura 4. Comparación de frecuencia en la búsqueda <<cryptocurrency,blockchain>>



Fuente: (Google Trends, 2021)

La diferencia que existe entre las criptomonedas y los *tokens*, técnicamente, es que las criptomonedas operan bajo su propia red *Blockchain*, como por ejemplo *Bitcoin (BTC)* o *Ethereum (ETH)*. Los *tokens*, en cambio, necesitan de una red para poder crearse, como, por ejemplo, el token *Shiba Inu*, que opera bajo la red de *Ethereum*. Aun así, es común denominar criptomonedas a los *tokens*, ya que pueden servir como método de intercambio o de pago. Por lo tanto, en este trabajo, se denominarán a ambos como criptomonedas, criptodivisas o activos digitales, ya que es un dato irrelevante para el fin del mismo. Gran parte de la información de este apartado ha sido obtenida mediante esta referencia (CoinMarketCap, 2021).

2.2.1. Contratos inteligentes (*Smart Contracts*)

Los contratos inteligentes denominados *Smart Contracts*, son bloques de código registrados en las redes *Blockchain* y que aseguran la ejecución y el cumplimiento de acuerdos entre una o más partes. Por lo tanto, un *token* es un activo digital que hace referencia a lo que se establece en el contrato de su creación (Di Angelo & Salzer, 2021).

Se hace referencia a los *tokens* como la aplicación que puede llegar a dañar la imagen del *blockchain* y de las criptomonedas. Se puede crear un *token* a partir de cualquier objeto, ya sea físico o virtual. Como activo digital, además de una criptomoneda, los *tokens* criptográficos son gestionados mediante un contrato más, comúnmente en la red de *Ethereum*.

En la transición de lo privado a lo público, cualquier empresa puede crear una cantidad de *tokens* mediante las herramientas disponibles en cada red, como por ejemplo en la red de *Ethereum*. La generación de los *tokens* se realiza mediante *Smart Contracts.* De esta manera, pueden emitir un determinado suministro de *tokens* y comercializar sus productos y servicios en función de esta nueva criptodivisa. La aceptación pública de la nueva moneda dependerá, en gran medida, de la red en la que opera la empresa (Sartipi, 2021).

2.2.1.1. *Initial Coin Offering (ICO)*

Initial Coin Offering (ICO) se denomina a la oferta inicial de una moneda, en este caso, lo que se pretende financiar es el nacimiento de una nueva criptomoneda mediante el uso de *Smart Contracts*. Es un mecanismo que sirve para financiar proyectos empresariales. Se trata de *tokens* virtuales escasos, protegidos por criptografía, que tienen un valor debido a su escasez y su demanda. La venta de este tipo de *tokens* provee de capital, y de fondos para desarrollo, a la empresa que los oferta (Momtaz, 2020).

Esta nueva forma de financiamiento de capital llamada ICO recaudó más de 30 mil millones de dólares estadounidenses entre 2016 y 2019 (Samieifar & Baur, 2021). Al mismo tiempo, el mercado cripto se ha vuelto famoso por las bromas, estafas y fraudes. Su crecimiento y su novedad han atraído el interés de los empresarios, inversores y reguladores tal y como se analiza en algunos de los estudios más recientes (Howell et al., 2020). Es por ello que se prestará especial atención a este apartado a lo largo de este trabajo.

2.2.2.Monedas estables

Las criptomonedas se caracterizan por una fuerte volatilidad y es por ello que se han creado las monedas estables denominadas *stablecoins*. Este tipo de criptodivisas son *tokens* que pueden hacer referencia al valor de monedas *fiat*, como el dólar o el euro, por ejemplo. La criptomoneda estable más conocida, haciendo referencia al dólar, es USDT.

También pueden hacer referencia al valor de bienes materiales como el oro, inmuebles u otras criptomonedas. Por otro lado, también existen monedas que no hacen referencia a ninguna otra moneda y que son controladas mediante algoritmos para que se mantengan estables, los inversores las pueden usar como refugio en momentos de gran volatilidad del mercado (Jeger et al., Nov 02, 2020).

Durante los últimos años, se ha puesto en duda el respaldo económico que pueden tener las *stablecoins*. Existen estudios que analizan el comportamiento y la fiabilidad de las mismas (Jeger et al., Nov 02, 2020).

2.2.3.Rasgos identificativos de las criptomonedas

Existen cuatro rasgos identificativos cuando se habla de criptomonedas o *tokens*, el nombre, la sigla, el *Smart Contract* y el logo. Cabe resaltar la relevancia que tienen estos rasgos ya que pueden ser utilizados por estafadores para engañar a sus víctimas, con el fin de confundirlos y defraudarlos. A continuación, se van a explicar cada uno de los rasgos resaltando la relevancia que tienen a la hora de la identificación.

2.2.3.1. *Nombre y Sigla*

El nombre normalmente caracteriza a una moneda, pero puede variar dependiendo de la red en la que se esté operando. Por ejemplo, para almacenar *Bitcoin* en una cartera que opera bajo otra red diferente, como puede ser la red *Binance Chain*, el nombre del *token* cambiaría a **Bitcoin BEP2**. De la misma manera que ocurre con las criptomonedas estables, estos *tokens* estarían respaldados bajo el mismo valor de *Bitcoin*, aunque difieran en el nombre.

Además, en cada red, se pueden crear diferentes *tokens* bajo el mismo nombre. Por lo tanto, el nombre caracteriza parcialmente a una criptodivisa, no siendo el único factor identificativo de la misma.

La sigla también es una característica importante de una moneda, pero de la misma manera que ocurre con el nombre, en cada red se podrían usar las mismas siglas haciendo referencia a diferentes criptomonedas. También puede ocurrir que se usen siglas diferentes para referirse a la misma moneda. Siguiendo con el ejemplo anterior, la sigla original de *Bitcoin* es **BTC**, pero bajo la red *Binance Chain* pasaría a ser **BTCB**.

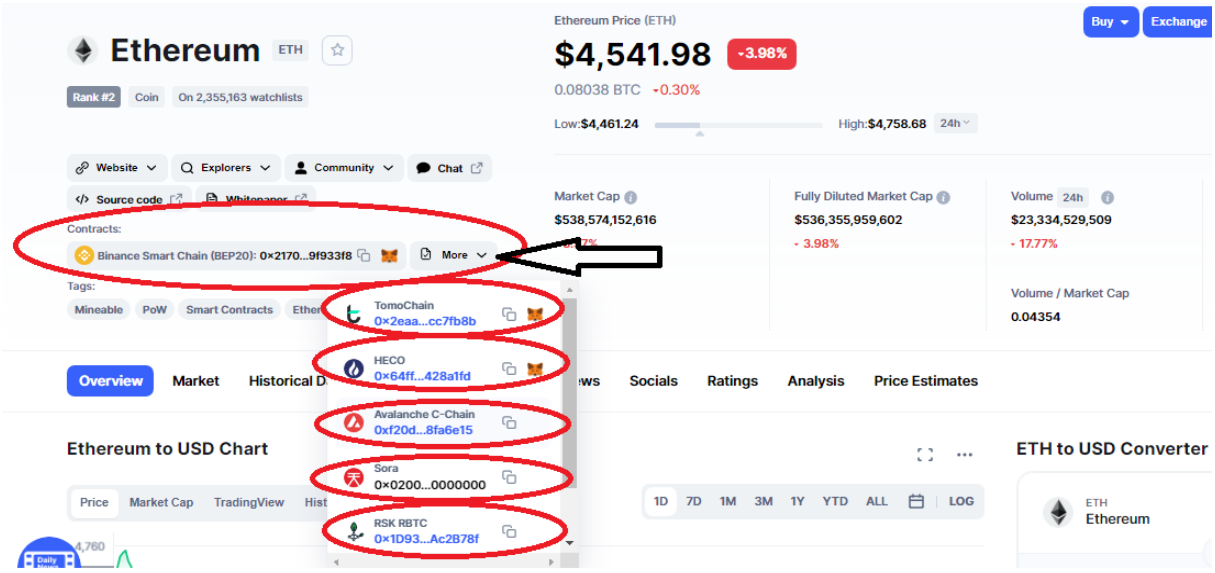
Por lo tanto, la sigla caracteriza parcialmente a una criptodivisa, no siendo el único factor identificativo de la misma.

2.2.3.2. *Smart Contract*

El contrato inteligente es uno de los aspectos más relevantes para la identificación de una criptomoneda. Es único en la red en la que se opera, es decir, es necesario comprobar a que redes pertenece dicha criptomoneda.

En la Figura 5, se puede observar cómo la criptodivisa *Ethereum* se puede obtener bajo más de cinco redes *Blockchain*, disponiendo de un *Smart Contract* diferente en cada una de las mismas. Los círculos rojos de la Figura 5, hacen referencia a los diferentes contratos, de las diferentes redes, desde los cuales se puede obtener *Ethereum*. En este caso, se puede obtener mediante las redes *TomoChain*, *HECO*, *Avalanche*, *Sora* y *RSK RBTC*, entre otras. Además, el número de redes *Blockchain* disponibles aumenta cada año, pudiéndose aumentar la lista de redes por criptomoneda al mismo tiempo.

Figura 5. Contratos inteligentes asociados a Ethereum



Fuente: (CoinMarketCap, 2021)

Por lo tanto, el contrato inteligente, aun siendo único en cada red, caracteriza parcialmente a una criptodivisa, no siendo el único factor identificativo de la misma.

2.2.3.3. *Logotipo*

El logo es uno de los rasgos identificativos que más caracteriza una criptomoneda. Es utilizado como método de marketing y es lo primero que se muestra a los futuros inversores en las distintas plataformas disponibles como *CoinMarketCap*.

Como se ha comentado al inicio de este trabajo, los logos se están comenzando a mostrar en los lugares más importantes del mundo, empiezan a ser la imagen visible de algunos estadios de los equipos más famosos de la NBA, como el de *Los Angeles Lakers* (Young, 2021). En su estadio se muestra el logo de la criptomoneda *Crypto.com Coin*, en este caso, no se utiliza el nombre completo de la moneda ni su sigla CRO, solo el logo junto a su página web, tal y como se muestra en la Figura 6.

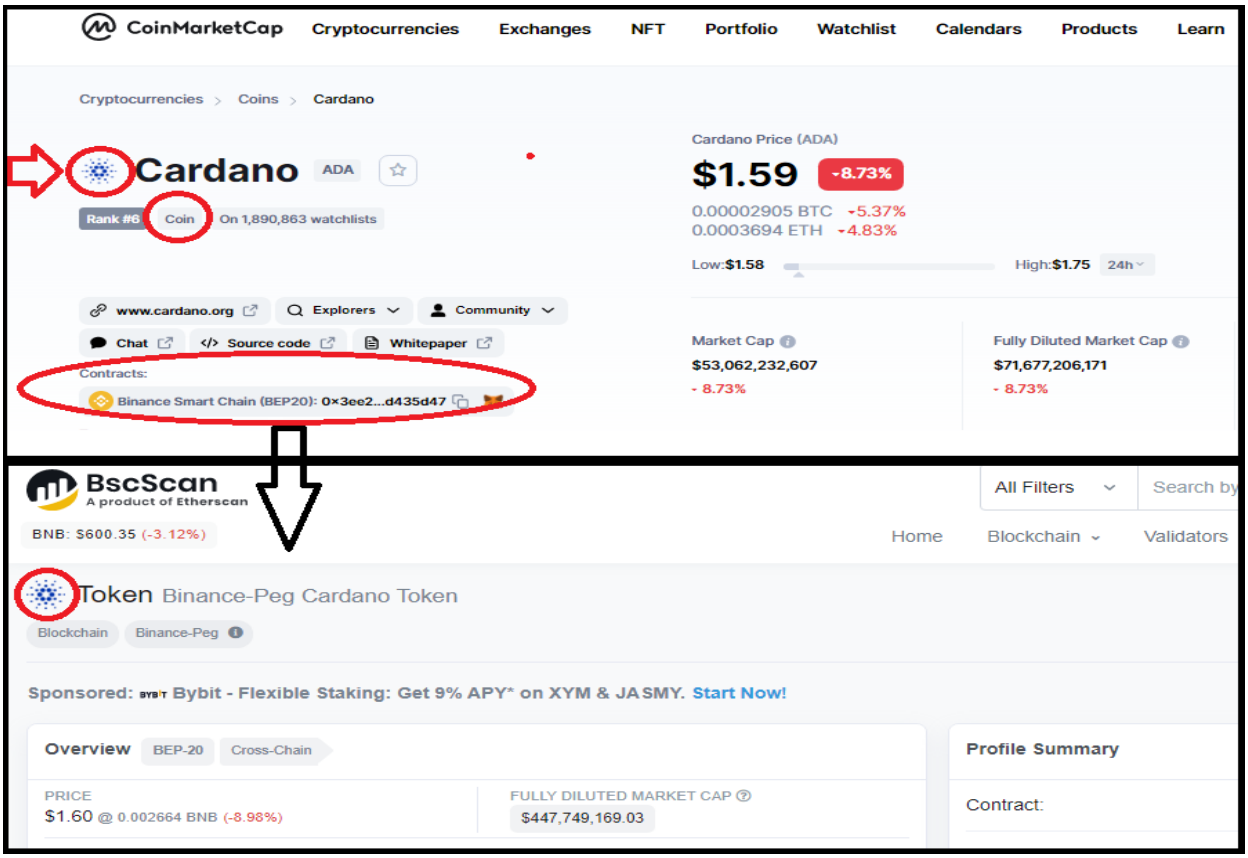
Figura 6. Estadio del equipo de la NBA Los Angeles Lakers

Fuente: (García Domínguez Rafael, 2021)

En el caso de que una criptodivisa concreta opere en diferentes redes *Blockchain*, usando un *Smart Contract* diferente en cada red, nombres diferentes y siglas diferentes, el logo suele ser similar o el mismo. En algunos casos el logo difiere en su forma, tamaño, dimensionalidad o color. Aun así, de la misma manera que los humanos reconocemos a los gatos en sus diferentes razas, también somos capaces de reconocer estos logotipos en cada una de estas variaciones. Estás variaciones dificultan el reconocimiento, pero mantienen el patrón que los hace reconocibles.

En este caso, se puede observar cómo la moneda ADA, cuando opera bajo la red *Binance Smart Chain,* cambia de nombre, pero sigue manteniendo el logo, tal y como se muestra en la Figura 7.

Figura 7. Comparación de los logos de la moneda de Cardano en diferentes redes



Fuente: (CoinMarketCap, 2021)

Además, mientras la adopción de las criptodivisas va aumentando, están empezando a vender desde criptomonedas representativas en formato físico hasta camisetas, todas ellas con sus logos impresos, todo ello como método de publicidad y marketing. Además, el marketing está llegando lugares polémicos e inimaginables, como, por ejemplo, el metro de Londres. Se ha llegado a denunciar la irresponsabilidad que supone el incitar la compra de criptomonedas en el transporte público, en este caso, mediante el uso del logo de *Bitcoin*, según informa *BBC News* (Read, 2021).

Las criptomonedas se representan y se diferencian mejor mediante sus logos, es por ello que la comunidad cripto comienza a usarlos como método de atracción hacia sus posibles futuros inversores. El logo de *Bitcoin* mostrado en la Figura 8, difiere del logo usado en otros ámbitos como en el de las criptomonedas en formato físico o el que se muestra en la página oficial de *CoinMarketCap*. Aun así, aunque visualmente se diferencien los logos de cada criptomoneda, cabe destacar la heterogeneidad que tienen los logos en sus diferentes usos, como por ejemplo en el marketing.

Figura 8. Marketing mediante el uso del logo de Bitcoin en el metro de Londres

Fuente: (Read, 2021)

Los logos mantienen ciertas características que los humanos pueden diferenciar visualmente. Como ejemplo de ello y comparación con la anterior imagen, en la Figura 9, se puede comprobar la heterogeneidad y las características del logo de *Bitcoin* en los carteles publicitarios de *eToro*.

Figura 9. El logo de Bitcoin en el cartel publicitario de eToro

Fuente: (eToro, 2021)

Los logos de las criptodivisas pueden estar impresos en papel, en camisetas, forjados en hierro, etc. Pueden diferir en todas sus versiones, aun así, la visión humana es capaz de diferenciarlos.

2.3. Análisis del mercado de las criptomonedas

Es complicado detectar si una criptomoneda o un *token* es realmente un activo fraudulento, pero existen fuentes confiables que ofrecen datos estadísticos sobre los mismos. Estas páginas web ofrecen información en tiempo real sobre el precio, capital de mercado, suministro circulante, página web… de cada activo digital que incluyen.

Se denomina **White Paper** al manifiesto que se redacta por las personas que crean la criptomoneda, concretamente, detalla todo lo relacionado con la misma. Este manifiesto contiene la tecnología usada, aplicaciones futuras, conceptos… También se dispone de la hoja de ruta denominada **Roadmap**, que se traduce en el camino ideal por el que debería transitar la criptomoneda para cumplir sus objetivos, siempre que no se trate de una estafa. Las tasas de fracaso de las ICO han sorprendido a los reguladores y varios estudios empíricos documentan evidencia de fraude (Howell et al., 2020).

Para que una criptodivisa se liste en una página como *CoinMarketCap* tiene que cumplir con una serie de requisitos como pueden ser el **White Paper**, **RoadMap**, página web, grupos de *Telegram* asociados con un alto nivel de participantes, redes sociales… Todo ello para detectar los fraudes y asegurarse de que no acaben listados.

2.3.1. CoinMarketCap

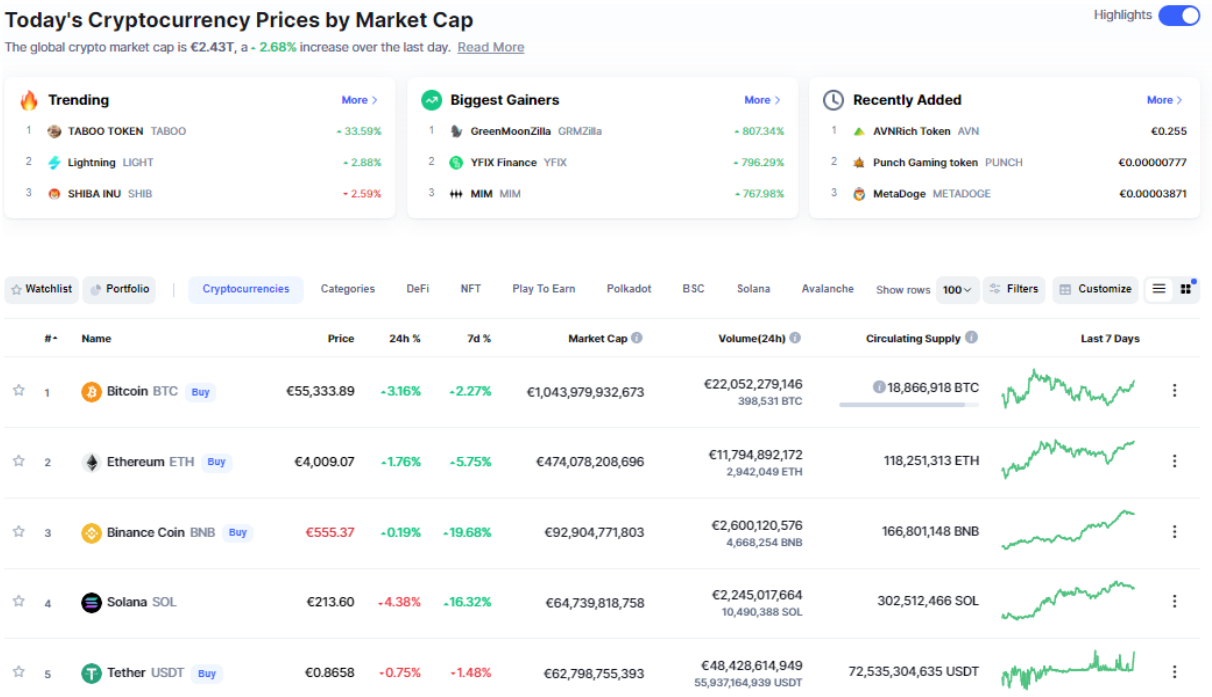
Es una de las fuentes más fiables en lo que se refiere a criptomonedas y tokens basados en *blockchain*. ofreciendo datos estadísticos de cada una de ellas (Chris Brummer, 2019). *CoinMarketCap* ofrece información sobre más de doce mil criptomonedas (CoinMarketCap, 2021).

A 8 de noviembre de 2021, *Bitcoin* es la moneda con mayor capitalización de mercado, obteniendo una suma total que asciende a más de 1 trillón de euros. La segunda moneda es *Ethereum*, seguida de *Binance Coin* y *Solana*.

Los principales datos que ofrece sobre cada criptomoneda o *token* son: ranking, logo, nombre, precio, porcentaje de ganancias/pérdidas en 24 horas, porcentaje de ganancias/pérdidas en 7 días, volumen de mercado en 24 horas, monedas en circulación y una gráfica asociada. Además de la lista principal de criptomonedas ordenada por capitalización de mercado, *CoinMarketCap* ofrece las tendencias, las mayores subidas de la jornada, las mayores pérdidas y las añadidas recientemente. Es importante saber que cuanto más arriba esté una criptomoneda en el ranking de *CoinMarketCap*, mayor capitalización de mercado tiene y, por lo tanto, mayor seguridad puede

ofrecer a sus posibles inversores, tal y como se muestra en la Figura 10. Una mayor liquidez provee de un mayor capital para desarrollo, un aspecto importante cuando se habla de *blockchain*.

Figura 10. Página principal de CoinMarketCap



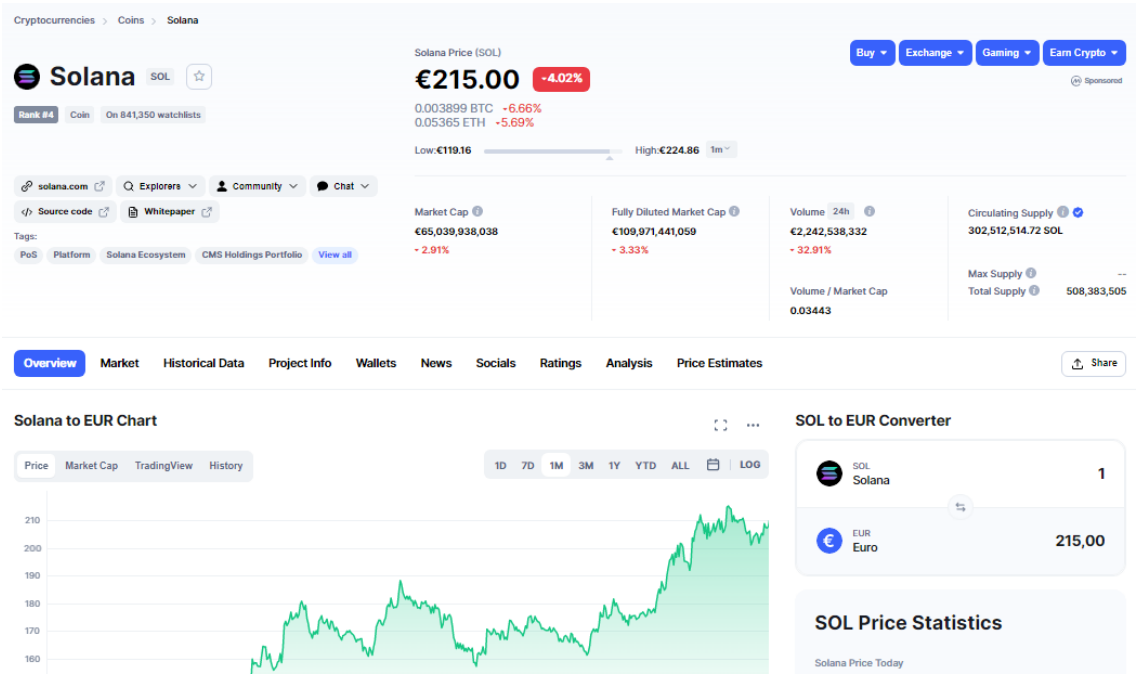
Fuente: (CoinMarketCap, 2021)

Una moneda con una baja capitalización de mercado, puede que no disponga de suficiente liquidez para realizar las transacciones correctamente y, por consecuente, disminuye la seguridad que puede dar a sus inversores.

Las criptomonedas y *tokens* usan gran parte de su capital para marketing, por lo tanto, disponer de suficiente liquidez puede marcar la diferencia entre el éxito y el fracaso.

En la web de *CoinMarketCap*, al pulsar sobre una moneda en particular, se muestran todos y cada uno de los datos necesarios para realizar una inversión con mayor conocimiento y seguridad, tal y como se muestra en la Figura 11.

Figura 11. Información exhaustiva de la criptomoneda Solana



Fuente: (CoinMarketCap, 2021)

La información genera una mayor seguridad, aun así, como casi todo en esta vida, no existe nada completamente seguro. Las inversiones en criptomonedas no están reguladas por las autoridades financieras, por lo tanto, el riesgo asociado a las mismas es muy elevado (Feinstein & Werbach, 2021).

2.3.2.Tarifas en redes *Blockchain*

Las tarifas, denominadas **gas fees**, son las encargadas del mantenimiento y de la competitividad de la red *Blockhain* en el mercado, ya que unas tarifas elevadas impiden el uso generalizado de la red.

En la red de *Bitcoin* o *Ethereum*, por ejemplo, las tarifas hacen referencia a la comisión que cobran los miembros de la red cuando verifican transacciones, es decir, existen nodos que se encargan de registrarlas mediante la resolución de unos complejos algoritmos. Cada vez que se realiza una transacción, cada nodo, comúnmente denominado *minero*, intenta resolver el algoritmo asociado a esa transacción y, finalmente, el que lo consigue es recompensado mediante la criptomoneda de la red en la que opera. Por esa misma razón existen las tarifas, son las encargas de recompensar el trabajo de los *mineros* (G. A. Pierro & H. Rocha, 2019).

Aun así, existen diferentes estrategias para compensar a los miembros de las redes, pero es un campo amplio en el que no se va a profundizar en este trabajo.

Cabe destacar que, en mercados de criptomonedas centralizadas, estos últimos, pueden añadir una tarifa adicional por su trabajo, que nada tiene que ver con la tarifa asociada al mantenimiento de las redes *Blockchain*.

2.4. Plataformas de intercambio

Existen dos tipos de plataformas para el intercambio, compra y venta de activos digitales: las plataformas centralizadas (CEX) y las descentralizadas (DEX). Se usan las plataformas descentralizadas para no transmitir los activos mediante intermediarios y eliminar las cuestiones referentes a la seguridad y a la privacidad (Xia et al., 2021).

2.4.1. Centralizadas (CEX)

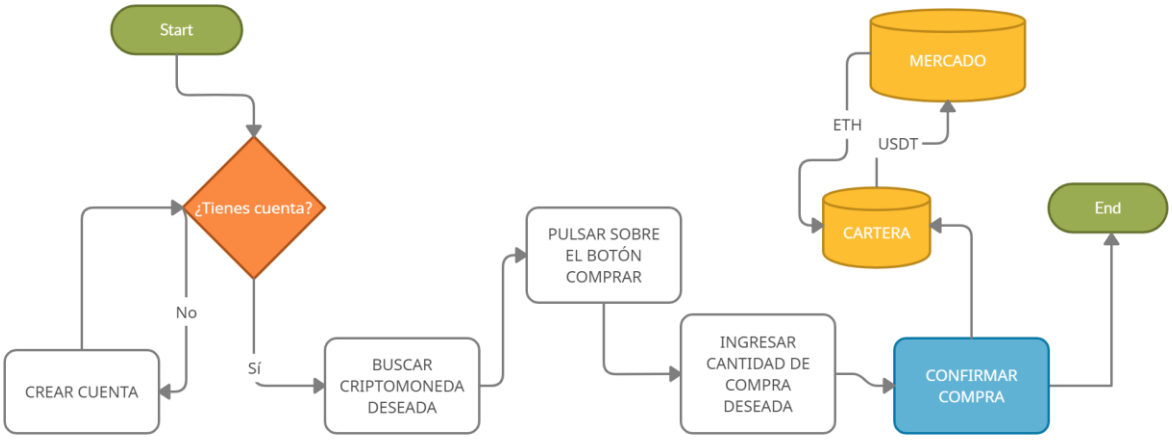
La plataforma de intercambio de *tokens* y criptomonedas *Binance* está entre las principales infraestructuras centralizadas existentes (Binance, 2021). *Binance* informó que, en el primer trimestre de 2021, el volumen de operaciones y el número de usuarios aumentaron en un 260% y 346% respectivamente (Momtaz, 2021).

Las plataformas de intercambio centralizadas se caracterizan por su facilidad en el momento de realizar la compra de criptomonedas o *tokens*. Además, como están centralizadas, requieren de un lugar físico para operar.

2.4.1.1. *Modo de intercambio*

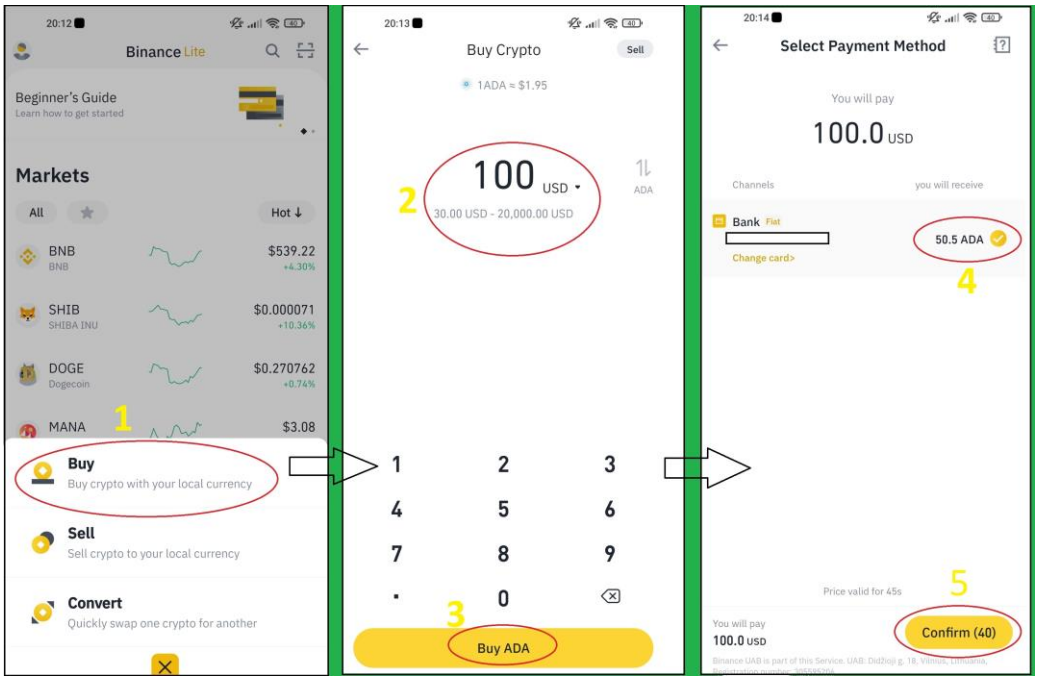
En primer lugar, hay que registrarse para obtener una cuenta y, con ello, se consigue automáticamente una cartera asociada dentro de la plataforma. Después, se comenzaría con la orden de compra mediante tarjeta de crédito o transferencia bancaria. A continuación, se buscaría la criptomoneda en el listado de monedas que ofrecen las plataformas como *Binance*. Después, se introduciría la cantidad que se quiere invertir, pudiendo elegir entre los diferentes tipos de monedas legales del mundo. Se elegiría USD, por ejemplo, y se pulsaría sobre **COMPRAR**. Tras comprobar que la cantidad de la criptomoneda que se recibe es correcta (teniendo en cuenta el valor momentáneo de la misma), se pulsaría sobre el botón **CONFIRMAR**, tal y como se muestra en Figura 12.

Figura 12. Diagrama de flujo que representa el intercambio de USDT por ETH en CEX



El proceso de intercambio que realiza una CEX en el mercado no es transparente, es decir, lo puede realizar como la empresa quiera, al ser una plataforma privada y, por ahora, no regulada. En la Figura 13, se muestra la interfaz que tiene una plataforma como *Binance*.

Figura 13. Interfaz de intercambio en la aplicación móvil de Binance



Fuente: (Binance, 2021)

Normalmente, habría que esperar unos segundos para que se realice la acción y la criptomoneda se deposite en la cartera asociada a la cuenta. En cinco sencillos pasos se realiza el intercambio correctamente.

2.4.2. Descentralizadas (DEX)

Las plataformas DEX se caracterizan por la transparencia y la accesibilidad. Cualquiera puede usar este tipo de plataformas ya que no requieren de un registro previo (Fabian Schär, 2021). Como operan en redes descentralizadas, no están adheridas a un lugar físico concreto.

Las investigaciones más recientes denominan a *Uniswap* como la DEX con mayor volumen de mercado, ya que está implementada dentro de la red de *Ethereum*, es decir, la red con mayor cantidad de *tokens* (Fabian Schär, 2021). Existen estudios que analizan el alto coste de las tarifas de *Ethereum*, la tarifa de una única transacción puede llegar a costar cientos de euros. Este es uno de los motivos por el cual *Uniswap* está perdiendo volumen de mercado (G. A. Pierro & H. Rocha, 2019).

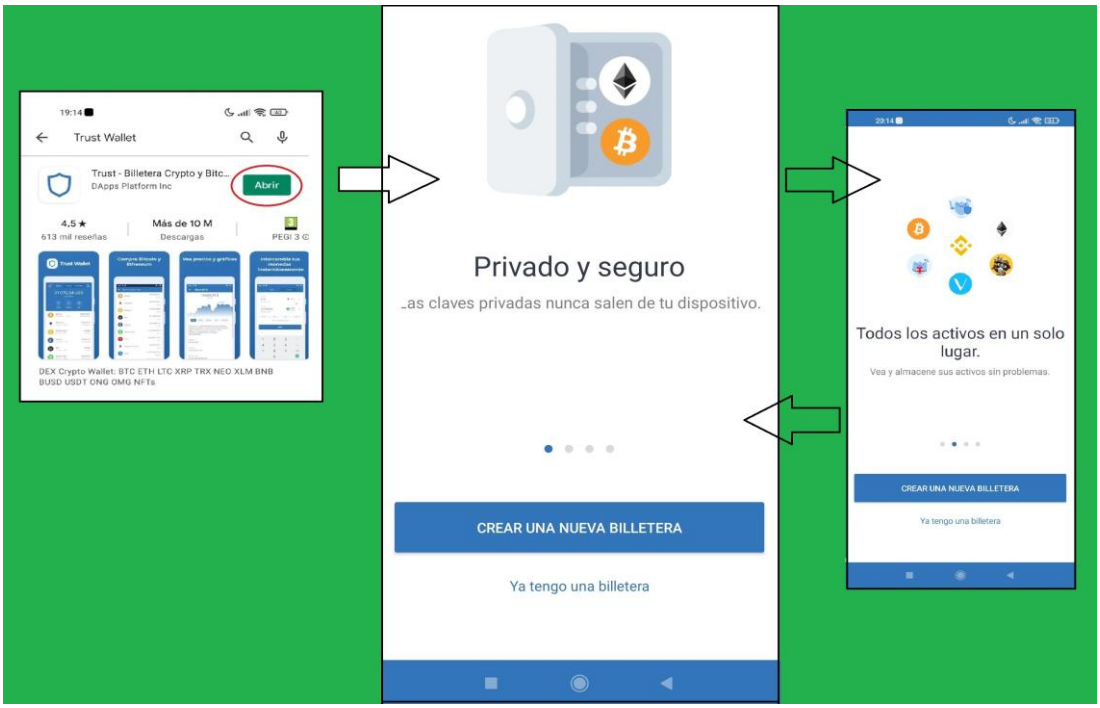
A día de hoy, 7 de noviembre de 2021, *PancakeSwap (V2)* es la DEX con mayor volumen de mercado, según los datos que ofrece *CoinMarketCap* (CoinMarketCap, 2021). *PancakeSwap (V2)* está implementada en la red *Binance Smart Chain* (BSC), propiedad de *Binance* (Binance, 2021). Las tarifas de *Binance Smart Chain* oscilan entre los céntimos de euro, mejorando drásticamente los altos costes de *Ethereum*.

El único requisito para operar en este tipo de mercados, consiste en poseer un monedero de criptomonedas, comúnmente denominado **wallet**. Este tipo de monederos virtuales no están asociados a ninguna persona en particular, el anonimato es una de sus características.

2.4.2.1. *Modo de intercambio*

En primer lugar, se debe obtener un monedero virtual, en este caso, se ha elegido el monedero descentralizado *Trust Wallet* (Trust Wallet, 2021). El monedero es compatible con la red de *Ethereum* y *Binance Smart Chain* entre muchas otras. Actualmente, *Binance* es propietario de esta billetera de código abierto. Crear el monedero virtual es tan sencillo como bajarse la aplicación móvil de *Trust Wallet,* tal y como se muestra en la Figura 14 (Binance, 2021).

Figura 14. Descargando Monedero virtual Trust Wallet



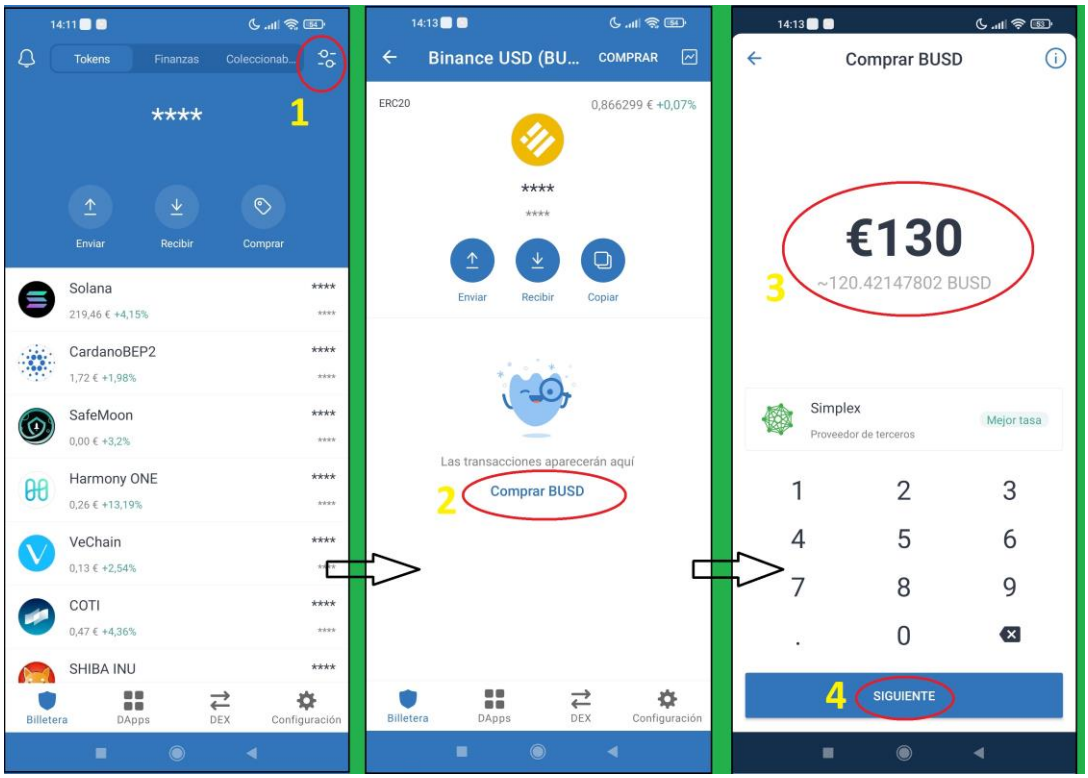
Fuente: (Google Play, 2021)

A continuación, sería conveniente comprar una de las criptomonedas estables compatibles con el monedero, por ejemplo, BUSD, es decir, la moneda estable de *Binance* (Binance, 2021).

Las criptomonedas estables no difieren en su valor, son equivalentes a la moneda que hacen referencia, es decir, BUSD siempre tendrá el mismo valor que el dólar estadounidense. Ese es el motivo principal por el cual conviene empezar adquiriéndolas, ya que el riesgo de invertir en ellas es menor.

El siguiente paso, consiste en buscar la criptomoneda BUSD entre todas las disponibles. Una vez encontrada, se pulsaría sobre **Comprar BUSD** y, a continuación, se ingresaría la cantidad que se quisiera intercambiar. Por último, se pulsaría sobre el botón **SIGUIENTE**, tal y como se muestra en la Figura 15. Tras pasar por una pasarela online de pago, se obtendrían las criptomonedas correspondientes en la dirección pública de la billetera virtual asociada.

Figura 15. Comprando la criptomoneda BUSD desde Trust Wallet



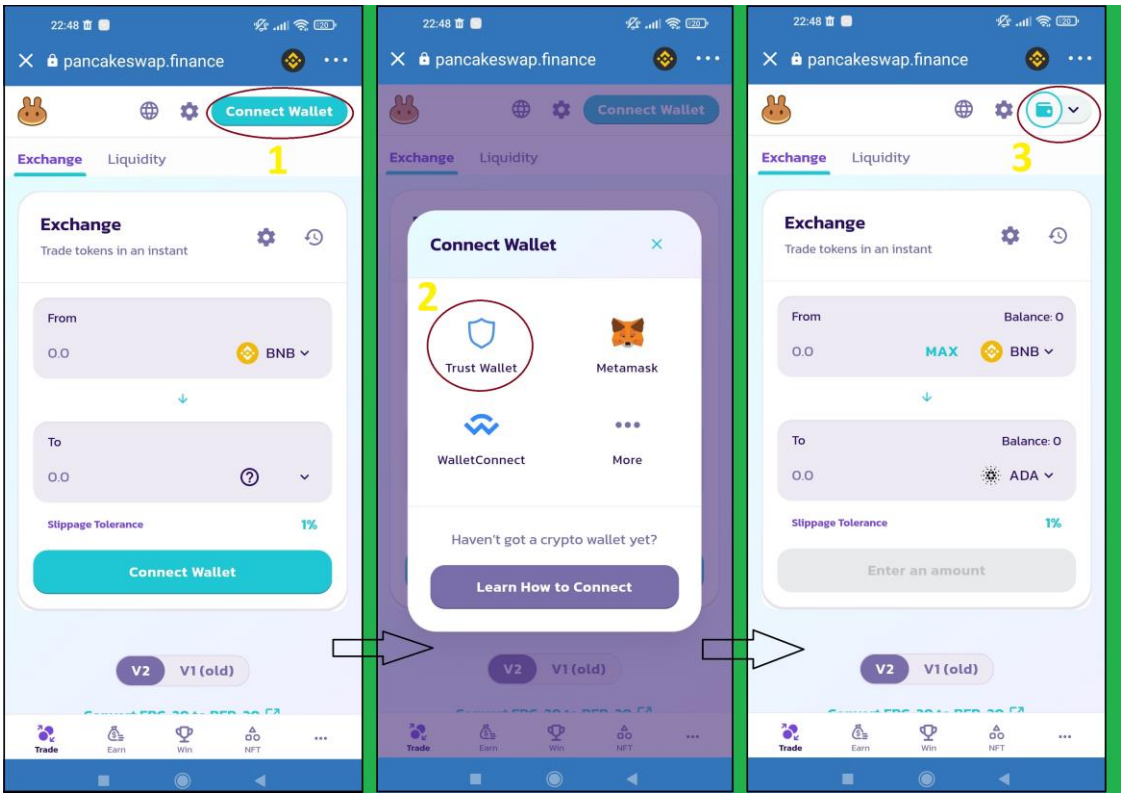
Fuente: (Trust Wallet, 2021)

A partir de este momento, el proceso de intercambio se asemeja al realizado en la plataforma centralizada, el representado en la Figura 12.

Es importante saber que, aunque la billetera no esté asociada a una persona en particular, todas y cada una de las transacciones realizadas son publicadas en la red *blockchain* en la que se opere y, además, son rastreables mediante las direcciones públicas asociadas a la billetera.

El siguiente paso, consiste en conectar la billetera virtual a una plataforma de intercambio descentralizada. En este caso la DEX elegida es *PancakeSwap (V2)*, por ser una de las más conocidas y por los motivos económicos explicados anteriormente. Se pulsa sobre el botón **Connect Wallet** para comenzar a sincronizar la cartera con la plataforma. Después, entre las carteras compatibles existentes, se pulsa sobre el icono de **Trust Wallet**. Por último, es conveniente comprobar que en la parte superior derecha se muestra la cartera virtual. En 3 sencillos pasos se podría comenzar a operar en *PancakeSwap (V2)*, tal y como se muestra en la Figura 16.

Figura 16. Conectando la cartera Trust Wallet en PancakeSwap



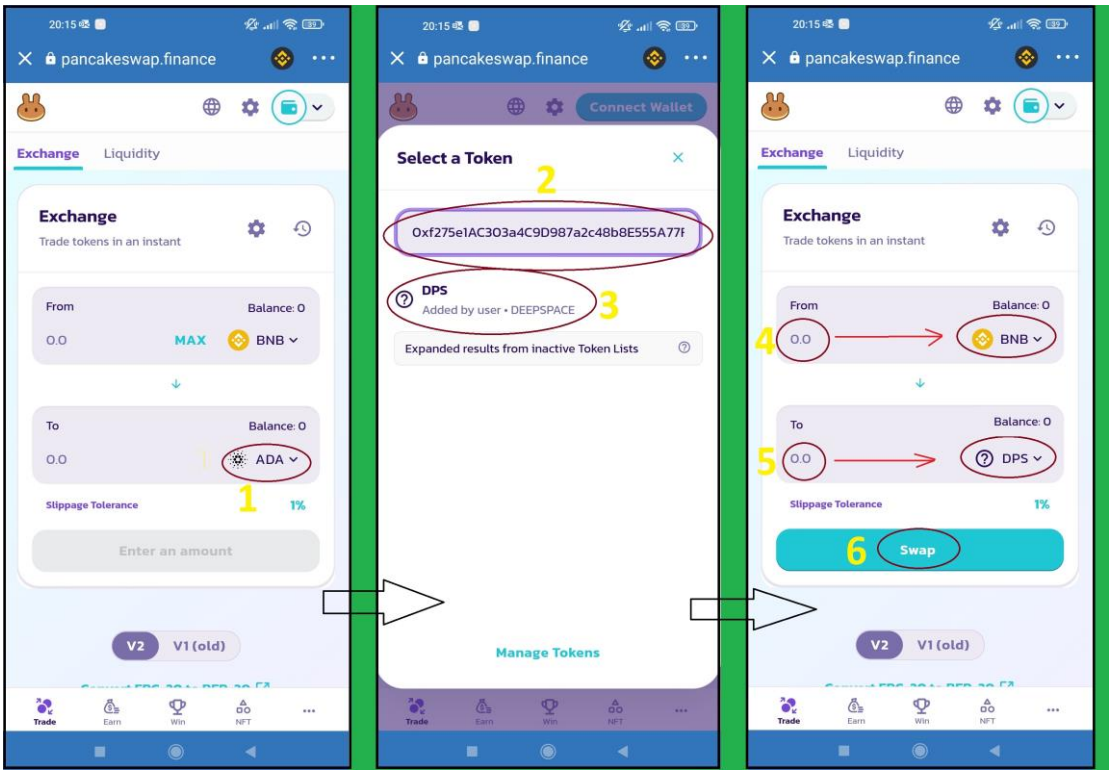
Fuente: (PancakeSwap, 2021)

A diferencia de las plataformas CEX como *Binance*, que imponen las tarifas por intercambio en la moneda que se recibe, descontando automáticamente dicha cantidad, en las plataformas DEX como *PancakeSwap (V2)*, las tarifas se abonan en la moneda de la red *Blockchain* en la que opera (PancakeSwap, 2021).

El siguiente paso consistiría en comenzar la operación de intercambio. En primer lugar, se intercambiaría BUSD por *Binance Coin* (BNB). Esto se debe a que BNB es la criptomoneda oficial que opera en *PancakeSwap*, además, es necesario disponer de ella para pagar las tarifas por cada transacción que se realiza, tal y como se ha explicado anteriormente.

Una vez se dispone de BNB, se pulsaría sobre la criptomoneda que está en el recuadro **From**, la cual hace referencia a la moneda que se quiere intercambiar, en este caso, se elegiría la moneda BNB. A continuación, se pulsaría sobre el activo que se encuentre en el recuadro **To**, y se ingresaría el contrato de la siguiente criptodivisa a comprar. Después, se pulsaría sobre la moneda que hace referencia al contrato y se introduciría la cantidad de BNB que se va a usar para el intercambio. Automáticamente, se mostraría la cantidad que se recibiría de la nueva criptomoneda y, si todo fuese correcto, se pulsaría sobre **Swap**, tal y como se muestra en la Figura 17.

Figura 17. Intercambiando activos desde PancakeSwap (V2)



Fuente: (PancakeSwap, 2021)

Tras unos segundos, los BNB y la comisión por realizar la transacción desaparecerían de la cartera y, al mismo tiempo, se recibiría la cantidad indicada de la nueva criptomoneda, finalizando con éxito la transacción realizada.

2.5. Fraudes (Scams)

Se cuestiona el dudoso valor que pueden llegar a tener los *tokens* asociados a las ICO, ya que las empresas que los ofertan no tienen un control en su evolución tecnológica o en su política monetaria. En el momento en el que se lleva a cabo la recaudación de fondos, ningún inversor es capaz de prever el precio que puede llegar a tener cada *token.* La empresa tampoco puede comprometerse con la previsión del precio, ya que estimar un precio incorrecto podría llevar a 0 el valor del mismo. Esto se debe a que la fiabilidad del *token*, en gran parte, depende de la confianza que generan los miembros del equipo.

Existen casos en los que se emiten *tokens* fraudulentamente, sin intención de crear una empresa. Incluso sabiendo que son *tokens* reales, se pone en duda su valor ya que no están directamente relacionados sobre los beneficios futuros de la empresa. (Momtaz, 2020).

Existen investigaciones que analizan el riesgo, comportamiento y los posibles modelos que pueden seguir las criptomonedas según el mercado en el que se encuentren, la capitalización de mercado que tengan y el momento en el que se sitúen. Estos modelos tienen en cuenta las estadísticas y la información de cada activo digital (Liu et al., 2019).

Se realizó un estudio sobre el porcentaje de ICOs que podrían clasificarse como fraude, y llegaron a la conclusión de que un 80% de los mismos lo eran. Después, otro estudió reveló que el porcentaje era mucho menor al previsto (Liebau & Schueffel, 2019).

Un estudio identificó más de 7.000 víctimas, las cuales sumaban un total de más de 17 millones de dólares en pérdidas, todas ellas defraudas bajo la red de *Ethereum* (Gao et al., 2020).

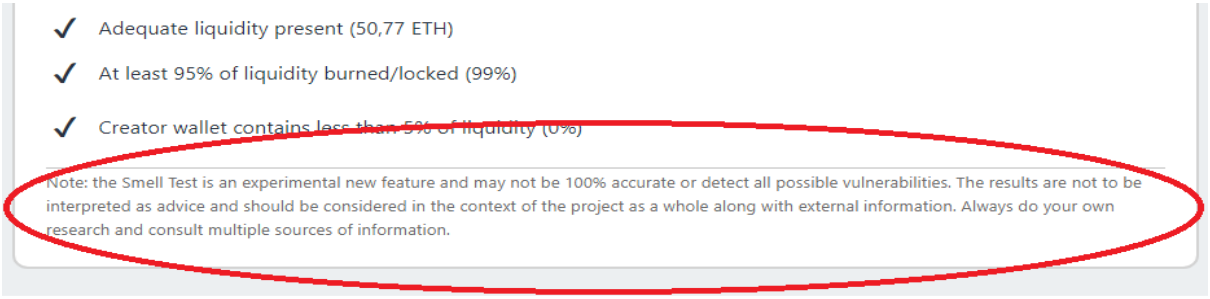
Un estudio identificó más de 10K *tokens* fraudulentos en la plataforma DEX *Uniswap*, es decir, aproximadamente el 50% de los tokens enumerados en aquel momento. A pesar de ello, los *tokens* de estafa en DEX no han sido estudiados sistemáticamente y las técnicas existentes no se pueden aplicar para identificarlos (Xia et al., 2021).

En menor o mayor medida, la existencia de activos digitales fraudulentos está demostrada. de hecho, hay estudios que siguen analizando las relaciones con los movimientos del mercado cripto para poder destaparlas(Kamps & Kleinberg, 2018).

2.5.1. Token Sniffer

Token Sniffer, es una página web que busca tokens recién listados en diferentes redes *blockchain*. A cada token le agrega una puntuación teniendo en cuenta su seguridad. Muchos de los *tokens* que se listan con la etiqueta de “riesgo bajo”, al poco tiempo, son considerados fraude por la comunidad. Por lo tanto, no es una fuente completamente fiable. La propia página web lo advierte, tal y como se muestra en la Figura 18 (TokenSniffer, 2021).

Figura 18. Advertencia sobre la precisión en la detección de Token Sniffer



Fuente: (TokenSniffer, 2021)

Shitcoin es el nombre que se usa para denominar, despectivamente, a una criptomoneda que no tiene un valor y un uso real. De hecho, al comparar los resultados en el número de búsquedas que nos ofrece *Google Trends* sobre **Token Sniffer** y **Shitcoin**, en los últimos 5 años, se observa un aumento y una relación directa entre ambos, tal y como se muestra en la Figura 19.

Figura 19. Comparación de la frecuencia en la búsqueda de <<Token Sniffer, Shitcoin>>

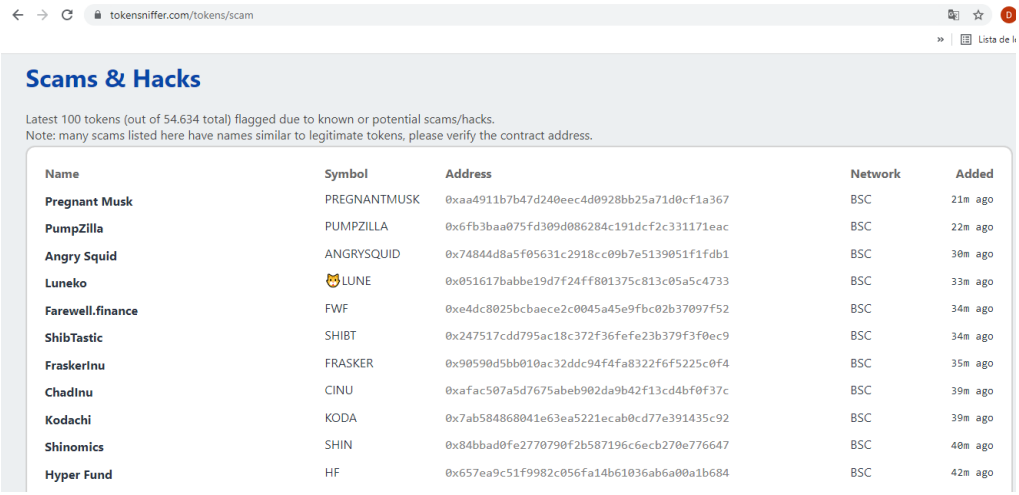
Fuente: (Google Trends, 2021)

Es común encontrar listados fraudes y **Shitcoins**, cada hora, en plataformas como *Token Sniffer*. Los estafadores se quedan con el dinero de sus víctimas aprovechando las oportunidades que puede ofrecer una nueva criptodivisa. Lo más usual es que sean *tokens* de poca capitalización de

mercado. Esta herramienta puede ser útil para comprobar la cantidad de *tokens* fraudulentos que se listan a diario (TokenSniffer, 2021).

En la Figura 20**,** se puede observar cómo en un intervalo de tiempo aproximado de 10 minutos, se pueden llegar a listar más de 7 supuestos nuevos fraudes en la red BSC. Desde la web se hace referencia a su nombre, símbolo, contrato y a la red en la que operan. Estos datos pueden ofrecer una aproximación real de la cantidad de *tokens* fraudulentos que hay en circulación.

Figura 20. Lista de los últimos supuestos nuevos fraudes en Token Sniffer



Fuente: (TokenSniffer, 2021)

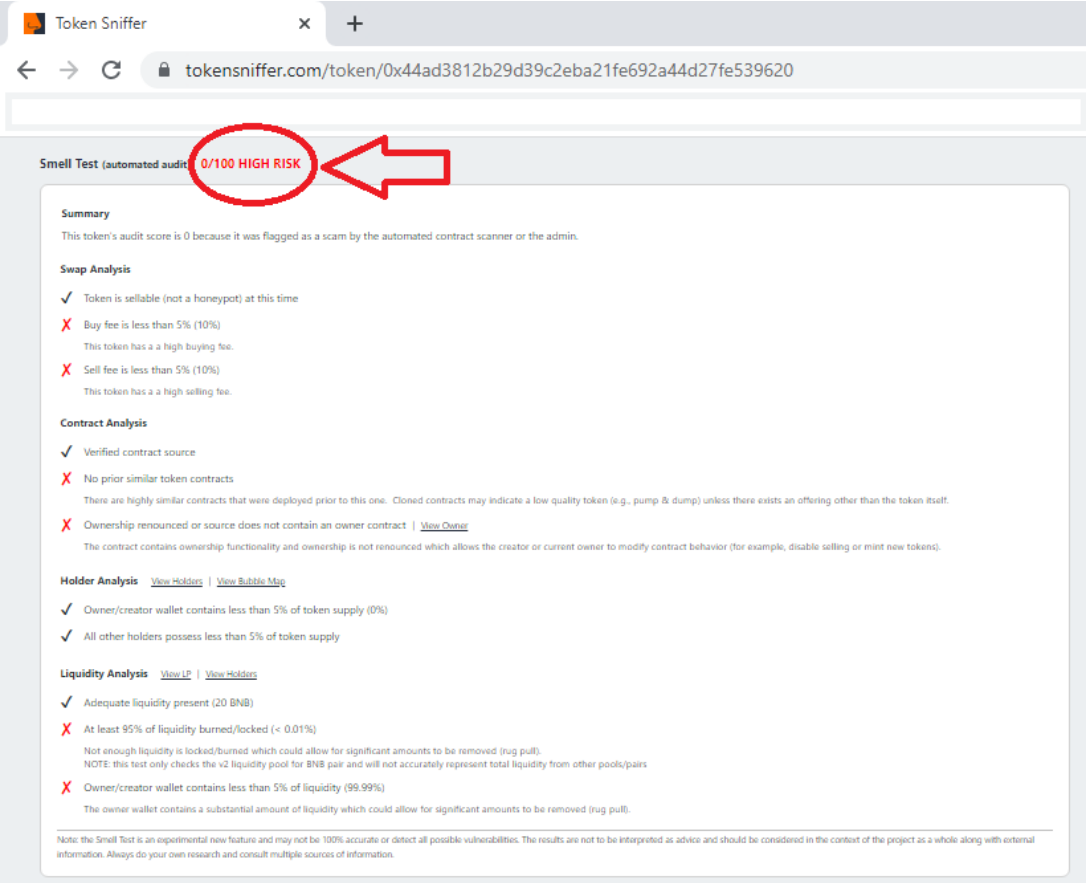
Los contratos inteligentes, mediante los cuales se distribuyen estos *tokens,* pueden tener problemas de seguridad que pueden permitir un uso no intencionado. Por esta razón, a la hora de invertir, es importante disponer de confianza suficiente sobre los lugares de donde se obtiene la información sobre los mismos (Fabian Schär, 2021).

En la página *Token Sniffer,* pulsando sobre cualquiera de los *tokens* listados, se puede comprobar el tipo de auditoría que se ha llevado a cabo y sus resultados. Entre los apartados auditados y revisados, destacan:

- Análisis del intercambio.
- Análisis del contrato
- Análisis sobre los inversores
- Análisis sobre la liquidez
- Análisis sobre contratos similares
- Reportes de la comunidad

La puntuación que recibe un *token* fraudulento es de 0 puntos, en un intervalo de 0 a 100, es decir, es un activo con un porcentaje de riesgo alto, tal y como se muestra en la Figura 21. Algunos de los motivos que llevan a fijar esta mala puntuación son: una tarifa de intercambio elevada, el dudoso contrato y que el 95% de la liquidez esté desbloqueada.

Figura 21. Auditoría realizada a uno de los supuestos tokens fraudulentos



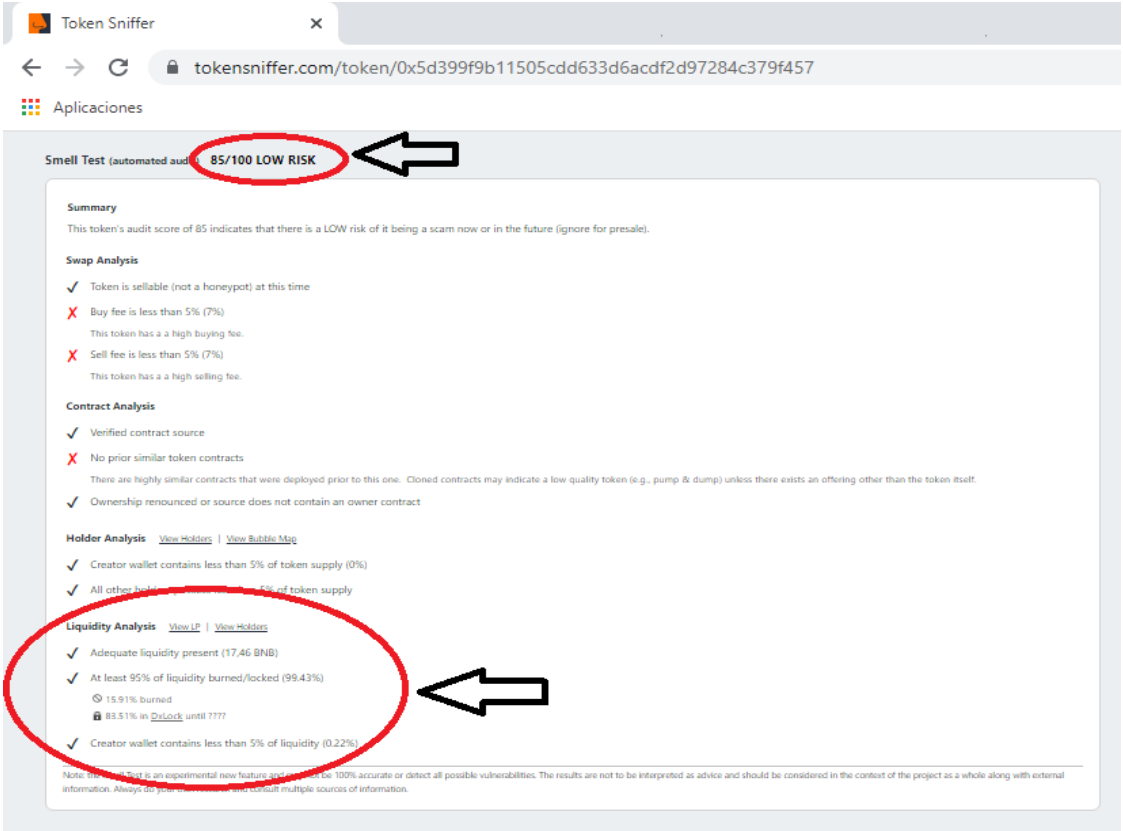
Fuente: (TokenSniffer, 2021)

El 95% de la liquidez de la criptodivisa desbloqueada, implica que el creador del *token* podría retirar en cualquier momento casi el total de la liquidez, creando la imposibilidad de vender a todos los demás inversores. Técnicamente, a esta manera de actuar que especialmente ocurre en plataformas DEX, se le denomina **Rug-Pull** (CMC Alexandria, 2021).

Buscando cualquier otro *token* con una puntuación más elevada, es decir, con un riesgo de inversión bajo, se podría comprobar que aprueba casi todos los puntos auditados. Dispone de un contrato verificado, un buen análisis de los inversores y, al menos, el 95% de la liquidez bloqueada, tal y como se muestra en la Figura 22. Existe, al menos, un detalle que puede hacer que el *token* auditado pueda convertirse en fraude. Observando el apartado sobre la liquidez, se comprueba que la liquidez está al menos al 95% bloqueada pero no se establece el tiempo que lo va a estar,

tal y como se muestra en la Figura 22. En este caso, el 15,91% de la liquidez está quemada, es decir, se ha enviado a una cartera en desuso, tal y como se muestra en la Figura 22. La acción de *quemar tokens* es utilizada para reducir la cantidad de *tokens* o monedas en circulación y, de esta manera, controlar la inflación. *Bitcoin* es la principal moneda deflacionaria y de alto valor del mundo. (Karantias et al., Febrero 10-14, 2020). *Token Sniffer* confirma, en este caso, que el 83,51% de la liquidez está bloqueada bajo contrato, pero no especifica durante cuánto tiempo, tal y como se muestra en la Figura 22. El tiempo que se mantiene bloqueada la liquidez es un dato importante, ya que bloquearla durante 1 hora, puede hacer que el *token* apruebe la auditoria durante esa hora y, después, se convierta en un posible fraude.

Figura 22. Analizando un token de riesgo bajo en Token Sniffer



Fuente: (TokenSniffer, 2021)

Normalmente, los proyectos emergentes bloquean la liquidez al menos 1 año, una forma de emitir confianza a todos sus posibles inversores. El tiempo de bloqueo es comprobable, pero requiere un cierto nivel sobre conocimientos informáticos.

Todos estos detalles son analizados por *hackers* para poder encontrar y recrear fallos de seguridad en los contratos y, de esta manera, conseguir un beneficio por ello. Por lo tanto, queda demostrado que el uso de este tipo de plataformas no garantiza una inversión realmente segura.

Una gran cantidad de fraudes operan en la red BSC, aprovechando el bajo coste de las transacciones y la cantidad de personas jóvenes, o poco experimentadas, que la pueden utilizar. Es una red accesible para casi todas las carteras, lo que la convierte en un escenario perfecto para cometer los ya referenciados delitos virtuales.

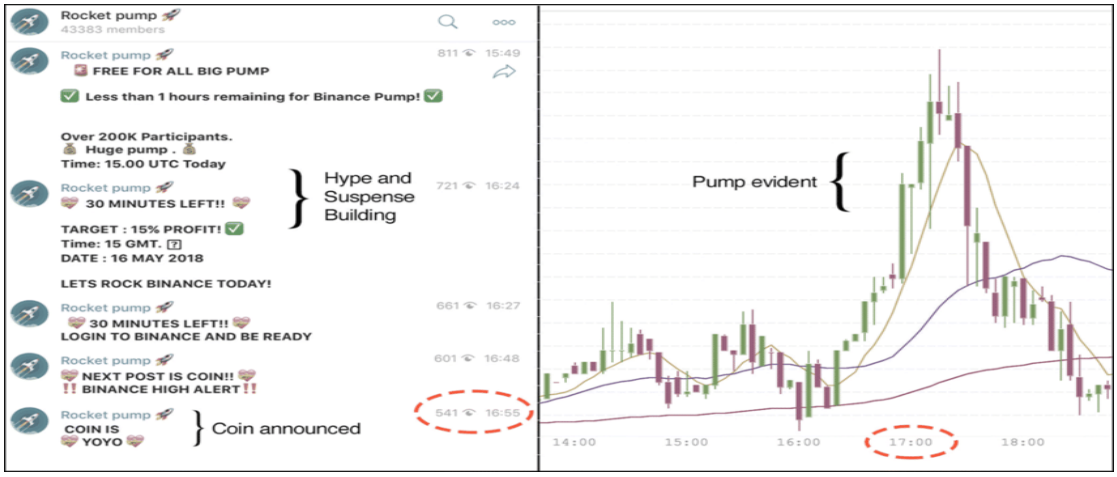
2.5.2.*Pump-and-Dump* en grupos de *Telegram*

Existen miles de grupos de *Telegram* en los cuales se construyen comunidades cripto. Estas comunidades hacen referencia a cada una de las diferentes monedas virtuales que existen, aportando información, seguridad y, además, ayudan a la toma de decisiones. Es recomendable acceder a ellos mediante los enlaces que se ofrecen en sus respectivas páginas web o en sus cuentas oficiales de *Twitter*.

Se denomina **Pump-and-Dump** a la manipulación de precios del mercado mediante la difusión de información falsa. Se están realizando estudios para comprobar su relación con posibles estafas en el mundo cripto. Además, existen estudios que demuestran una relación directa entre grupos de *Telegram* y los precios de algunas criptomonedas (Kamps & Kleinberg, 2018).

En el lado izquierdo de la Figura 23, se muestra cómo en un grupo de *Telegram,* con más de 43 mil personas participantes, se anima a invertir y realizar un **Pump-and-Dump** a una hora determinada. Recomiendan la inversión garantizando un 15% de beneficio, todo ello mediante la manipulación del mercado. En el lado derecho de la Figura 23, se muestra el supuesto hecho de la manipulación del mercado, ya que se observa un incremento importante en el valor de la moneda, en el momento justo en el que se da el aviso de compra.

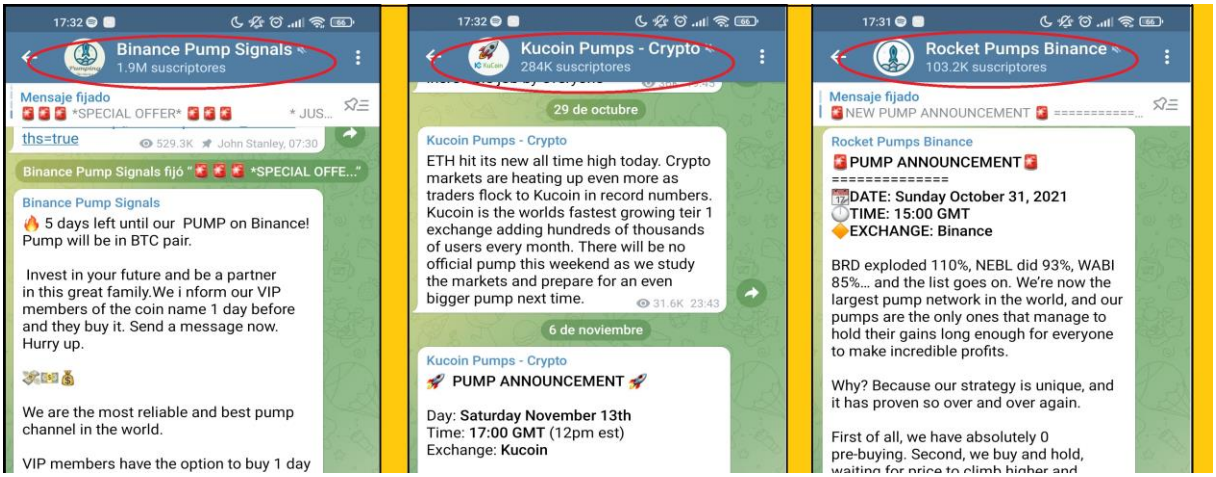
Figura 23. Posibles ejemplos de manipulación de mercado usando Telegram



Fuente: (Kamps & Kleinberg, 2018)

Existen numerosos grupos que supuestamente realizan **Pump-and-Dump** y, además, también suelen ser utilizados por terceros, para poder dar a conocer las criptomonedas emergentes. Un ejemplo de la cantidad de personas que usan este tipo de canales se muestra en la Figura 24. En esta imagen, se pueden observar tres canales de *Telegram* en los cuales se habla de realizar compras compulsivas de ciertos *tokens* en momentos determinados. En el primer grupo se encuentran 1.9 millones de personas, en el segundo 284 mil y, en el tercero, 103 mil. La suma total de personas que se encuentran dentro de estos tres grupos asciende a más de 2 millones. Y esto es solo una pequeña muestra del total de lo que se ha encontrado.

Figura 24. Algunos ejemplos de grupos que realizan supuestamente Pumping

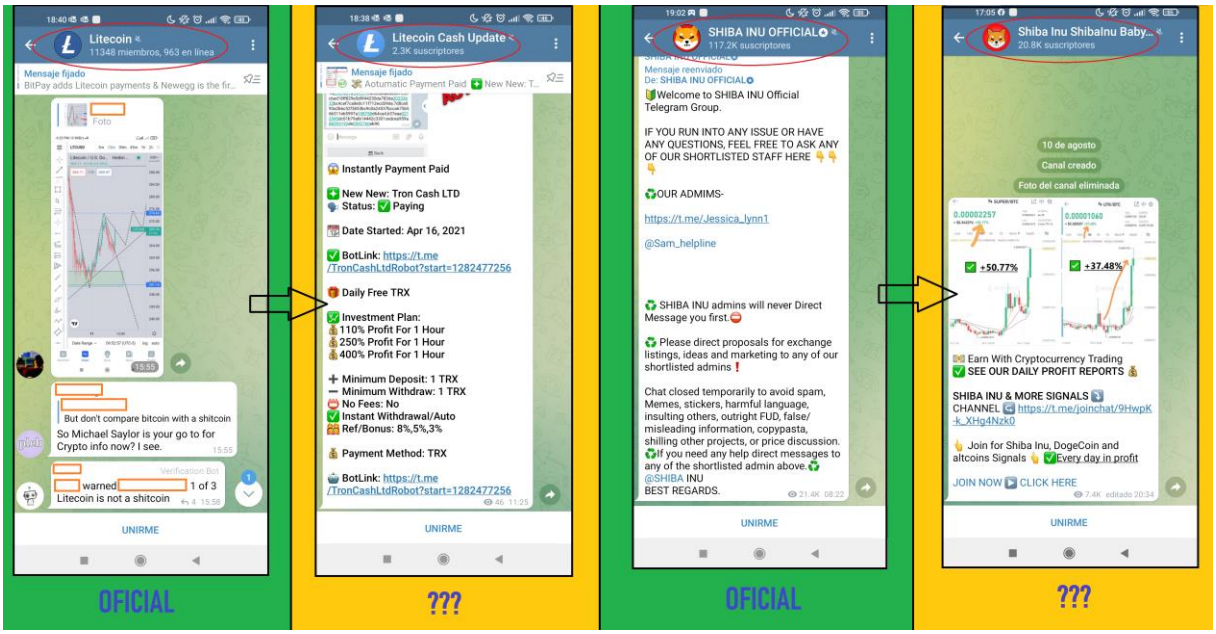


Fuente: Telegram

2.5.2.1. *Suplantación de identidad*

También existen grupos de *Telegram* en los que se pueden cometer posibles fraudes, es común encontrarse con grupos que hacen referencia a criptomonedas oficiales, pero que realmente pueden llegar a tener otro fin. Utilizan la imagen de las criptomonedas principales para construir otra comunidad paralela a la oficial, tal y como se muestra en la Figura 25. Algunos de los grupos de dudosa procedencia pueden llegar a tener más de 20 mil personas, las cuales pueden pensar que están en un grupo oficial, tal y como se muestra en la Figura 25. En la Figura 25, se observa cómo, en los grupos que no son oficiales y los cuales utilizan logos y nombres muy similares, hablan de conseguir rentabilidades altas invirtiendo en lugares de dudosa procedencia. En los grupos oficiales, en cambio, es usual que los administradores y moderadores prohíban este tipo de conductas entre los participantes. En los grupos oficiales, normalmente, no se realizan consejos sobre inversiones ni se prometen ningún tipo de rentabilidades asociadas.

Figura 25. Grupos oficiales con sus logos y otros de dudosa procedencia:



Fuente: Telegram

Cualquiera pueda crear un canal en *Telegram* y asignarle una imagen perteneciente a una criptomoneda y, como se ha comentado antes, existen más de 12.000 criptodivisas declaradas en *CoinMarketCap*. Teniendo en cuenta el crecimiento de todo el ecosistema *Blockchain* durante los últimos años, en un tiempo, podría haber más grupos falsos que reales, un problema al que todavía no se le ha encontrado solución.

2.5.2.2. *Fear Of Missing Out* (FOMO)

Fear Of Missing Out (FOMO) se denomina al miedo por quedarse fuera. Es común usar esta expresión cuando se habla de criptomonedas. En el ámbito de la *blockchain,* cada día salen decenas de nuevos proyectos basados en criptodivisas y, además, nuevas marcas se incorporan al ecosistema. Las últimas noticias de la marca coreana *Samsung*, informan sobre la decisión de incorporar galerías de cripto-activos en sus nuevos televisores. Esta marca mundial incorporará la parte más visible de la *blockchain* en su interfaz gráfica. Se podrán visitar galerías NFT desde el televisor y, por lo tanto, se verán reflejados los logos de las criptomonedas con las que se podrán adquirir, tal y como se muestra en la Figura 26 (Pérez, 2022).

Figura 26. Tweet de Samsung en el que incluye el logo de la criptomoneda MANA

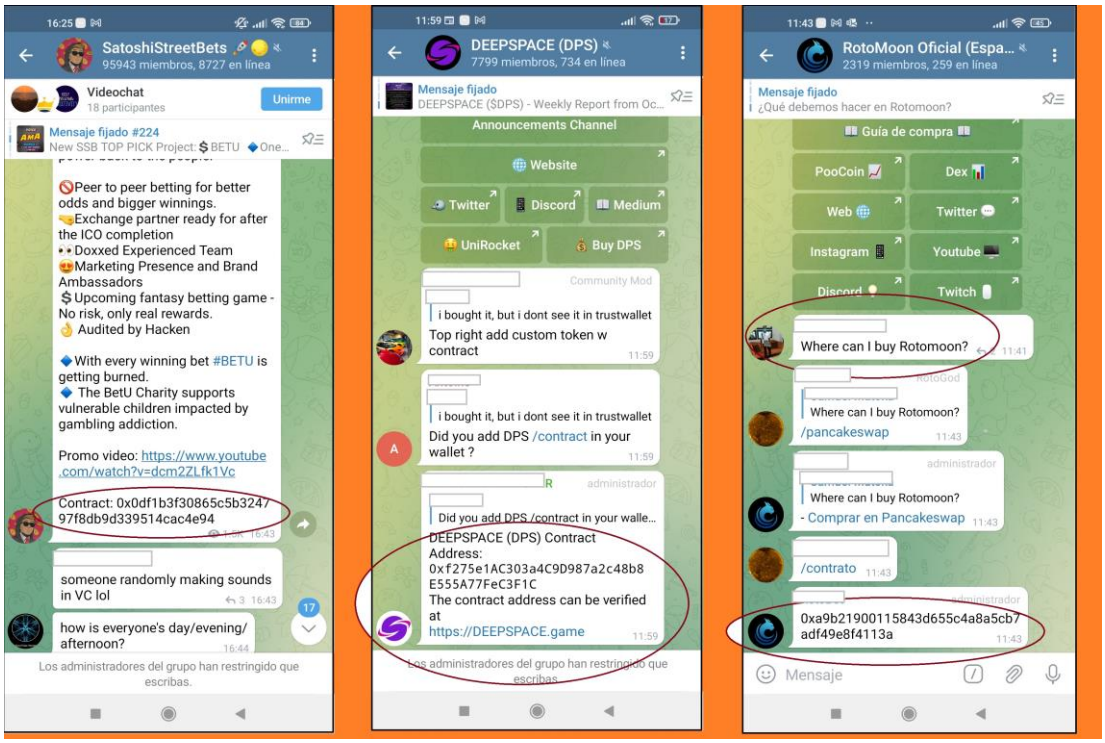
Fuente: Twitter

Muchos inversores llegan al mundo cripto con la intención de construir una fortuna en pocos días, tal y como se ha demostrado en los anteriores grupos de *Telegram*.

El problema real surge cuando se juntan supuestos grupos falsos de *Telegram*, personas con poca experiencia en inversiones, miles de contratos, criptomonedas con nombres idénticos que hacen referencia a monedas similares y, finalmente, el FOMO. En estas circunstancias, las personas invierten en cualquier cosa, a cualquier hora y sin importarles de donde obtienen la información para ello, tal y como se ha mostrado en los grupos anteriores de *Telegram*.

En la Figura 27, se observa como en grupos de *Telegram* de hasta 95 mil personas, se comparten contratos de criptomonedas sin especificar la red en la que operan, con la única confianza que dan sus logos y los nombres de usuarios que, seguramente, ni los conocen.

Figura 27. Conversaciones sobre criptomonedas en grupos aleatorios de Telegram



Fuente: Telegram

Por motivos de seguridad, como método identificativo de una criptodivisa y para realizar una inversión en la misma, se desaconseja el uso del contrato que ofrece cualquier persona por Internet o por cualquier grupo de *Telegram*.

2.6. Resumen del contexto

El número de activos digitales aumenta cada año considerablemente y, con ello, el número de personas que atrae se multiplica a una velocidad mayor.

Las criptomonedas, *tokens* y NFTs están construidos sobre la *blockchain*, es decir, sobre un ecosistema que se puede considerar seguro. El problema real que existe se centra en las personas o empresas que crean estos activos y no en la *blockchain* como tal, es decir, la base es segura pero los activos que se crean sobre ella puede que no lo sean.

Se han analizado algunos de los problemas de seguridad que pueden tener los contratos de las nuevas criptodivisas y, además, los fallos que tienen las plataformas que auditan este tipo de

activos digitales. La gran mayoría de los fraudes encontrados disponen de baja capitalización de mercado, teniendo en cuenta el *ranking* de *CoinMarketCap* (CoinMarketCap, 2021).

El uso del contrato, el nombre y la sigla, como métodos de identificación de una criptodivisa, no son suficientes, siendo estos métodos los únicos disponibles actualmente. De hecho, se ha observado como, con los métodos identificativos actuales, los fraudes y las personas afectadas siguen aumentando cada año.

Cabe destacar la importancia que tienen los logos a la hora de identificar una criptodivisa y, también, la existencia de un uso fraudulento de los mismos con el fin de atraer a posibles víctimas a inversiones de dudosa procedencia.

Se ha informado sobre la tendencia alcista que existe en torno al uso de los logos y las imágenes digitales del mundo *blockchain*, utilizadas por grandes equipos deportivos y marcas comerciales como *Los Ángeles Lakers, Adidas, eToro* y *Samsung.* Con ello, se ha mostrado la heterogeneidad que existe entre los logos en sus diferentes aplicaciones y, además, las características distinguibles que existen entre los mimos. Es decir, diferentes logos pueden caracterizar a una misma moneda si mantienen las características esenciales de las mismas.

Por lo tanto, tras mostrar el número de posibles estafas que están emergiendo, la importancia que tienen los logos y la diversidad de los lugares en los que se comienzan a encontrar, sería de gran ayuda la existencia de alguna herramienta adicional que pueda diferenciar un cripto-activo, de gran capitalización de mercado, mediante la detección de su logo, ya que podría complementar perfectamente a los métodos de identificación actuales.

Sería conveniente que, mediante alguna herramienta, los inversores pudiesen obtener los datos oficiales de las criptomonedas caracterizadas por sus logos, cuando estos aparezcan en los diversos lugares en los que se pueden encontrar, tal y como se ha mostrado durante este trabajo, ya sea en el mundo real, en Internet, en aplicaciones como *Telegram,* o hasta dentro del *Metaverso*.

Tras comprobar que una gran cantidad de inversores utilizan *Telegram* y dispositivos móviles, sería positivo encontrar alguna herramienta que, mediante el uso de la cámara o la captura de pantalla, pueda mejorar la identificación de las criptomonedas, evitando que los grandes inversores inviertan en activos fraudulentos.

3. Estado del arte

En este apartado se va a analizar el estado del arte relacionado con las principales tecnologías de LD (*Logo Detection*), comenzando con una breve introducción para acabar centrando la atención en los logotipos y analizando las carencias y dificultades que entraña este apartado de la Inteligencia Artificial. Se prestará especial atención a las investigaciones más recientes ya a las técnicas y los algoritmos con mejores resultados. Se explicarán las arquitecturas asociadas a estos algoritmos, pero sin profundizar en exceso, ya que, implementarlos individualmente, queda lejos de la perspectiva de este proyecto.

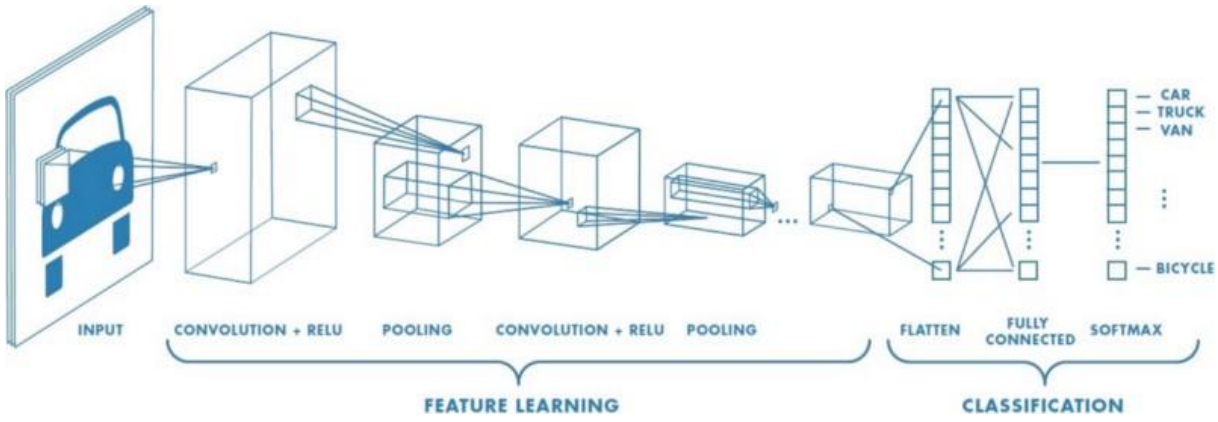
Se analizarán los conjuntos de datos más relevantes y utilizados en LD. Tras ello, se dará paso a las soluciones actuales de la Inteligencia Artificial para la resolución de esta problemática. Además, se mostrarán las limitaciones más más destacadas asociadas a la detección de logos, mostrando las tecnologías que pueden ayudar a paliarlas. Después, se explicará la técnica del *Transfer Learning*, la cual ha revolucionado el sector de la detección de logos, aumentando el rendimiento asociado a la problemática de los mismos.

3.1. Hitos destacados de la Inteligencia Artificial para la detección de logotipos

En este apartado se tratarán los hitos más relevantes que ayudaron a dar un paso adelante en el ámbito de la detección de logotipos. El descubrimiento de las *Redes Neuronales Convolucionales* (CNN) fue uno de los hitos más destacados, mejorando drásticamente el rendimiento de los modelos y, con ello, los resultados en el momento de la clasificación. Esta tecnología ha ido mejorando en ámbitos de detección de logos e imágenes, ayudando a pronosticar hasta problemas de salud en medicina (Yamashita et al., 2018).

La arquitectura de la CNN es análoga al patrón de conectividad que tienen las neuronas del cerebro humano, está inspirada en la corteza visual humana. En neurología, las neuronas individuales responden a los estímulos solo en una región restringida del campo visual, conocida como campo receptivo. La colección de estos campos cubre toda la imagen visual. Y esto es lo que ocurre análogamente en las CNN, tal y como se muestra en la en la Figura 28.

Figura 28. Arquitectura de una red CNN



Fuente: (Saha, 2018)

Uno de los principales problemas que tienen las CNN se basa en el excesivo tiempo de entrenamiento. Algunos estudios hablan de las mejoras que ofrecen las *Region Based Convolutional Networks* (R-CNN) en cuestiones de velocidad y precisión. Las R-CNN aplican un mecanismo de búsqueda selectiva para extraer regiones de interés. Aun así, estos nuevos clasificadores se basaban en 2000 propuestas de región por imagen, dando lugar a un gran tiempo de entrenamiento (Shaoqing Ren et al., 2017).

Las *Fast Region Based Convolutional Networks* (Fast R-CNN) llegaron como mejora a las R-CNN, mejorando la arquitectura y, con ello, logran una mayor velocidad de entrenamiento. Este tipo de red calcula de forma independiente las características de la red neuronal en cada una de las 2000 regiones de interés. Aun así, seguían teniendo problemas de cuello de botella durante el entrenamiento y, por esa razón, surgieron las *Faster Region Based Convolutional Networks* (Faster R-CNN) (Shaoqing Ren et al., 2017).

Las Fast R-CNN utilizan la búsqueda selectiva para para la generación de los puntos de interés, por lo tanto, las *Faster R-CNN* propusieron integrar la generación de estos puntos dentro de la propia red neuronal. Existen estudios que hablan sobre la problemática que están ejerciendo conjuntos de datos más grandes y variados. Esto da lugar a que algunos de los modelos actuales no consigan una suficiente precisión y, de esta manera, se abra un nuevo nicho a nuevas investigaciones. Estos estudios prueban las mejoras que ofrecen nuevos clasificadores, dejando a un lado las clásicas CNN y sus variantes, y comenzando a usar las *Faster R-CNN* como clasificadores *baseline* (Tüzkö et al., 2018).

3.2. Conjuntos de datos para el entrenamiento de modelos

En este apartado se van a analizar los conjuntos de datos más utilizados para la detección de logos, sus características y, con ello, sus limitaciones. Además, se explicará alguna de las técnicas actuales para la obtención de nuevos *datasets*.

Gran parte de las investigaciones de los últimos años, en la resolución de la problemática de la detección de logos, clasifican correctamente una pequeña cantidad de clases. Este tipo de investigaciones logran buenos resultados, en muchos de los casos, con una precisión cercana al 98%. El uso de *datasets* con un número limitado de clases se debe, principalmente, al límite computacional y al escaso número de conjuntos de datos etiquetados disponibles (Sahel et al., 2021)

Los conjuntos de datos más utilizados en el ámbito de LD se muestran en la Tabla 1. En esta tabla se muestra el nombre de los *datasets,* el número de clases que tienen, el número de imágenes totales, el número de objetos totales y su disponibilidad.

Tabla 1. Conjuntos de datos existentes para LD

Dataset	♯Logos	♯Images	♯Objects	Availability
BelgaLogos [21]	37	10,000	-	Yes
FlickrLogos-27 [22]	27	1,080	4,671	Yes
FlickrLogos-32 [46]	32	8,240	5,644	Yes
Top-Logo-10 [54]	10	700	-	Yes
WebLogo-2M [53]	194	1,867,177	-	Yes
QMUL-OpenLogo [55]	352	27,083	-	Yes
Logos-in-the-Wild [58]	871	11,054	32,850	Yes
Logo-2K+ [60]	2,341	167,140	-	Yes
LogoDet-3K [59]	3,000	158,652	194,261	Yes
MICC-Logos [48]	13	720	-	No
FlickrBelgaLogos [26]	34	10,000	2,695	No
Logo-18 [17]	18	8,460	16,043	No
Logo-160 [17]	160	73,414	130,608	No
Logos-32plus [1]	32	7,830	12,302	No
Video SportsLogo [28]	20	2,000	-	No
CarLogo-51 [63]	51	11,903	-	No
Open Brands [20]	1,216	1,437,812	3,113,828	No
SynthLogo [35]	604	280,000	-	No
PL2K [11]	2,000	295,814	-	No
FoodLogoDet-1500	**1,500**	**99,768**	**145,400**	**Yes**

Fuente: (Hou et al., 2021)

Cabe destacar que, de los 20 conjuntos de datos etiquetados existentes, tan solo 8, es decir, un 40% del total, contienen más de 200 logotipos diferentes a predecir. Además, de estos 8 conjuntos que tienen más de 200 clases, tan solo 4 contienen la ubicación de los objetos en cada imagen. Lo que supone una limitación a la hora de investigar en conjuntos de datos con gran variedad de clases con objeto sin ubicar. Algunos estudios comentan posibilidad y la necesidad que existe de probar y experimentar con nuevos algoritmos y conjuntos de datos (Sahel et al., 2021).

Además, entre los conjuntos de datos encontrados, no existe ninguno que contenga diversas imágenes sobre criptomonedas. La colección de imágenes más amplia encontrada, es la que se encuentra en la página web *Crypto Logos*. En esta página se encuentran cientos de imágenes de criptomonedas sin tener en cuenta su capitalización, utilidad… Además, cada criptodivisa es representada mediante un único logo, el perteneciente a *CoinMarketCap* (Crypto Logos, 2022).

3.3. Métricas destacadas para la evaluación de los modelos

Existen diversas métricas para la evaluación de los modelos en *Inteligencia Artificial*, se van a explicar algunas de las métricas más utilizadas en el ámbito de los logotipos. Las métricas más comunes varían según el tipo de modelo a evaluar. Se pueden encontrar modelos que clasifiquen imágenes o, también, modelos que detectan los logotipos en la imagen y en qué parte de la misma se encuentran. A continuación, se van a mostrar algunos ejemplos según el tipo de modelo.

3.3.1. En la clasificación de imágenes

En la clasificación de logotipos se predice el logotipo que aparece en el total de la imagen. Entre las métricas más relevantes en la clasificación de imágenes se encuentran la exactitud (*Accuracy*) y la pérdida (*Loss*) (Sahel et al., 2021).

3.3.1.1. *Exactitud (Accuracy)*

La exactitud, denominada *Accuracy*, mide el porcentaje de imágenes que el modelo ha acertado. Cuanto mayor sea el coeficiente, entre 0 y 1, mejor funcionará el modelo.

3.3.1.2. *Función de pérdida (Loss)*

La función de pérdida, denominada *Loss*, evalúa la desviación entre los valores reales de las imágenes y las predicciones realizadas por la red neuronal. Cuanto menor sea el valor de esta función, mayor eficiencia tendrá el modelo asociado a la red neuronal.

3.3.2.En la detección de objetos en imágenes

En el caso de que se quieran detectar los logotipos que aparecen en una imagen, entre las métricas más utilizadas se encuentran *Average Precission* y *Mean Average Precision*, las cuales se explicarán a continuación.

3.3.2.1.*Average Precision (AP)*

AP es el promedio de las predicciones de las clases sobre varios umbrales, es decir, es el promedio de los valores de precisión para diferentes niveles de **recall** El valor resultante comprende entre 0 y 1, siendo 1 el mejor valor de precisión. Es una métrica popular en el ámbito del reconocimiento de objetos (Sahel et al., 2021).

3.3.2.2.*Mean Average Precision (mAP)*

Comúnmente se le denomina el **precision-recall** de los objetos. Este nombre se debe, principalmente, a que es una de las métricas más populares para medir la precisión de los detectores de objetos. Mide como la red comprende lo importante y elimina la información no deseada. La precisión es más precisa cuando *mAP* es más alta (Sahel et al., 2021). Su fórmula es la siguiente:

$$\boldsymbol{mAP} = \frac{1}{\mathrm{N}}\sum_{i=1}^{N}(AP_i)$$

En la fórmula de *mAP*, *AP* hace referencia a *Average Precision*, la métrica mostrada anteriormente. Por lo tanto, *mAP* es la media de todos los valores obtenidos de *AP* respecto al total de clases (Sahel et al., 2021).

3.4. Soluciones actuales de la Inteligencia Artificial para el reconocimiento de logos

En este apartado se explicarán algunos de los algoritmos más destacados y utilizados. En primer lugar, se explicarán algunas de las métricas más relevantes para evaluar el rendimiento de los modelos que se crean a partir de los mejores algoritmos. Es por ello que se tendrán en cuenta los algoritmos que mejores resultados obtengan según las investigaciones más recientes.

Para observar las pruebas realizadas con los diferentes conjuntos de datos, en los últimos años, se dispone de una tabla con algunos de los estudios más relevantes y los modelos de resolución óptimos (Sahel et al., 2021). En la Tabla 2 se puede observar cómo, gran parte de los estudios analizados, utilizan el conjunto de datos *FlickrLogo-32*. Las métricas utilizadas para evaluarlos son *Accuracy* y la media de la precisión entre los estudios realizados, *Mean Average Performance (mAP)*. De los 14 estudios mostrados, VGG16 se posiciona como una de las soluciones con mejores resultados en *datasets* con una cantidad elevada de clases, tal y como se puede observar en la Tabla 2. Con 160 clases obtiene un *Accuracy* del 85,8% con un tiempo de entrenamiento de 1362 minutos, un tiempo aceptable teniendo en cuenta los demás resultados. Algunos entrenamientos de *R-CNN* ascienden a más de 8000 minutos. VGG16 está clasificado como una solución *Fast R-CNN***.** (Hoi et al., 2015).

Tabla 2. Algoritmos y soluciones del estado del arte de LD

Reference	Year	Approach	Dataset	Performance
[7]	2019	YOLOv3	VLD-3	76.6-98.8
[8]	2017	Faster R-CNN	FlickerLogo	0.52-0.67
[1]	2019	OSLD - SDML	BLAC and Flickr-32	84.5-90.9
[9]	2018	RetinaNet	INbreast - GURO	86-1.00
[10]	2017	Faster R-CNN	FlickrLogos	mAP 51-66%
[11]	2020	Faster R-CNN and YOLOv2	WebLogo-2M	mAP 36.8-46.9
[12]	2017	DenseNet16 -, ResNet101 -, VGG16	FlickrLogos-32 SportsLogos	0.37-0.46
[13]	2017	Scalable Logo Self Training (SLST)	WebLogo-2M	mAP 34.37%
[14]	2007	ANN and Support Vector Machine (SVM), Fisher classifier	Tobacco-800	39.3-84.2%
[5]	2015	RCNN – FRCN – SPPnet	Logos-18 test set Logos-160 test set	81.3-95.2%
[15]	2019	Faster R-CNN, SSD, YOLOv3	FlickrLogos-32, PL2K	0.565 mAP
[6]	2017	CNN	FlickrLogos-32 Logos-32plus	0.1-95.8%
[2]	2016	Fast Region-based Convolutional Networks (FRCN)	ILSVRC, FlickrLogos-32	mAP 54.5-73.74%
[16]	2018	Retina U-Net, Mask R-CNN, Faster R-CNN +, U-Faster R-CNN, + DetU-Net	LIDC-IDRI	mAP 29-50.5%
[17]	2017	Faster R-CNN	FlickrLogo-32 TopLogo-10	mAP 20.5-81.1%
[18]	2020	DenseNet-CNN	FlickrLogo-32	92.8%

Fuente: (Sahel et al., 2021)

A partir de ahora, se van a analizar algunos de los algoritmos más destacados del estado del arte para la detección de logos, añadiendo algunos que puedan resultar relevantes para esta misión.

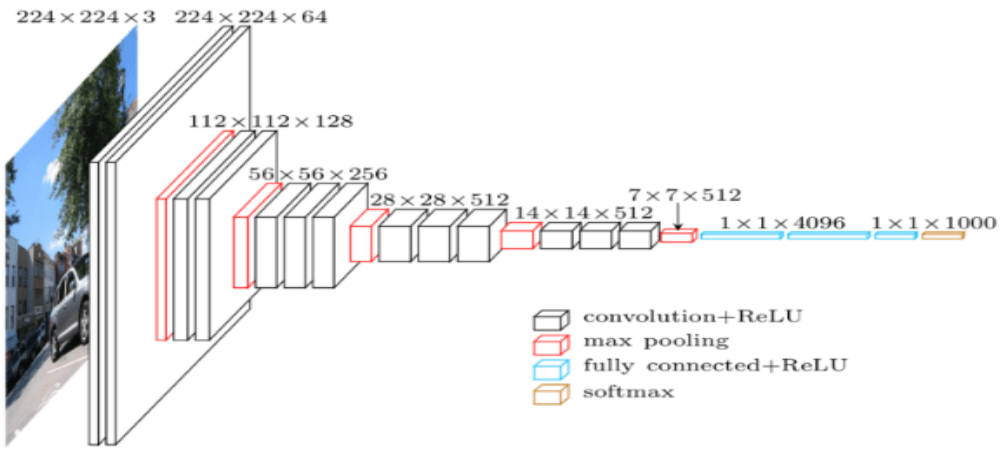
3.4.1.VGG

Esta familia de algoritmos fue utilizada en la competición *ILSVRC-2014* por el equipo de VGG. Se encuentra entre las arquitecturas CNN más simples utilizadas en las competencias de *ImageNet*. Debido a su profundidad son lentas de entrenar obteniendo modelos de gran tamaño. Gran parte de la información de este apartado está obtenido mediante esta referencia (kumar, 2021).

3.4.1.1. *VGG-16*

En su arquitectura se encuentran capas *convolucionales* activadas por la función *Relu*, capas *max pooling,* capas neuronales ocultas totalmente conectadas y, finalmente la última capa *Softmax* con las clases a predecir. La arquitectura agrega hasta 16 capas para aprender los pesos, y de ahí viene su nombre, tal y como se muestra en la Figura 29.

Figura 29. Arquitectura de VGG-16

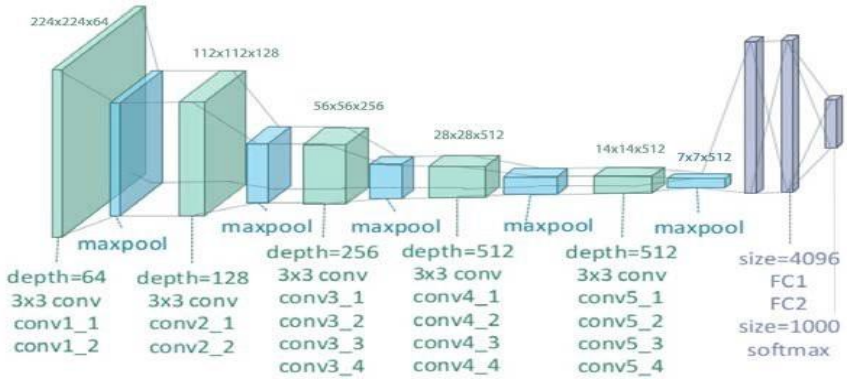


Fuente: (kumar, 2021)

3.4.1.2. *VGG-19*

La arquitectura de VGG-19 es muy similar a la de la VGG-16. La diferencia principal de VGG-19 radica en el número de capas, agregando hasta 19 capas para aprender los pesos, tal y como se muestra en la Figura 30.

Figura 30. Arquitectura de VGG-19



Fuente: (kumar, 2021)

3.4.2.GoogleNet

GoogleNet es un algoritmo creado por Google, entrenado con *ImageNet*, que consiguió entrenar redes neuronales más pequeñas con mejores resultados, en comparación con las CNN. La red de *GoogleNet* utilizó para la clasificación una nueva variante llamada *Inception,* utilizando R-NN para la detección (Garcia-Dominguez et al., 2020).

La arquitectura de *GoogleNet*, que consiste en 22 capas, 27 capas incluyendo la capas *max pooling.* Además, cuenta con 9 módulos de la variante *Inception*, tal y como se muestra en la Tabla 3 (Alake, 2021).

Tabla 3. Arquitectura de GoogleNet

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Fuente: (Alake, 2021)

3.4.3.Inception

Inception es un algoritmo de clasificación creado para redes muy profundas, las cuales son propensas al sobreajuste. La mayor parte de la información de este apartado ha sido extraída del siguiente enlace (Raj, 2020).

Esta red utiliza una CNN inspirada en *LeNet*, es decir, en una estructura de redes neuronales clásica. Las redes neuronales profundas son computacionalmente costosas, es por ello que *Inception* propuso limitar el número de canales de entrada de la red, agregando una convolución adicional de dimensión 1x1, antes de las convolucionales con dimensiones de 3x3 y 5x5. Aunque añadir una capa más parezca contraproducente, las convoluciones 1x1 son mucho más eficientes, computacionalmente, que las 5x5.

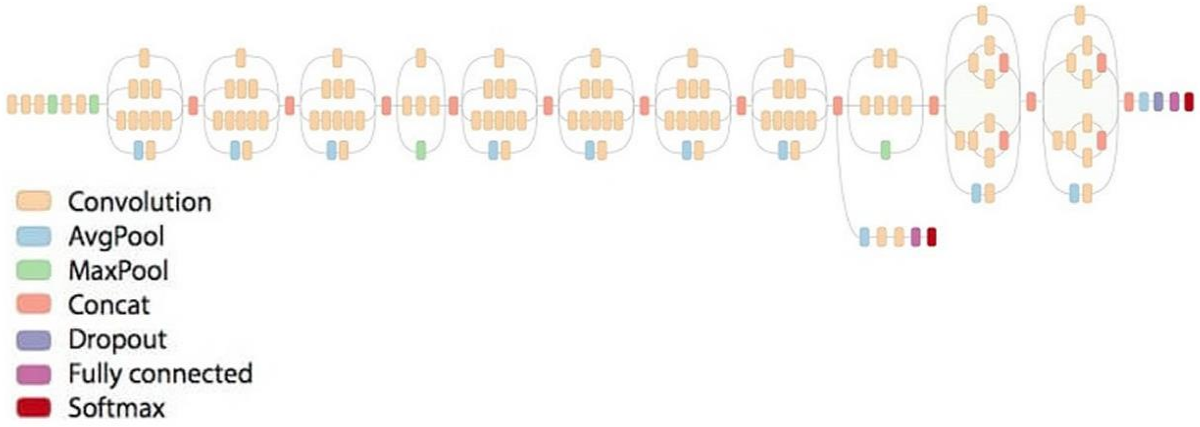
Actualmente existen las versiones V1, V2, V3 y V4 de *Inception*, aunque la más usada y que mejores resultados obtiene en la clasificación de logotipos es *InceptionV3,* según algunos de los últimos estudios.

3.4.3.1. *InceptionV3*

InceptionV3 se presentó en la misma investigación que *InceptionV2.* Se investigaron las posibilidades de mejorar la versión anterior sin retocar mucho los módulos. Este es uno de los motivos por el cual se ha decido centrarse en *InceptionV3* para este trabajo.

La arquitectura de la versión *InceptionV3* tiene como base la red convolucional de sus predecesores, añadiendo varias mejoras como el uso de *Label Smoothing*. Esta técnica intenta mejorar el sobreajuste mediante el uso de la función de pérdida. Además de disponer de diversas capas convolucionales, en vez de darle profundidad a los filtros, se les dio anchura, de esta manera se conseguía mejorar la red sin obtener una pérdida de información a cambio (Raj, 2020). La arquitectura de la red se muestra en la Figura 31.

Figura 31. Arquitectura de InceptionV3



Fuente: (Milton-Barker, 2019)

3.4.4.InceptionResnet

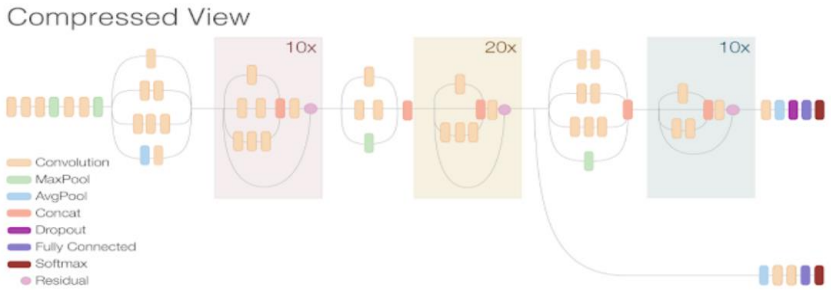
Inception-Resnet e *InceptionV4* se presentaron en el mismo documento. Estas arquitecturas intentaban conseguir módulos más uniformes, con el objetivo de mejorar el rendimiento (Garcia-Dominguez et al., 2020).

3.4.4.1. *InceptionResnetV2*

InceptionResnetV2 es un algoritmo de la familia *Inception,* con una arquitectura más profunda que *InceptcionV3* y, según algunos estudios recientes, con mejor un mejor rendimiento en la clasificación.

La arquitectura comprimida de *InceptionResnetV2* se muestra en la Figura 32, la compresión es debido a su profundidad. Su arquitectura es mucho más precisa que la de modelos anteriores, aun así, el nivel requerido de memoria no es tan elevado como el esperado.

Figura 32. Arquitectura de InceptionResnetV2

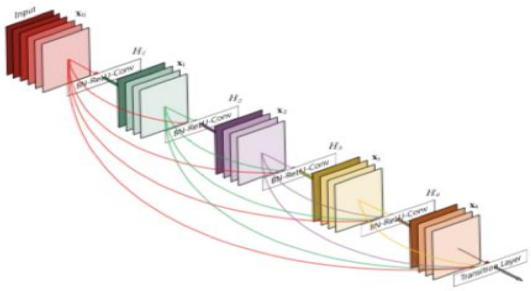


Fuente: (Mehta et al., 2018)

3.4.5.DenseNet

Denset se creó gracias a la premisa de que mediante una conexión densa se lograsen menos parámetros y con una alta precisión. Este algoritmo utiliza entradas adicionales de todas las capas anteriores, pasándoles sus propios mapas de características. Cada capa recibe el conocimiento colectivo de los procesos anteriores, consiguiendo un número de canales menor y una mayor eficiente computacional (Tsang, 2019). La arquitectura se muestra en la Figura 33.

Figura 33. Arquitectura densa de DenseNet



Fuente. (Ruiz, 2018)

3.4.6. YOLO

You Only Look Once (YOLO) utiliza *Deep Learning* y las CNN para el reconocimiento de objetos en tiempo real. Algunos estudios recientes obtienen mejores resultados que las *Faster R-CNN,* en el ámbito del reconocimiento de objetos. Estos estudios comienzan a realizar pruebas en conjuntos de datos más amplios y con un número de clases mayor, animando a los investigadores a usar este tipo de *datasets* (Wang et al., 2020).

Existen diferentes versiones de YOLO, entre las más actuales y destacadas están *YOLOV3*, *YOLOV4* y *YOLOV5*. Aunque sus aplicaciones están más centradas a la detección de objetos en tiempo real junto con su ubicación en la imagen, es por ello que no se profundizará en este algoritmo.

3.5. FIrmcla

Flrmcla es una herramienta de código abierto para la clasificación de imágenes. Es capaz de clasificar imágenes mediante el uso de diferentes algoritmos disponibles en su implementación como *VGG*, *ResNet*, *GoogleNet*, etc (Garcia-Dominguez et al., 2020).

Puede ser una herramienta idónea para poner a prueba algunos de los diferentes tipos de algoritmos mencionados anteriormente. Además, puede resultar útil para probar diferentes algoritmos de detección de imágenes en su conjunto.

3.6. Problemas a resolver en la Inteligencia Artificial del reconocimiento de logos

En este apartado se van a tratar las principales limitaciones actuales que tiene el reconocimiento de logos en la IA. Se comenzará tratando la limitación que existe en los conjuntos de datos y el etiquetado de los mismos. Se seguirá con la problemática proveniente de la cantidad de clases a predecir y, finalmente, se comentarán los tiempos de ejecución utilizados para resolver la problemática de LD.

3.6.1. Escasez en el número de conjuntos de datos

La escasez que existe en torno a los conjuntos de datos para LD se ha mostrado anteriormente. Esta problemática hace que los datos investigados hasta el día de hoy necesiten revisiones y se requieran nuevos estudios para afianzarlos.

3.6.1.1. *Limitación en el número y balanceado de instancias*

La limitación en el número del conjunto de datos viene de la mano con la limitación que existe en el número de imágenes por cada uno de ellos. Observando la Tabla 1, se puede comprobar el número escaso de imágenes que tienen algunos *datasets* como, por ejemplo, *MICC-Logos*, el cual consta de 13 clases y un total de 720 imágenes. Es decir, en el mejor de los casos, constará de aproximadamente 55 imágenes por clase.

3.6.2. Limitada cantidad de clases a predecir por conjunto de datos

La limitación en el número de clases es otro de los problemas a la hora de la detección de imágenes, ya que la mayor parte de los *datasets* utilizados, en el estado del arte, disponen de menos de 200 clases para clasificar. Esto deja al descubierto la problemática que existe en entornos donde las clases a predecir superen estas 200 clases. Sería recomendable poner a prueba el rendimiento de los algoritmos con *datasets* más variados.

3.6.3. Dificultad en el etiquetado de los conjuntos de datos

En la problemática de reconocimiento de objetos, actualmente, es imprescindible que los conjuntos de datos estén correctamente etiquetados. Cada imagen debe ir con su etiqueta. El problema radica cuando los conjuntos de datos son a nivel de objeto, etiquetar cada objeto es una labor muy laboriosa y que, normalmente, se realiza manualmente.

Los algoritmos de la familia YOLO trabajan a nivel de objeto. Esto supone el tener que disponer, previamente, de un *dataset* etiquetado a este nivel. También existe la posibilidad de etiquetarlos mediante herramientas como *RobotFlow, LabelImage…* *(ultralytics, 2022)*.

3.6.3.1. *RobotFlow*

RobotFlow es una de las herramientas más completas para organizar, etiquetar y distribuir, en distintos formatos, los conjuntos de datos de los que dispongamos. Gran parte de la información de este apartado se ha obtenido mediante la siguiente referencia (RobotFlow, 2022).

Es una herramienta ideal para localizar los objetos y delimitarlos, manualmente, en cada una de las imágenes del *datasets.* Además, en la propia herramienta, se da la posibilidad de utilizar su propia herramienta de *Data Augmentation*. Cabe destacar que el proceso de aumentar el *datasets* está limitado, en su plan gratuito, a 3 nuevas imágenes por cada imagen.

Es importante destacar que, en el plan gratuito, se pueden subir hasta 1000 imágenes. Para hacerse una idea, de todos los conjuntos de datos mencionados en este trabajo, sólo 2 disponen de menos de 1000 imágenes.

Esta herramienta está indicada y recomendada por la comunidad de YOLO (ultralytics, 2022). En el caso de que el conjunto de datos supere las 1000 imágenes, algo normal según los *datasets* nombrados en el estado del arte, los planes de pago ascenderían a 1000$ al mes.

Para hacerse una idea de la cantidad de conjuntos de datos etiquetados que existen en *RobotFlow,* públicamente, disponen de tan solo 39. Lo que explica la dificultad que existe en etiquetar cada objeto de un *dataset*. Es por ello que, por el tiempo disponible para este proyecto, se intentará evitar el etiquetado manual de los datos.

3.6.3.2. *Programas de código abierto para el etiquetado*

Existen alternativas de código abierto para el etiquetado de imágenes como *LabelImg*. Es una herramienta gráfica gratuita que guarda las anotaciones de las imágenes del conjunto de los datos en XML. También es compatible con los formatos de los algoritmos de YOLO. Aún así, no dispone de *Data Augmentation*. Al ser una herramienta menos sofisticada y proporcionar menos soporte que *RobotFlow,* el tiempo a invertir en el etiquetado podría llegar a ser mayor (LabelImg, 2022).

3.6.3.3. *Preparado manual del Dataset*

Una alternativa básica, para el etiquetado de conjuntos de datos, se basa en etiquetar cada objeto de cada imagen manualmente mediante un documento de texto tipo XML. El tiempo y esfuerzo requeridos en el etiquetado son aún mayores que con las herramientas anteriores (ultralytics, 2022).

3.6.4. Elevados tiempos de ejecución

Los tiempos de ejecución de los algoritmos previamente mencionados, con una GPU pueden llegar a durar horas, incluso días. Por lo tanto, el tiempo de ejecución en un ordenador convencional, pueden llegar a ser demasiado costoso para el mismo.

3.7. Soluciones a algunas de las limitaciones

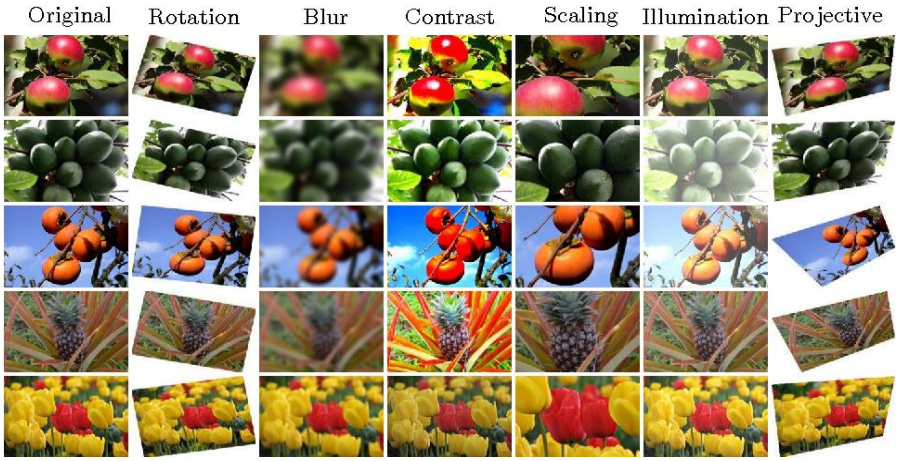
Algunas de las dificultades del estado del arte han sido solucionadas total o parcialmente. A continuación, se mostrarán algunas de estas soluciones.

3.7.1. *Data Augmentation* para el aumento de conjuntos de datos

Data Augmentation es un conjunto de técnicas que permite obtener conjuntos de imágenes más grandes y variados. Para conseguirlo trata de variar, voltear, girar… un conjunto de imágenes, con el fin de obtener más imágenes semejantes al conjunto de datos inicial. Construyendo de esta manera un conjunto de datos más amplio y variado. Existes estudios que demuestran la mejora que ejercen este tipo de técnicas en las *Redes Neuronales Convolucionales.* En estos estudios se analiza el comportamiento de las CNN utilizando *datasets* con y sin *Data Augmentation*. Concluyen con un mejor rendimiento de las CNN mediante la generalización y la variedad del conjunto de datos que ofrecen este tipo de técnicas (Romero et al., 2017).

Entre las técnicas para aumentar el conjunto de datos se dispone de algunas como la rotación, el desenfoque de movimiento denominado *blur*, cambios en el contraste, escalado o *zoom* de los objetos, cambios en la iluminación y cambios de perspectiva. Algunos ejemplos de este tipo de técnicas se muestran en la Figura 34. En esta Figura 34, se observan diferentes imágenes de frutas y flores y, además, sus respectivos cambios para realizar el aumento de imágenes mediante el uso de las originales.

Figura 34. Data Augmentation aplicado a diferentes imágenes



Fuente: (Takimoglu, 2021)

Dependiendo del conjunto de datos se pueden utilizar unas técnicas u otras. Por ejemplo, existen técnicas para el aumento de los datos que varían el color de las imágenes originales. Estas técnicas pueden no ser útiles si el cambio de los colores es un aspecto muy representativo de las clases del conjunto de datos del que se dispone. Un ejemplo de ello puede ser la clasificación de logos de marcas de bebidas, en la que la marca *Coca-Cola* está claramente referenciada mediante el color rojo. Es por ello que, si se variase el color de los logos de *Coca-Cola*, la efectividad en el reconocimiento de esta clase podría verse afectada.

El conjunto de técnicas de *Data Augmentation* es capaz de aumentar sustancialmente los conjuntos de datos. Aun así, al variar o aumentar los datos en exceso, se podrían perder las características principales del conjunto, imposibilitando la mejora y el aprendizaje del modelo a la hora del entrenamiento. Este tipo de variaciones excesivas, pueden ocurrir cuando se añade demasiado ruido en la imagen. También al realizar un *zoom* excesivo o incluso cuando se altera el color del conjunto de las imágenes, como se ha mencionado anteriormente. En la Figura 35, se puede observar un ejemplo de ello. Se observa como un ruido excesivo puede hacer irreconocible el logo de *NEAR Protocol* en la imagen seleccionada.

Figura 35. Ruido *excesivo al aplicar Data Augmentation*

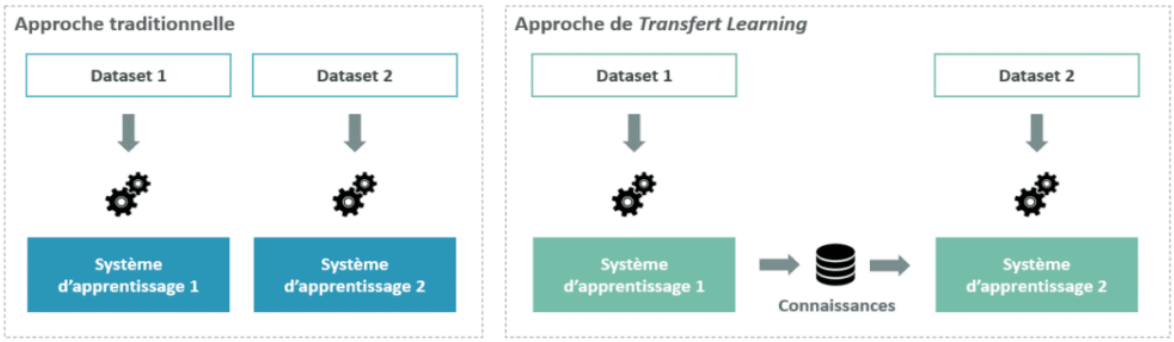
Es de vital importancia eliminar las imágenes que contengan un exceso de ruido del conjunto de datos, ya que no aportan ningún tipo de característica a la clase y pueden empeorar la calidad del modelo.

3.7.2. *Transfer Learning* para el elevado tiempo de entrenamiento

Transfer Learning (TL) se refiere al conjunto de métodos que consisten en poder transmitir conocimiento, mediante diferentes métodos, para resolver problemas de una tipología similar (Attal, 2022).

En el ámbito del *Machine Learning*, la principal diferencia que existen entre enfoque tradicional y el *TL*, se basa en la reutilización de modelos previamente entrenados por parte del *TL*. En la Figura 36**,** se puede observar como el enfoque tradicional se entrenan modelos diferentes para conjuntos de datos diferentes. En TL, en cambio, se ve cómo se reentrenan los modelos que ya han sido entrenados previamente. En el ámbito de la detección de logotipos, si se cuenta con grandes cantidades de datos y con un número de clases muy variadas, el entrenamiento podría ser inasumible para un equipo doméstico. Y aquí es donde gana en importancia el *Transfer Learning*, puede ayudar a entrenar modelos aprovechando al máximo los recursos disponibles.

Figura 36. Diferencias entre el enfoque tradicional y Transfer Learning



Fuente: (Attal, 2022)

Algunas de las últimas investigaciones tratan problemas de medicina mediante el uso del *Transfer Learning*. Utilizan equipos domésticos, es decir, con una CPU *Intel-Core I5* y 7.7 GiB de *Ram*. En este tipo de investigaciones, el número de clases diferentes a predecir se reduce a 7 (Garcia-Dominguez et al., 2020)

Estudios recientes, analizan el *Transfer Learning* mediante *Redes Neuronales Convolucionales Profundas* en paralelo, de esta manera aprovechan la evolución computacional que se está dando cada año. Estos estudios consiguen buenos resultados, en el mejor de los casos, alrededor de un 98% de precisión. Cabe destacar que los *datasets* usados tienen alrededor de 32 clases, como el *dataset* **Logos32-plus** (Karimi & Behrad, Mar 2019).

Existen estudios que intentan predecir el tipo de comida entre 101 clases de comida diferentes, construyen modelos mediante el uso de *Transfer Learning* y logran una precisión del alrededor del 88%. Por lo tanto, cuantas más clases diferentes haya y más difieran los objetos en su forma, más complicado es construir un modelo y más baja la precisión del mismo. En este caso, usan los modelos preentrenados *MobileNetV2*, *ResNet50*, *InceptionResNetV2* y *Xception*, obteniendo los mejores resultados con el modelo *InceptionResNetV2* y logrando una precisión cercana al 89% (Sánchez López, 2019).

Actualmente también existe la problemática a la hora de etiquetar ciertos logos cuando realmente no lo son, es decir, en algunos modelos actuales se etiquetan marcas de ropa en imágenes en las que no hay ningún tipo de marca. Para solucionar esta problemática se está usando más de un clasificador de manera secuencial, es decir, dejando los resultados de uno de los clasificadores como entrada del siguiente, de manera que el último consiga mejores resultados. En general se consiguen buenos resultados, probado en modelos que consiguen predecir 8 tipos de marcas con alrededor de un 98% de precisión. Aun así, es un ámbito en el que se sigue trabajando por el cambio que ejercen las marcas en sus logos y la limitación que esto produce en los *datasets* ya construidos (Le, 2021).

3.8. Tecnologías para la implementación

Se van a analizar los diferentes servicios tecnológicos que existen para los diferentes apartados de la detección de logos, mostrando las marcas y empresas más importantes del sector.

3.8.1. Entrenamiento de los modelos

Existen diversas herramientas para el entrenamiento mediante el uso de *Redes Neuronales*. A continuación, se explicarán algunas de estas herramientas explicando a un alto nivel su rendimiento y su utilidad para el cometido del *Transfer Learning*.

3.8.1.1. *Tensorflow*

Es una plataforma de extremo a extremo y de código abierto utilizada para el aprendizaje automático desarrollada por *Google Brain Team* (TensorFlow, 2021). Su arquitectura flexible permite implementar los cálculos en una o más CPU, GPU, servidores… Existen investigaciones recientes que prueban los diferentes usos que puede tener *TensorFlow* para la detección de imágenes mediante redes neuronales, usando el lenguaje de programación *Python* (Abu et al., 2019).

Tensorflow tiene la opción de realizar *Transfer Learning* mediante *Redes Neuronales* ya entrenadas. Dispone de una gran cantidad de opciones para este cometido, todo ello a un alto y bajo nivel de programación, siendo ideal para configurar al máximo los entrenamientos deseados.

3.8.1.2. *Keras*

Keras es una biblioteca de código abierto diseñada para entrenar *Redes Neuronales,* puede ejecutarse sobre *TensorFlow*, *Microsoft Cognitive Toolkit* o *Theano.* Es una herramienta útil y sencilla que permite entrenar *Redes Neuronales* en poco tiempo, acelerando la creación de modelos de aprendizaje profundo.

Es una opción idónea para realizar Transfer Learning, ya que dispone de opciones para eliminar las últimas capas de las *Redes Neuronales* ya entrenadas de manera sencilla, todo ello a un alto nivel de programación (Cassimiro, 2021).

3.8.1.3. *Pythorch*

Pythorch es una biblioteca para *Python* que facilita la creación de proyectos de aprendizaje profundo, permitiendo flexibilidad a la hora de expresar los modelos en *Python*. Trabaja a bajo nivel de programación, lo que puede dificultar su uso en personas inexpertas. Los desarrolladores solo tienen que preocuparse por la problemática que den sus modelos y no por los errores que tenga el propio lenguaje de programación. *Pythorch*, además, admite cálculos dinámicos que permiten modificar el comportamiento de la red sobre la marcha, a diferencia de los gráficos estáticos que ofrece *TensorFlow* (Tran, 2020).

Utilizando *Pythorch* se puede realizar *Transfer Learning* con una gran variedad de ajustes obteniendo un gran rendimiento, una opción ideal para programadores expertos que conozcan bien la herramienta (Rosebrock Adrian, 2021).

3.8.2.Despliegue de los modelos

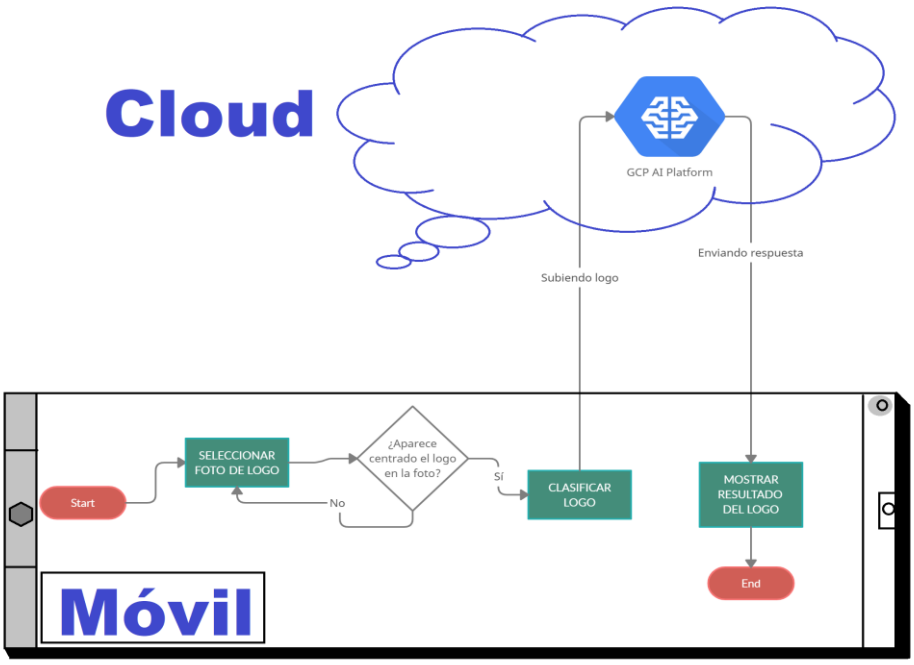
El modelo generado para la clasificación puede ser desplegado en la nube o en el propio dispositivo. Es necesario e importante tomar la decisión sobre donde ubicar el modelo generado, ya que tendrá consecuencias en la implementación final del proyecto y en su seguridad, tal y como se explicará en los siguientes apartados.

3.8.2.1. *Online: desplegados en la nube*

Empresas como IBM, *Google*, *Amazon* y *Microsoft* comienzan a disponer de herramientas para poder implementar soluciones de *Inteligencia Artificial* en la nube. Algunas de estas herramientas pueden alojar los modelos entrenados en sus servidores.

Además, las continuas actualizaciones y los seguimientos del modelo pueden hacerse de manera sencilla (Shaikh, 2022). En la parte de En la Figura 37 se muestra el diagrama de flujo de una aplicación móvil que clasifica los logos mediante un modelo almacenado en *Google Cloud*.

Figura 37. Diagrama de flujo que representa la clasificación de logos en la nube



Algunos de los beneficios que tienen este tipo de implementaciones *online*, para el desarrollador, se basa en la exclusividad, ya que un modelo albergado en la nube puede ser menos accesible a los usuarios. Es decir, a la hora de la clasificación, el dispositivo manda una petición a la nube y, el usuario, queda únicamente a la espera de la respuesta. Por lo tanto, el usuario no tiene acceso al modelo de clasificación, solo a la respuesta del mismo.

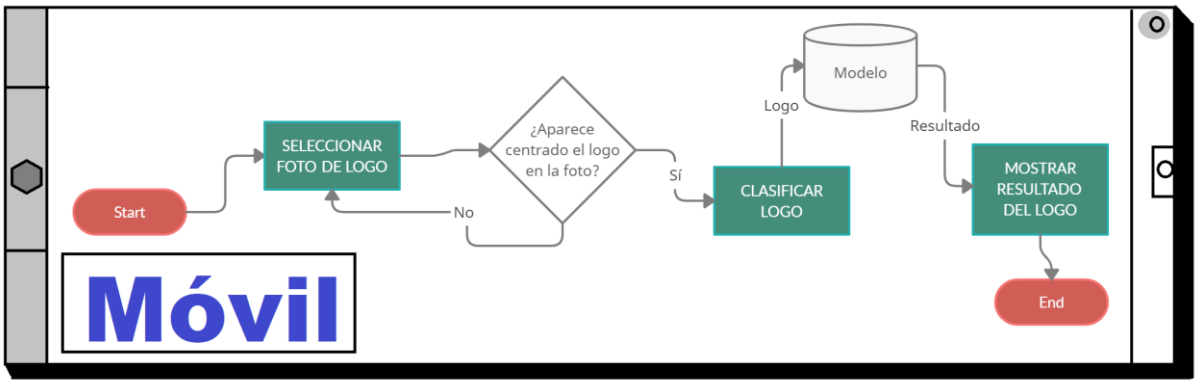
Por otra parte, uno de los inconvenientes que tienen este tipo de modelos se basa en su disponibilidad, ya que la falta de conexión a internet impide el acceso al modelo y, por consecuente, al correcto funcionamiento de la propia aplicación.

3.8.2.2. *Offline: desplegados en el dispositivo*

Desplegar los modelos en el dispositivo se centra en la opción de disponer de todo lo necesario de la clasificación dentro de la propia aplicación. Gracias a ello se consigue autonomía, ya que se depende de empresas externas para el buen funcionamiento de la aplicación.

Por otra parte, la velocidad a la que clasifica el modelo dependerá de la velocidad del *hardware* que tiene el propio dispositivo. Por lo tanto, esta velocidad se verá gravemente afectada si no se dispone de un dispositivo de categoría media-alta (Shaikh, 2022). El diagrama de flujo que representa la clasificación de logos *offline* se muestra en la Figura 38.

Figura 38. Diagrama de flujo que representa la clasificación de logos offline



Al escoger este tipo de implementaciones *offline,* es importante tener en cuenta el tamaño que ocupa el modelo, ya que algunos modelos pueden llegar a pesar más de 100MB. Se puede convertir en un problema ya que este tamaño dobla la media de lo que ocupan las aplicaciones móviles hoy en día.

3.8.2.2.1. PyTorch Mobile

PyTorch Mobile es un *plugin* que proporciona un entorno de ejecución para modelos de aprendizaje automático dentro de los dispositivos móviles. Actualmente se encuentra en una versión *Beta* y se está probando en entornos de desarrollo (PyTorch, 2022).

3.8.2.2.2. TensorFlow Lite

Es una librería que se utiliza para poder llevar *TensorFlow* a los dispositivos móviles. Cabe destacar que durante los últimos 5 años ha ido creciendo el interés por esta librería (Google Trends, 2021).

3.9. Problemas a resolver en las tecnologías para la implementación

Las tecnologías avanzan e innovan cada año, aun así, existen ciertas limitaciones a tener en cuenta. A continuación, se van a analizar algunas de estas limitaciones.

3.9.1. Altos recursos computacionales

La clasificación de logos, previamente, requiere de grandes cantidades de datos para poderse llevar a cabo. Tratar un gran volumen de datos requiere de equipos de gran capacidad. Mediante el *Transfer Learning* se puede disminuir la capacidad necesaria de estos equipos hasta cierto punto. Cuando se pasa de clasificar 32 clases a 200 clases, el coste computacional asociado al entrenamiento y tratamiento de los datos aumenta sustancialmente.

3.9.2. Falta de documentación

La falta de documentación de algunas de las nuevas herramientas de *Inteligencia Artificial*, en el ámbito de la detección de logos, pueden dificultar el desarrollo. Como ejemplo de ello está el

plugin PyTorch Mobile, ya que algunos desarrolladores evitan su uso por la falta de documentación que existe del mismo (Shaikh, 2022).

3.9.3.Limitaciones en la nube

El almacenamiento en la nube no se libra de ciertas limitaciones que pueden afectar al desarrollo e implementación de los proyectos. A continuación, se explicarán algunas de las limitaciones más comunes en referencia a LD.

3.9.3.1. *Dificultades al utilizar datasets de gran tamaño*

El entrenamiento de los modelos de LD requieren de conjuntos de datos de gran tamaño. Esto conlleva que haya que subir los datos a la nube, existiendo algunas limitaciones a la hora de mover este tipo de archivos entre apartados de *Google Drive*, por ejemplo. Se han documentado errores al trabajar con conjuntos de datos de gran tamaño en *Google Colab* (Konkiewicz, 2021).

Además, las cuentas gratuitas de almacenamiento en la nube como *Google Drive*, tienen una capacidad máxima de 15GB, pudiendo no ser suficiente para algunos conjuntos de datos.

3.9.3.2. *Cortes de conexión que interrumpen el trabajo*

Entrenar modelos con un gran volumen de datos requiere de muchas horas, incluso días. Es por ello que hay que tener en cuenta las limitaciones en la conexión que tienen soluciones como *Google Colab.* Los usuarios reportan conexiones continuas en las modalidades gratuitas, que se siguen dando incluso en las versiones de pago. Todo ello está documentado en la propia página de *Google,* donde explican las 12 horas de vida útil que tiene cada *Notebook* y la posible desconexión que puede ocurrir por cierto tiempo de inactividad (Google Colab, 2022). Estas desconexiones causan la pérdida total del entrenamiento realizado hasta el momento.

3.9.3.3. *Coste elevado*

Normalmente, el uso de las infraestructuras en la nube se tarifica por horas. Es por ello que, en el ámbito de la *Inteligencia Artificial,* este tipo de soluciones estén más orientadas al entorno

empresarial. Aun así, los altos costes por el uso de este tipo de servicios pueden hacer de las infraestructuras *cloud* poco accesibles a medianas y pequeñas empresas.

3.10. Resumen del estado del arte

Es indudable la evolución de la *Redes Neuronales Convolucionales* (CNN) hasta redes más complejas y profundas para solucionar los problemas en la detección de logos.

Se han analizado los *datasets* y el número de clases que intentan predecir, todo ello durante los últimos años, comprobando la utilización de conjuntos de datos con un limitado número de clases.

Queda demostrada la utilización del *Transfer Learning* para el problema de LD durante los últimos años, solucionando, parcialmente, el problema del límite computacional que se tiene actualmente. Además, observando los resultados de las investigaciones, el algoritmo *VGG16* se posiciona como una de las mejores soluciones, teniendo en cuenta el tiempo de entrenamiento y las métricas de evaluación.

Se ha mostrado la dificultad que existe en la delimitación de objetos en conjuntos de datos grandes y algunas herramientas para mejorarlo. Aun así, el coste que suponen herramientas como *RobotFlow*, puede que no sean asequibles para algún tipo de colectivos, como, por ejemplo, estudiantes e investigadores. Las alternativas *open-source* a *RobotFlow*, como *LabelIMG*, requieren un tiempo excesivo de etiquetado, aportando grandes requerimientos a proyectos que cuenten con una cantidad elevada de clases.

Sabiendo que en la problemática de LD se usan conjuntos con un número limitado de clases, sería conveniente investigar con conjuntos de logotipos que tengan un mayor número de clases a la media. De esta manera se podría diversificar el conocimiento ya adquirido.

Por lo tanto, el ámbito de LD es un lugar para seguir investigando, faltan muchos nuevos *datasets* por crear y descubrir. También es necesario poner a prueba los algoritmos ya diseñados en conjuntos de datos más variados y, si fuese necesario, crear otros. Aun así, la técnica del *Transfer Learning*, teniendo en cuenta los buenos resultados que ha obtenido, ya comentados anteriormente, puede ser la técnica idónea para seguir investigando en este nicho de la detección de logos.

Por lo tanto, el problema de la detección y asociación de logos en el ámbito de las criptomonedas, al disponer de una gran cantidad de logos diferentes, podría ser la opción ideal para tratar este nicho de investigación que existe en este ámbito.

4. Objetivos y metodología de trabajo

En este apartado se tratarán los objetivos del proyecto y su metodología, los cuales mostrarán el aporte que ofrece este proyecto.

4.1. Objetivo general

El objetivo general de este proyecto es **construir una aplicación móvil que ayude a evitar los fraudes en el intercambio de criptomonedas**. Se llevará a cabo mediante la detección de los logos de las criptomonedas que aporte el usuario y, tras ello, se le remitirá información relevante de las mismas. El usuario facilitará los logos a la aplicación mediante capturas de pantalla que obtenga de grupos de *Telegram* o de cualquier parte de Internet. La aplicación será capaz de reconocer los logos de las criptomonedas con mayor capitalización de mercado y, tras ello, mostrará la información más relevante asociada a dichas criptomonedas.

4.2. Objetivos específicos

Los objetivos específicos son los siguientes:

1) **Generar un *dataset* adecuado:** Obtener un *dataset* adecuado para en el entrenamiento del modelo.

2) **Evaluar todas las alternativas del estado del arte:** Analizar la idoneidad de los distintos algoritmos del estado del arte para el reconocimiento de logos seleccionando las mejores alternativas. También se analizarán las distintas plataformas actuales para albergar los modelos.

3) **Obtener un modelo para la clasificación:** Se ha de obtener un modelo que sea capaz de clasificar los logotipos de las criptomonedas. Para ello se probarán distintas alternativas, desde la creación de un modelo desde cero, hasta el uso de *Transfer Learning*.

4) **Desarrollar una aplicación móvil:** Se desarrollará una aplicación móvil que contenga una base de datos y el algoritmo implementado mediante una API.

5) **Desarrollar una página web con los términos y condiciones legales:** La página web deberá recoger los términos y condiciones legales asociados a la recogida y almacenamiento de datos personales y a la legalidad vigente en el entorno de las criptomonedas.

6) **Evaluar la solución propuesta:** Se tienen que realizar pruebas al algoritmo y a la aplicación implementada.

4.3. Metodología de trabajo

En este apartado se describe la metodología utilizada con el fin de implementar el proyecto, dando una visión generalizada de las herramientas utilizadas y el proceso de elaboración correspondiente.

4.3.1.EDT

Una *Estructura de Desglose de Trabajo (EDT)* es una base importante para la planificación de un proyecto. En la Figura 39, se muestra la *EDT* correspondiente a este proyecto.

Figura 39. EDT del proyecto

Fuente: (GlooMaps, 2022)

4.3.1.1. *Descripción de tareas*

Se elaboró una lista de tareas y al terminarlas, se les asignó la fecha de inicio y fin, tal y como se muestra en la Tabla 4.

Tabla 4. Listado de tareas del proyecto

Tarea

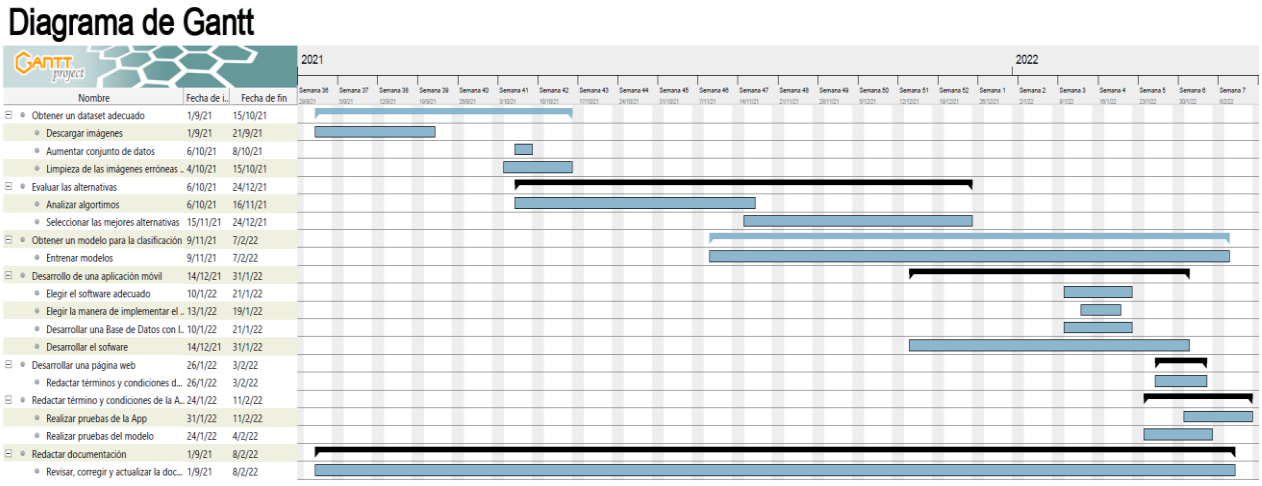
Nombre	Fecha de inicio	Fecha de fin
Obtener un dataset adecuado	1/9/21	15/10/21
Descargar imágenes	1/9/21	21/9/21
Aumentar conjunto de datos	6/10/21	8/10/21
Limpieza de las imágenes erróneas del conjunto de datos	4/10/21	15/10/21
Evaluar las alternativas	6/10/21	24/12/21
Analizar algortimos	6/10/21	16/11/21
Seleccionar las mejores alternativas	15/11/21	24/12/21
Obtener un modelo para la clasificación	9/11/21	7/2/22
Entrenar modelos	9/11/21	7/2/22
Desarrollo de una aplicación móvil	14/12/21	31/1/22
Elegir el software adecuado	10/1/22	21/1/22
Elegir la manera de implementar el modelo	13/1/22	19/1/22
Desarrollar una Base de Datos con las criptomonedas dentro de la App	10/1/22	21/1/22
Desarrollar el sofware	14/12/21	31/1/22
Desarrollar una página web	26/1/22	3/2/22
Redactar términos y condiciones de la App	26/1/22	3/2/22
Redactar término y condiciones de la App	24/1/22	11/2/22
Realizar pruebas de la App	31/1/22	11/2/22
Realizar pruebas del modelo	24/1/22	4/2/22
Redactar documentación	1/9/21	8/2/22
Revisar, corregir y actualizar la documentación	1/9/21	8/2/22

Fuente: (GanttProject, 2022)

4.3.1.2. *Diagrama de Gantt*

El *Diagrama de Gantt* se define como la representación gráfica del tiempo que dedicamos a cada una de las tareas en un proyecto concreto. El diagrama de Gantt realizado para el proyecto es el mostrado en la Figura 40.

Figura 40. Gantt del proyecto



Fuente: (GanttProject, 2022)

4.4. Herramientas, bibliotecas y librerías utilizadas

Se mostrarán algunas de las herramientas, bibliotecas y librerías destacadas para la elaboración de este proyecto, todas ellas del ámbito de la *Inteligencia Artificial*.

4.4.1. Herramientas utilizadas

Se mencionarán las herramientas destacadas para el desarrollo de la IA dentro del proyecto.

4.4.1.1. *Firebase*

Firebase es una plataforma en la nube de *Google* que sirve para el desarrollo de aplicaciones móviles. Además, dispone de módulos para el almacenaje de modelos predictivos en la nube. Es compatible con los modelos creados por *Keras* o *TensorFlow* transformándolos al formato *TFLite*. Por ello es una opción ideal para la elaboración de este proyecto (Firebase, 2022).

4.4.2. Bibliotecas y librerías

Es necesario elegir alguna biblioteca para el entrenamiento del modelo. Además, se ha de disponer de alguna librería para utilizar los modelos de *Inteligencia Artificial* dentro de las aplicaciones móviles. Entre las diferentes opciones, se han elegido las siguientes.

4.4.2.1. *Keras*

Es una biblioteca idónea para realizar *Transfer Learning*, al disponer de opciones para eliminar las últimas capas de las *Redes Neuronales* ya entrenadas de manera sencilla. Además, es la opción más utilizada por los profesores de UNIR en el *Máster en Inteligencia Artificial*. Es por ello que se dispone de más experiencia comparándola con otras bibliotecas como *PyTorch*. La experiencia es otro de los motivos para su elección.

4.4.2.2. *TensorFlow Lite*

Es una librería que se utiliza para poder llevar *TensorFlow* a los dispositivos móviles. El crecimiento en el interés, su utilidad y el soporte del que dispone son las razones por las que se ha elegido está librería.

5. Identificación de requisitos

En este apartado, se muestran listados aquellos puntos indispensables para el cumplimiento de todos y cada uno de los objetivos de este trabajo.

Los requisitos se han dividido en dos tipos:

- **Requisitos Funcionales:** Este tipo de requisitos recoge las acciones esenciales del proyecto, describiendo lo que el producto final debe hacer.
- **Requisitos No-Funcionales:** Define las propiedades que debe tener el producto final. Teniendo en cuenta la usabilidad y el rendimiento. Este tipo de requisitos tienen tanta importancia para el éxito del proyecto como lo tienen los *Requisitos Funcionales*.

5.1. Requisitos Funcionales

Los requisitos funcionales del proyecto son los siguientes:

FR1. El usuario ingresará el número de móvil para acceder a la aplicación aprovechando la opción gratuita que ofrece *Google Cloud Platflorm* y, de esta manera, se loqueará rápidamente sin contraseñas. **(Además, en siguientes versiones de la aplicación, el inicio de sesión mediante número de móvil podrá permitir conectar al usuario con sus contactos.)*

FR2. El sistema le enviará un código SMS de validación para acceder a la aplicación.

FR3. El usuario tendrá opción de ingresar el código SMS de validación para acceder a la aplicación.

FR4. El usuario tendrá la opción de ingresar su nombre de usuario y su foto de perfil.

FR5. El sistema registrará los datos de los usuarios, como el nombre, número de móvil, foto de perfil y las criptomonedas que clasifique dentro de la aplicación.

FR6. El usuario podrá modificar sus datos.

FR7. El usuario tendrá la opción de agregar un logotipo de criptomoneda mediante el uso de la cámara del dispositivo.

FR8. El usuario tendrá la opción de agregar un logotipo de cualquier criptomoneda mediante una captura de pantalla almacenada en su dispositivo móvil.

FR9. La aplicación tendrá que ser capaz de clasificar como mínimo las 100 primeras criptomonedas, ordenadas por capitalización de mercado, de *CoinMarketCap*.

FR10. La aplicación deberá clasificar los logotipos con una exactitud superior al 75%.

FR11. El sistema informará de los posibles errores en la clasificación de los logotipos.

FR12. El sistema mostrará el resultado de la clasificación del logotipo subido por el usuario.

FR13. El sistema dará acceso a información específica de cada criptomoneda clasificada por el usuario.

FR14. El usuario tendrá la opción de guardar la información de la criptomoneda clasificada.

FR15. La página web mostrará los términos y condiciones de la aplicación.

5.2. Requisitos No-Funcionales

Los requisitos no-funcionales del proyecto se listan a continuación:

NFR1. Los datos de los usuarios deberán estar cifrados.

NFR2. Los datos del usuario no serán accesibles por otros usuarios.

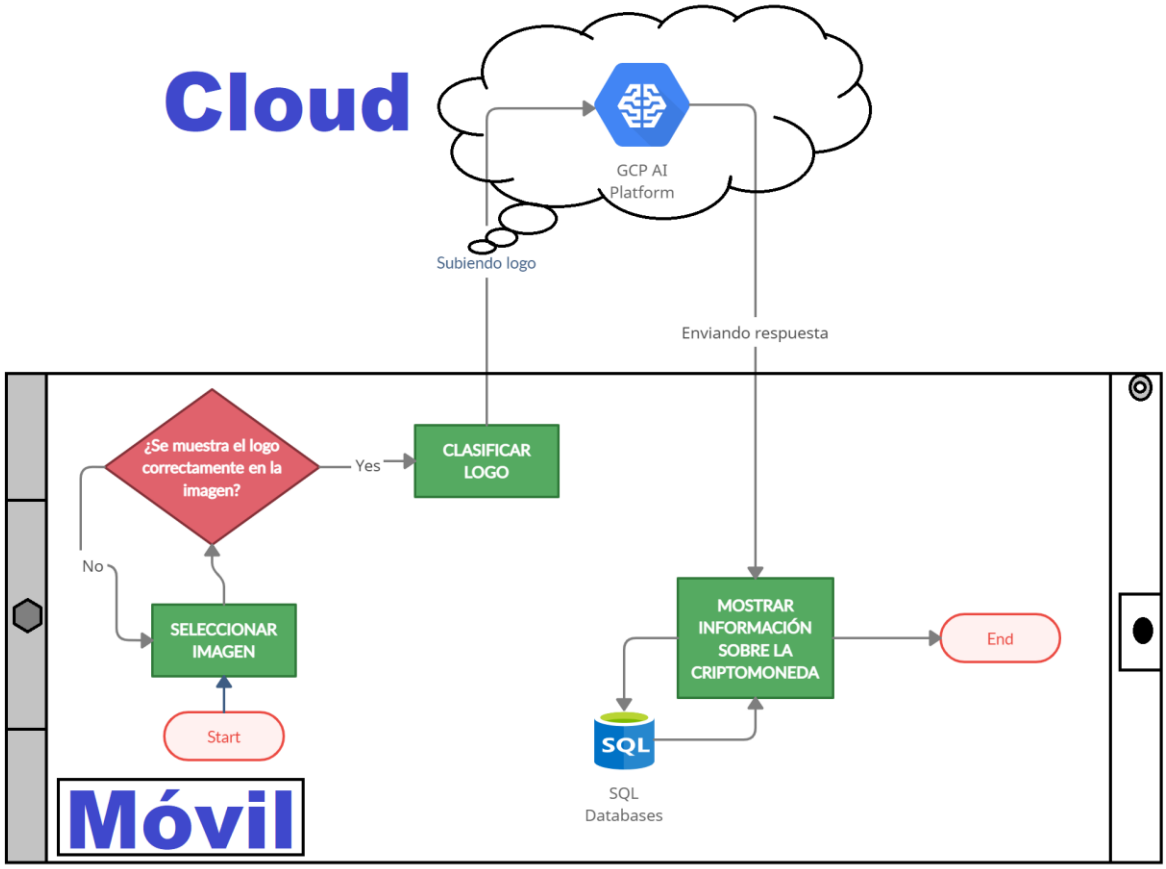
NFR3. Los usuarios deberán estar cifrados en *Firebase* mediante funciones *Hash*.

NFR4. El sistema no dará ningún tipo de recomendación sobre inversiones.

6. Descripción de la herramienta software desarrollada

En este apartado se va a profundizar en los diferentes pasos que se han seguido para la elaboración de la aplicación móvil. Este proceso abarcará la acción de generar el conjunto de datos, el entrenamiento del modelo y sus respectivos resultados, hasta la elaboración y puesta en marcha de la aplicación. Para comprenderlo mejor, el diagrama de flujo asociado a la aplicación es el que se muestra en la Figura 41.

Figura 41. Diagrama de flujo correspondiente a la aplicación propuesta

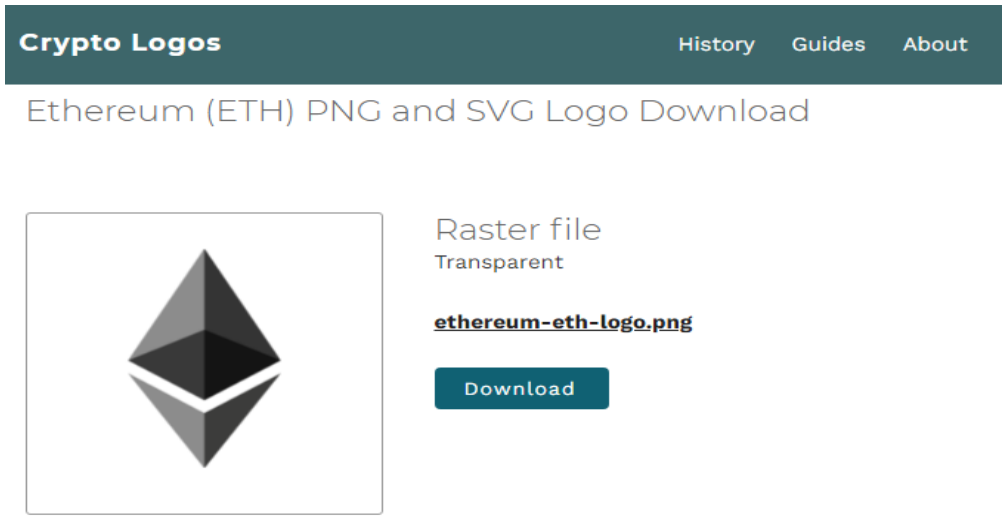


6.1. Generar un *dataset* adecuado

El primer problema a resolver se centra en la obtención del *dataset* inicial, uno que contenga una gran cantidad de logos previamente etiquetados. Esto se debe a que para alimentar cualquier *Red Neuronal* es necesario disponer de una gran cantidad de datos.

Como se explicó anteriormente, existen repositorios en páginas web como: https://cryptologos.cc/. Aun así, el problema que tienen este tipo de *datasets* es que solo disponen de un logo por cada criptomoneda, tal y como se muestra en la Figura 42.

Figura 42. Logo de Ethereum en la web de Cryptologos



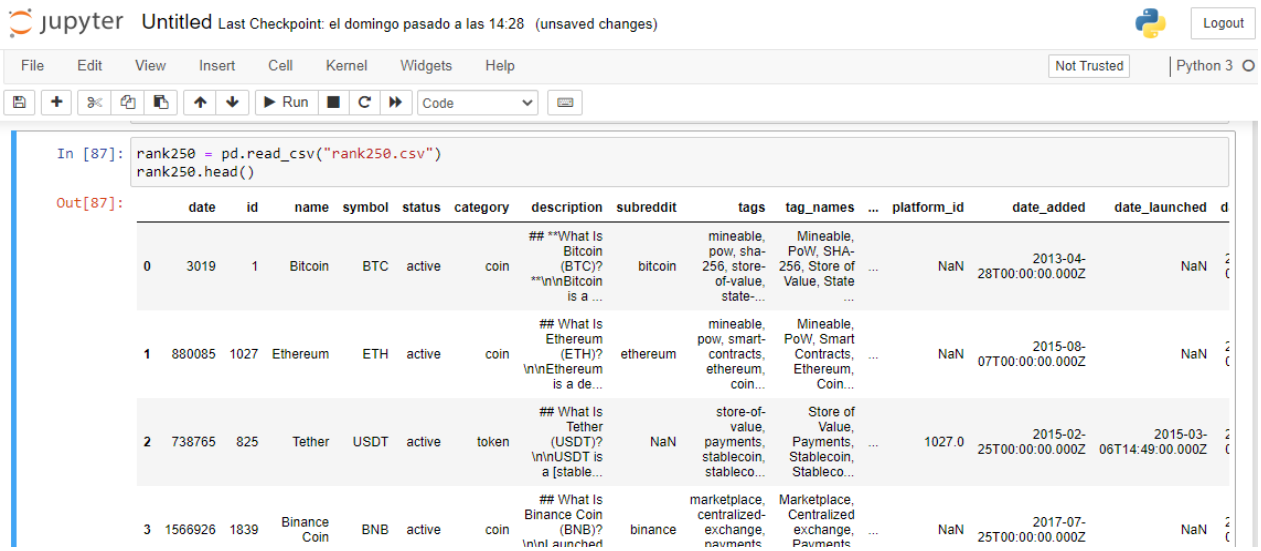
Fuente: (Crypto Logos, 2022)

En este caso, lo realmente necesario es disponer de una gran cantidad diferente de logos de cada criptomoneda, ya que esto ayuda a que las *Redes Neuronales* aprendan mejor a generalizar y se reduzca el riesgo de *overfifting*. Según algunos estudios recientes, si se dispone de una cantidad limitada de datos, es posible realizar *Data Augmentation* para aumentar el *dataset.*

Realizando una búsqueda sobre el logo de diferentes criptodivisas en *Google Imágenes*, se observa que, en este ámbito, a partir de las 40 primeras imágenes, comienzan a salir imágenes en las que los logos no son tan protagonistas. Por lo tanto, la solución planteada se basa en crear un algoritmo en *Python* que sea capaz de descargar, de los *links* que proporciona *Google Imágenes* en las búsquedas, las 40 primeras fotos de cada criptomoneda, de las primeras 250 criptomonedas en el *ranking* de *CoinMarketCap* (CoinMarketCap, 2021).

Existen algunos repositorios en *Kaggle.com* con los datos históricos de las criptomonedas de *CoinMarketCap*. El *dataset* elegido incluye más de 2 GB de información histórica dividida en 2 diferentes bases de datos. El siguiente paso, por lo tanto, sería la del tratamiento y la supresión de la información histórica innecesaria para el fin de este proyecto. Además de ordenar la información teniendo en cuenta el ranking de *CoinMarketCap*, tal y como se observa en la Figura 43 (Olasehinde Bisola, 2021).

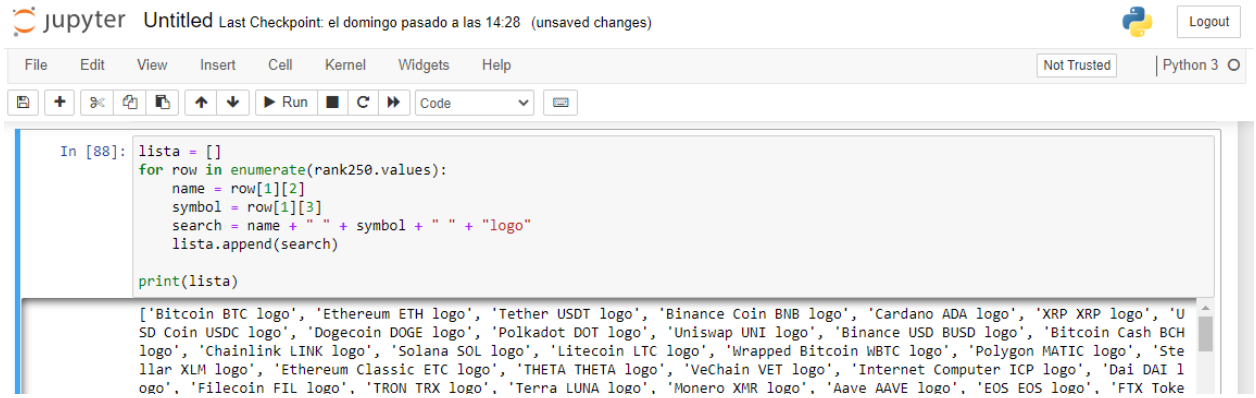
Figura 43. Ordenando el dataset descargado de Kaggle en Jupyter Notebook



	date	id	name	symbol	status	category	description	subreddit	tags	tag_names	...	platform_id	date_added	date_launched	d
0	3019	1	Bitcoin	BTC	active	coin	## **What Is Bitcoin (BTC)?** \n\nBitcoin is a ...	bitcoin	mineable, pow, sha-256, store-of-value, state-...	Mineable, PoW, SHA-256, Store of Value, State ...	...	NaN	2013-04-28T00:00:00.000Z	NaN	2 (
1	880085	1027	Ethereum	ETH	active	coin	## What Is Ethereum (ETH)? \n\nEthereum is a de...	ethereum	mineable, pow, smart-contracts, ethereum, coin...	Mineable, PoW, Smart Contracts, Ethereum, Coin...	...	NaN	2015-08-07T00:00:00.000Z	NaN	2 (
2	738765	825	Tether	USDT	active	token	## What Is Tether (USDT)? \n\nUSDT is a [stable...	NaN	store-of-value, payments, stablecoin, stableco...	Store of Value, Payments, Stablecoin, Stableco...	...	1027.0	2015-02-25T00:00:00.000Z	2015-03-06T14:49:00.000Z	2 (
3	1566926	1839	Binance Coin	BNB	active	coin	## What Is Binance Coin (BNB)? \n\nLaunched	binance	marketplace, centralized-exchange, payments	Marketplace, Centralized exchange, Payments	...	NaN	2017-07-25T00:00:00.000Z	NaN	2 (

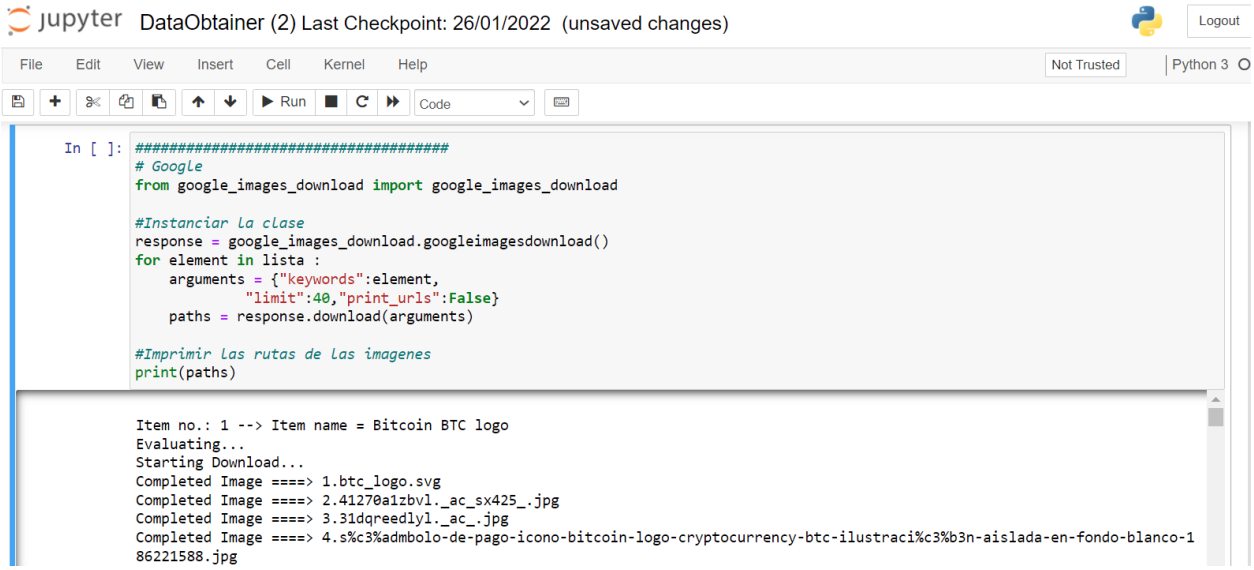
Convendría crear una lista con los nombres y las siglas de las criptomonedas, más la palabra "logo", ordenadas por capitalización de mercado, tal y como se muestra en la Figura 44. De esta manera se tendría todo preparado para automatizar la descarga del *dataset*.

Figura 44. Creando lista con las palabras clave de cada criptomoneda



El siguiente paso, consiste en construir un algoritmo que, por cada criptomoneda, sea capaz de descargar los primeros 40 logos que se encuentren adheridos en *Google Imágenes*. El resultado se muestra en la Figura 45.

Figura 45. Descargando 40 imágenes por cada criptomoneda en Jupyter Notebook



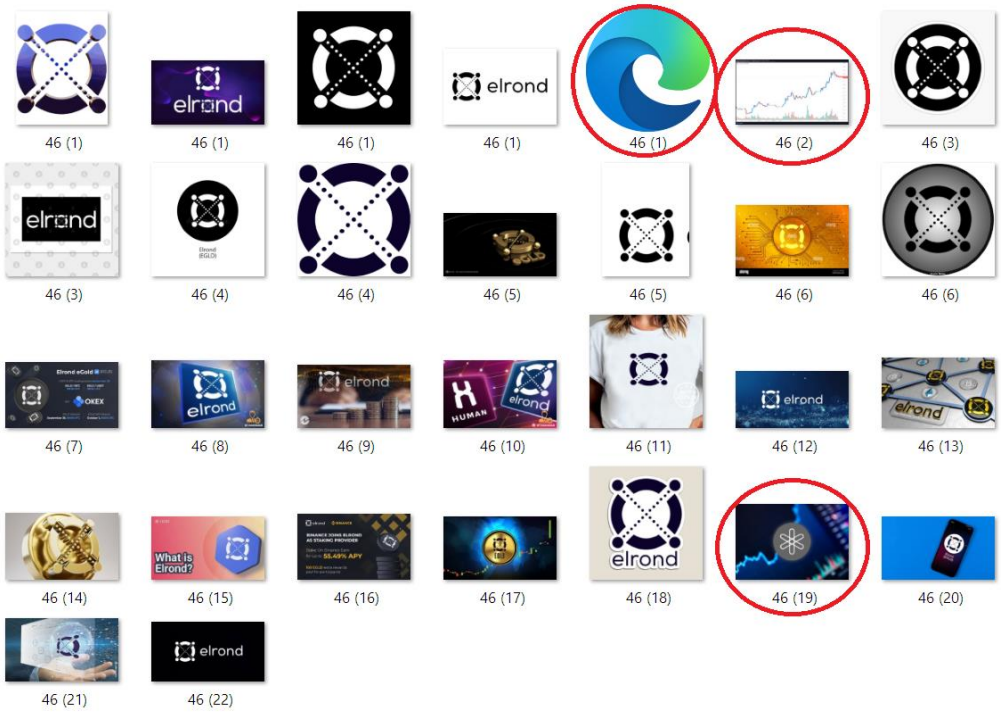
Tras la descarga, en total, se dispone de 40 logotipos por cada criptomoneda, de las 250 primeras criptomonedas del ranking de *CoinMarketCap*. Se divide aleatoriamente el conjunto, dejando un 25% de los logotipos para el conjunto de validación. La cantidad total de logotipos, balanceados entre las clases, asciende a 7.500 para el entrenamiento y, 2.500 para la validación, tal y como se muestra en la Tabla 5.

Tabla 5. Distribución de los logotipos del Dataset 1

Dataset 1	Entrenamiento	Validación
Nº Clases (Criptomonedas)	250	250
N.º Imágenes/Clase	30	10
Nº Total Imágenes	7500	2500

El siguiente problema a resolver se ubica en la calidad del *dataset*, ya que, de las 40 imágenes descargadas, por cada criptomoneda, algunas no corresponden a sus logos. Por lo tanto, para mejorar la calidad del conjunto de datos, es necesario revisar el *dataset* manualmente y borrar las imágenes inadecuadas para el fin del proyecto. Un ejemplo de las imágenes que se descargan incorrectamente, se muestra redondeadas en la Figura 46.

Figura 46. Muestra de algunas imágenes descargadas incorrectamente



Tras eliminar las imágenes que no corresponden a cada criptomoneda, en caso de que algunas clases queden desbalanceadas, es necesario volver a descargar más logotipos de dichas criptomonedas. Finalmente, el *dataset* se compone de 9615 logotipos repartidos entre las 250 clases. En cada clase, del conjunto de entrenamiento, hay entre 26 y 30 imágenes, manteniendo un balance entre las mismas, tal y como se muestra en la Tabla 6.

Tabla 6. Distribución de los logotipos del Dataset 1 tras revisarlo

Dataset 1 (Revisado)	Entrenamiento	Validación
NºClases (Criptomonedas)	250	250
N.º Imágenes/Clase	26-30	10
Tamaño (GB)	1,4	0,08
Nº Total Imágenes	7115	2500

En el peor de los casos, algunas clases disponen de 26 logotipos dada la dificultad de encontrar variedad en algunas de las criptomonedas menos conocidas.

6.1.1. *Data Augmentation* para aumentar la variedad del conjunto de datos

Algunos estudios afirman que un pequeño aumento de los datos no tiene mucha influencia en el rendimiento del clasificador. Y, además, que un gran aumento de los datos de entrenamiento tiene influencia en la buena generalización del modelo (Romero et al., 2017). Por lo tanto, se ha decidido generar 3 *datasets* realizando diferentes aumentos. De esta manera, aparte de probar la exactitud de algunos algoritmos del estado del arte, se van a poder comparar los resultados con distintitos niveles de *Data Augmentation*. El conjunto de validación siempre será el mismo, el mostrado en el *Dataset 1 (Revisado)*. Únicamente variará el conjunto de entrenamiento.

Para realizarlo de manera controlada, se ha diseñado un algoritmo de *Data Augmentation* que genera 3 tipos de aumento con las imágenes de entrenamiento del *Dataset 1 (Revisado),* el que se muestra en la Tabla 6. Cada conjunto de datos creado contiene al anterior, pero no al revés. De esta manera, cada conjunto de datos contiene un número de imágenes mayor al anterior.

En todas y cada una de las iteraciones del aumento de imágenes, las modificaciones que han tenido los nuevos logotipos generados han sido la rotación, el *zoom,* la inversión y el añadido de ruido mediante una función aleatoria. El grado de este tipo de modificaciones ha ido disminuyendo en cada una de las 3 iteraciones. La disminución del ruido es clave para obtener *datasets* mayores, pero sin obtener un ruido excesivo a cambio. A continuación, se detallarán los *datasets* obtenidos.

6.1.1.1. *Data Augmentation dividida en 3 iteraciones*

En la primera iteración, el algoritmo genera 10 imágenes por cada imagen de cada clase de entrenamiento del *Dataset 1 (Revisado)*. A cada una de estas imágenes se le añade un ruido aleatorio de valores elevados. El resultado del *Dataset 2* es la suma del *Dataset 1 (Revisado)* y las nuevas imágenes generadas.

En la segunda iteración, el algoritmo genera 35 imágenes por cada imagen de cada clase de entrenamiento del *Dataset 1 (Revisado).* A cada una de estas imágenes se le ha añadido un ruido aleatorio, de valores medios. El resultado del *Dataset 3* es la suma del *Dataset 2* y las nuevas imágenes generadas.

En la tercera iteración, el algoritmo genera 135 imágenes por cada imagen de cada clase de entrenamiento del *Dataset 1 (Revisado).* A cada una de estas imágenes se le ha añadido un ruido aleatorio, de valores bajos. El resultado del *Dataset 4* es la suma del *Dataset 3* y las nuevas imágenes generadas. La distribución de los 3 *datasets* resultantes, se muestra en la Tabla 7.

Tabla 7. Distribución de los 3 datasets de entrenamiento resultantes

	Dataset 2	Dataset 3	Dataset 4
NºClases (Criptomonedas)	250	250	250
N.º Imágenes/Clase	280-290	1100-1200	4600-4700
Tamaño (GB)	10,2	29,6	118,5
Nº Total Imágenes	71.148	298.433	1.170.014

6.1.1.2. *Limpieza de los conjuntos de datos*

Es posible que algunas de las imágenes generadas sufran de un ruido excesivo, o incluso puede que el *zoom* deje de lado algunos logotipos. Los 3 conjuntos de datos mostrados en la Tabla 7 han sido corregidos manualmente. Únicamente se han eliminado las imágenes en las que no se distinguía ningún tipo de logo. Esa es la razón por la que hay menos imágenes de las generadas. Aun así, se ha respetado el balanceado de las clases, tal y como se puede observar en la distribución por clases de la Tabla 7.

6.2. Evaluar las alternativas del estado del arte

En este apartado, se han evaluado las diferentes alternativas existentes en el ámbito de la *Inteligencia Artificial* para elaborar el proyecto. Comenzando por las técnicas para conseguir el modelo requerido, siguiendo con la evaluación y los requerimientos técnicos necesarios para el entrenamiento de los modelos. Finalmente, se ha elegido el lugar para almacenar el modelo óptimo.

6.2.1. Elección de los algoritmos para el *Transfer Learning*

Analizando los diferentes estudios del estado del arte y, tras la formación recibida por UNIR en el ámbito de las *Redes Neuronales*, con un volumen de datos grande, es inviable el alto coste computacional que supone entrenar una de estas redes desde cero. Es por ello que se ha decidido entrenar los modelos mediante el uso del *Transfer Learning*.

Entre los modelos viables con mejores resultados para este propósito, se encuentran el *VGG16*, *VGG19*, *InceptionResnetV2* e *InceptionV3*. Con viables se hace referencia a la posibilidad de entrenamiento, ya que algunos algoritmos requirieren las imágenes de entrada en forma de vector

y, con la cantidad elevada de imágenes de los *datasets* de este proyecto, es inviable el procesamiento de las mismas.

6.2.1.1. *Entrenamiento del modelo mediante el uso de GPU*

El entrenamiento ha sido probado mediante un ordenador doméstico *Intel Core I7* con 8GB de RAM y 500GB de almacenamiento. El coste computacional y el tiempo necesario para probar los distintos algoritmos hacía inviable la utilización de este tipo de ordenadores.

Finalmente, se ha elegido *Google Colab* mediante el uso de GPU. El gran tamaño de los *datasets*, en algunos casos, imposibilita el uso del servicio gratuito de *Google Colab* siendo necesario contratar la versión de *Google Colab Pro.*

El servidor utilizado dispone de una GPU *NVIDIA-SMI 460.32.03* con 12GB de RAM *y* 166GB de almacenamiento total. El servicio gratuito asigna alrededor de 62GB utilizables y, en la versión Pro, alrededor de las 124GB.

6.2.2. Almacenamiento del modelo óptimo

La primera decisión se basa en elegir el lugar donde se va a almacenar el modelo. Tras algunas pruebas, y comprobar que los modelos de este proyecto tienen un tamaño superior a 70MB, se ha decidido almacenarlos en la nube. No es común ver aplicaciones móviles tan pesadas, y esta es la razón principal por la que se ha tomado esta decisión. La segunda razón se basa en la seguridad y el mantenimiento, ya que almacenar el modelo en la nube hace más robusta la aplicación teniendo en cuenta futuras actualizaciones. Esta decisión hace posible que todos los usuarios mantengan el mismo modelo en todo momento.

6.3. Obtener un modelo para la clasificación

Se van a realizar pruebas con los diferentes *datasets* generados, para obtener un modelo óptimo. En primer lugar, se utilizará el *Dataset 2*, es decir, el conjunto de datos más pequeño. De esta manera se obtendrá un *feedback* sobre el rendimiento de cada algoritmo elegido. En el caso de que los resultados no sean los esperados, se reducirá el número de clases intentando obtener un modelo que disponga de una calidad mínima, es decir, un *Accuracy* superior al 75%.

6.3.1. Clasificando 250 criptomonedas

Se ha comenzado con el *Dataset 2,* probando los diferentes algoritmos elegidos para predecir las 250 criptomonedas. Después, se ha probado con el *Dataset 3*.

6.3.1.1. *Resultados utilizando el Dataset 2 (250 Clases)*

Se ha entrenado el modelo utilizando el *Dataset 2.* El mejor rendimiento lo obtiene el algoritmo VGG16, con un *Accuracy* del 57,2% en aproximadamente 239 minutos, tal y como se muestra en la Tabla 8.

Tabla 8. Resultados utilizando el Dataset 2 clasificando 250 criptomonedas

D2 (250 clases)	**Training Time (Min)**	**Max (Accuracy)**
VGG16	239	57,2%
VGG19	245	56,6%
InceptionResnetV2	243	54,6%
InceptionV3	145	51,07%

Los resultados obtenidos mediante el *Dataset 2,* están lejos del Accuracy mínimo esperado del 75%. Por lo tanto, se va a entrenar el modelo utilizando el *Dataset 3*, el cual cuenta con una diversidad mayor en cada clase.

6.3.1.2. *Resultados utilizando el Dataset 3 (250 Clases)*

Tras realizar varias pruebas con el algoritmo que mejores resultados obtuvo con el *Dataset 2*, el VGG16, se obtiene una mejora hasta llegar a un 65,9% de *Accuracy*. Aun así, sigue lejos del rendimiento esperado, tal y como se muestra en la Tabla 9.

Tabla 9. Resultados utilizando el Dataset 3 clasificando 250 criptomonedas

D3 (250 clases)	**Training Time (Min)**	**Max (Accuracy)**
VGG16	1.110	65,9%

En este paso, se ha utilizado como único algoritmo el VGG16. El motivo reside en el tiempo de entrenamiento, ya que cada entrenamiento supera las 18 horas. Un tiempo de entrenamiento excesivo que ni la versión *Pro* de *Google Colab* permite. Muchos de los entrenamientos acaban

en cortes de conexión y con mensajes de alerta de *Google Colab*. Estos mensajes de *Google* alertan de limitaciones sobre el tiempo de uso de su plataforma.

Por lo tanto, siguiendo con el objetivo principal del proyecto, se decide reducir el número de clases a 100. Es decir, se descartarán las últimas 150 clases, de cada *dataset,* según el orden de *CoinMarketCap*.

6.3.2.Clasificando 100 criptomonedas

Se va a intentar obtener un modelo optimo que clasifique las 100 primeras criptomonedas de *CoinMarketCap,* mediante su logo. Por lo tanto, se probarán, nuevamente los algoritmos elegidos. Se comenzará actualizando la información de los conjuntos de datos y, después, se entrenarán los algoritmos utilizando los nuevos *datasets* resultantes.

6.3.2.1. *Actualización de los conjuntos de datos*

Tras actualizar los conjuntos de datos y disponer de los logotipos sobre las primeras 100 criptomonedas, la distribución resultante es la que se muestra en la Tabla 10. Cabe destacar que la nueva redistribución no afecta al número de imágenes por clase. El conjunto de validación también se mantiene.

Tabla 10. Distribución de los nuevos 3 datasets de entrenamiento resultantes

	Dataset 5	Dataset 6	Dataset 7
NºClases (Criptomonedas)	100	100	100
N.º Imágenes/Clase	280-290	1100-1200	4600-4700
Tamaño (GB)	3,8	12,6	59,8
Nº Total Imágenes	71.148	120.438	470.000

Tras la experiencia ganada en el paso anterior, hay una notable mejora en el rendimiento al utilizar el *Dataset 3,* respecto al *Dataset 2,* ya que este último contaba con una menor variedad de imágenes. Por lo tanto, teniendo en cuenta las limitaciones computacionales, se utilizarán los conjuntos de datos con más imágenes por clase, es decir, el *Dataset 6* y el *Dataset 7*.

6.3.2.2. *Resultados utilizando el Dataset 6 (100 clases)*

Los resultados de los entrenamientos, con el *Dataset 6* al reducir el número de clases a 100, siguen sin acercarse al mínimo requerido para ser aceptados en este proyecto. Los resultados máximos obtenidos mediante el algoritmo VGG16 se encuentra en la Tabla 11.

Tabla 11. Resultados utilizando el Dataset 6 clasificando 100 criptomonedas

D6 (100 clases)	**Training Time (Min)**	**Max (Accuracy)**
VGG16	453	63,8%

6.3.2.3. *Resultados utilizando el Dataset 7 (100 clases)*

A continuación, se comprobará el rendimiento del algoritmo VGG16 con el *dataset* que mayor variedad de logotipos tiene por clase, el *Dataset 7*. El mejor resultado obtenido llega al 74% de *Accuracy*, tal y como se muestra en la Tabla 12.

Tabla 12. Resultados utilizando el Dataset 7 clasificando 100 criptomonedas

D7 (100 clases)	**Training Time (Min)**	**Max (Accuracy)**
VGG16	349	74%

Tras numerosas pruebas con este algoritmo y el *Dataset 6,* siendo el que mejores resultados está ofreciendo, se sigue sin llegar al objetivo esperado. Aun así, al acercarse tanto al *Accuracy* mínimo requerido del 75%, se guardará dicho modelo y se desarrollará una última prueba.

6.3.3. Clasificando 90 criptomonedas

En esta última prueba, se eliminan 10 clases más del *Dataset 7*, clasificando las primeras 90 criptomonedas, teniendo en cuenta el ranking de *CoinMarketCap*.

6.3.3.1. *Actualización de los conjuntos de datos*

Tras eliminar 10 criptomonedas del *Dataset 7,* se crea el *Dataset 8*, la actualización de los datos se muestra en la Tabla 13.

Tabla 13. Distribución de los datos de entrenamiento del nuevo Dataset 8

	Dataset 8
NºClases (Criptomonedas)	90
N.º Imágenes/Clase	4600-4700
Tamaño (GB)	55,3
Nº Total Imágenes	423.000

6.3.3.2. *Resultados utilizando el Dataset 8 (90 clases)*

Los mejores resultados obtenidos con el algoritmo VGG16, entrenando el modelo con el *Dataset 8* y clasificando 90 criptomonedas diferentes, se muestran en la Tabla 14.

Tabla 14. Resultados utilizando el Dataset 8 clasificando 90 criptomonedas

D8 (90 clases)	**Training Time (Min)**	**Max (Accuracy)**
VGG16	282	77%

Aunque se hayan reducido sustancialmente las clases a predecir, el objetivo de mantener un mínimo nivel de calidad en las predicciones se mantiene. Por lo tanto, el modelo resultante es el que se va a incluir en la aplicación, ya que supera el 75% de *Accuracy previamente* establecido.

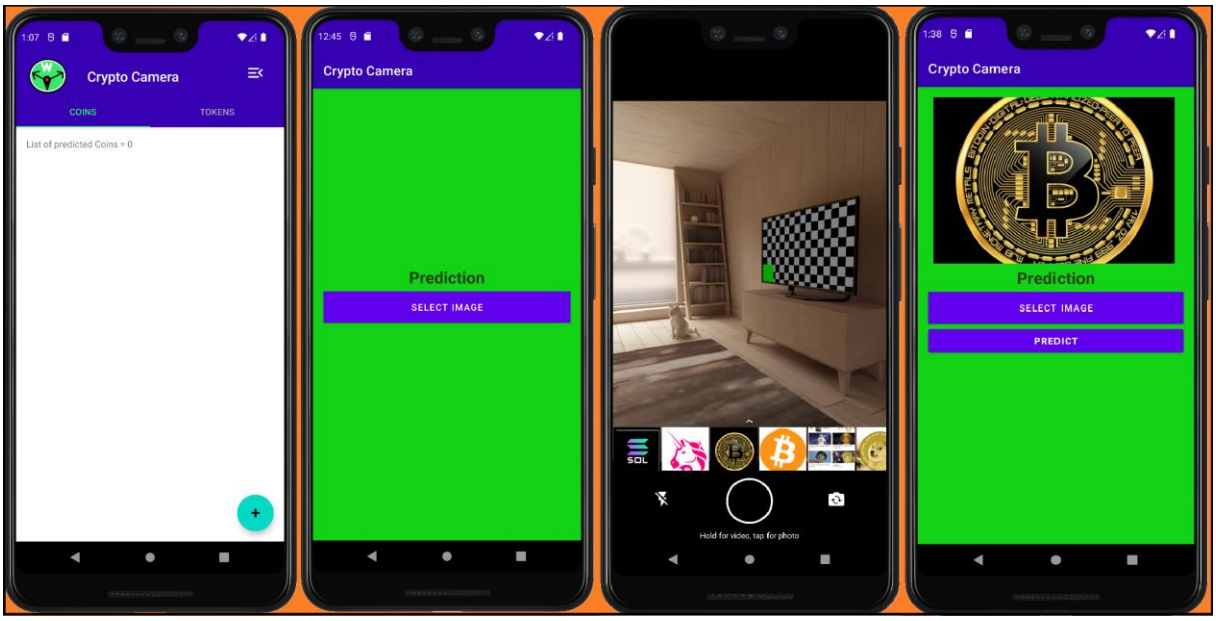
6.4. Desarrollar una aplicación móvil

La aplicación se ha desarrollado en *Android Studio* debido a la experiencia con esta herramienta. Sería conveniente desarrollar la aplicación también en IOS, pero debido al escaso tiempo disponible, es una tarea que se realizará más adelante.

A continuación, se mostrará la interfaz gráfica asociada a la aplicación. En esta interfaz gráfica se dispone de una pantalla principal, en la que el usuario dispondrá de los logotipos de las

criptomonedas y *tokens* que haya capturado anteriormente. Además, mediante el botón con el signo “+”, el usuario puede acceder a otra interfaz para clasificar nuevas criptomonedas, tal y como se observa en la Figura 47. El usuario dispone de la opción para cargar capturas previamente realizadas o, también, para realizar una foto de la criptomoneda en ese mismo momento. Esta última función está pensada para poder realizar capturas de los logotipos de las criptomonedas que aparezcan por la televisión, aunque puede ser usada en cualquier otro ámbito, tal y como se muestra en la Figura 47.

Figura 47. Interfaz gráfica de la aplicación desarrollada



6.4.1.Desarrollar las bases de datos

Se han desarrollado dos bases de datos para el correcto funcionamiento de la aplicación. La primera dentro del dispositivo y, la segunda, en *Firebase*.

La base de datos que se encuentra dentro del dispositivo móvil almacena los datos técnicos de cada criptomoneda. La segunda base de datos, alojada en *Firebase* y encriptada mediante funciones *hash*, almacena los datos de los usuarios. Por motivos de seguridad y para evitar la fuga de información confidencial de las personas que evalúan la aplicación, se evitará mostrar las estructuras de las bases de datos. Se ha tomado esta decisión teniendo en cuenta que este tipo de estructuras no requieren de *Inteligencia Artificial* y son irrelevantes para la evaluación del proyecto.

6.4.2.Conexión con *Firebase* para almacenar el modelo en la nube

El siguiente paso, consiste en conectar la aplicación de *Android Studio,* mediante la librería *TensorFlow Lite,* con *Firebase.*

En primer lugar, es necesario subir el modelo a *Firebase* en formato *TFLite.* El modelo óptimo del entrenamiento realizado en los pasos anteriores, se encuentra en formato H5. Por ese motivo, mediante un algoritmo en *Python*, se ha convertido el modelo del formato H5 al formato *TFLite*. Por último, se ha cargado el modelo resultante en *Firebase*.

6.4.3.Prueba de funcionamiento

En este apartado se mostrará la prueba básica de funcionamiento realizada para garantizar la funcionalidad en lo referente a la *Inteligencia Artificial*. Se mostrará una prueba de funcionamiento más extensa en la presentación oficial del proyecto.

En este caso, se ha probado con imágenes de diferentes criptomonedas, para su posterior predicción. La conexión con el servidor de *Firebase* y la predicción del modelo no han arrojado ningún tipo de error. Como prueba de ello, se muestra la predicción funcional de una de las criptomonedas, tal y como se refleja en la Figura 48.

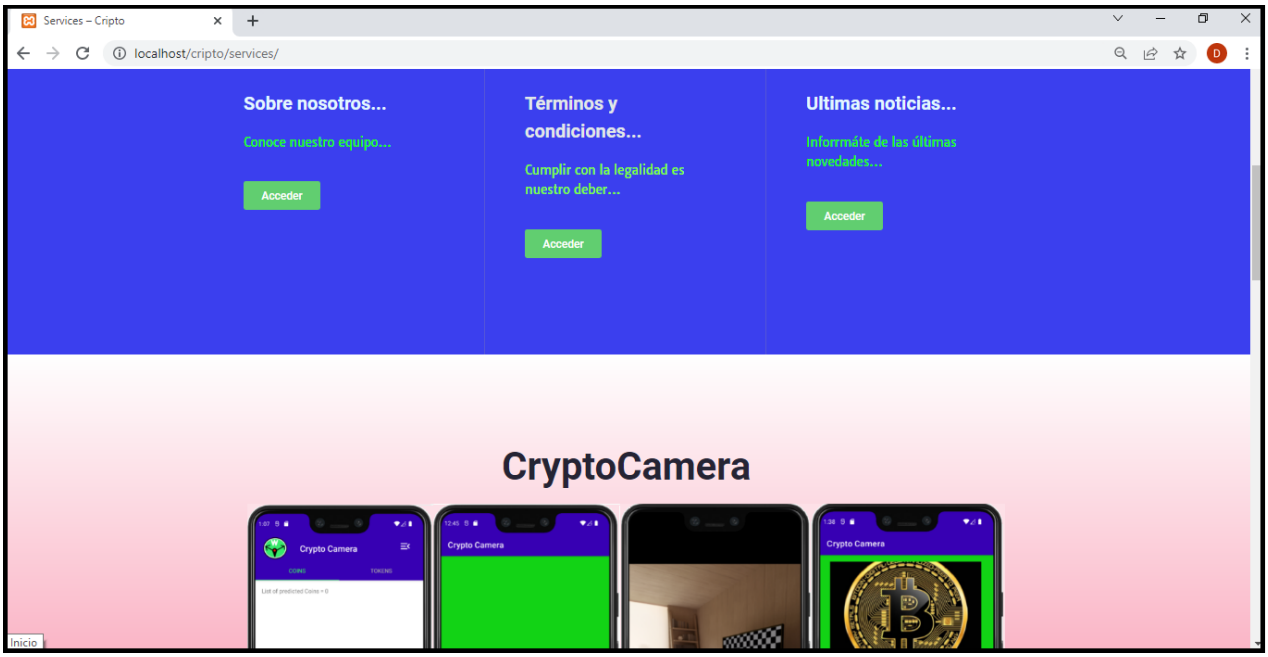
Figura 48. Prueba de funcionamiento con la criptomoneda Solana

6.5. Desarrollar una página web con los términos y condiciones legales

Se ha utilizado *Wordpress* para desarrollar una página web sencilla con los términos y condiciones legales asociados al tratamiento de datos. Es un paso necesario e imprescindible para poder subir la aplicación a *Google Play*.

Aun así, debido a la necesidad de cierto nivel de conocimiento para elaborar correctamente este tipo de trámites legales, se ha decidido esperar para ponerse en contacto con personas especializadas en este sector. Es por ello, que hasta no disponer de la seguridad de que se cumplan dichos requisitos, no se hará pública la aplicación. La página web provisional se muestra en la Figura 49.

Figura 49. Página web del proyecto



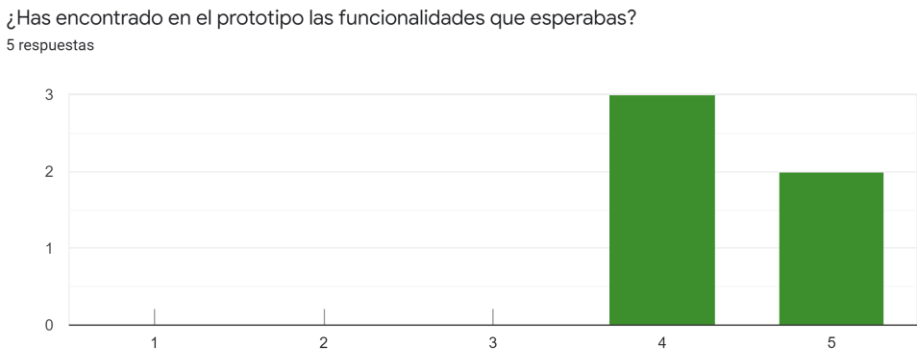
Además, es importante revisar y actualizar estos términos teniendo en cuenta la legalidad vigente de cada país. Es importante mencionar que esta aplicación no fomenta ni aconseja ningún tipo de inversión en este tipo de activos digitales. Las criptomonedas y los *tokens* son activos de alto riesgo, por lo tanto, es necesario disponer de cierto conocimiento para operar con los mismos.

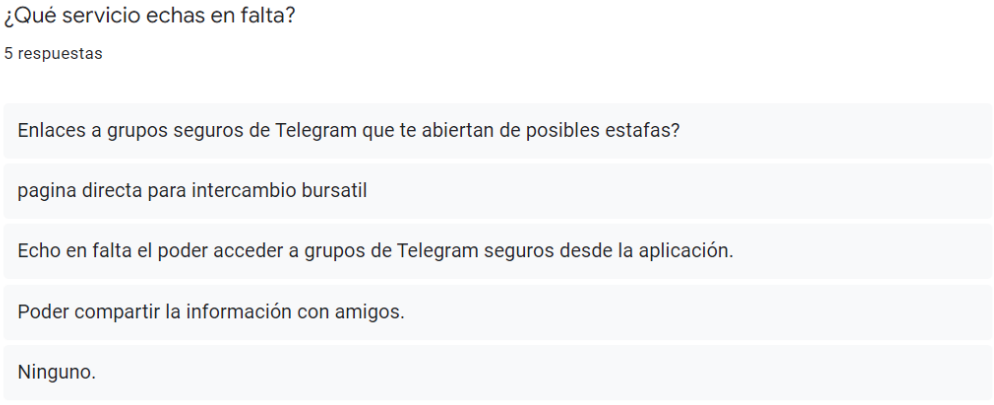
7. Evaluación de la aplicación móvil

En este apartado, se tratará de evaluar la aplicación con la colaboración de algunos usuarios y expertos del mundo de las criptomonedas. Es necesario resaltar la escasa formación profesional que existe en este ámbito. La mayor parte de los expertos no están directamente formados para este propósito. Por lo tanto, se intentará evaluar mediante expertos en esta materia. Se añadirán usuarios del mundo de las criptomonedas, aunque ninguno de ellos disponga de titulaciones oficiales pertenecientes a este ámbito.

La evaluación se ha realizado con 5 participantes mediante *Google Docs*. Entre los participantes se encuentran usuarios que han realizado transacciones con criptomonedas, creadores de *Smart Contracts*, *Artistas NFT* y *Gamers* de *juegos NFT* descentralizados. A todos ellos se les ha mostrado todas y cada una de las funcionalidades. Los resultados han sido satisfactorios en general con algunas propuestas de mejora. El resultado del documento ha sido el siguiente, el mostrado en la Figura 50.

Figura 50. Resultados de la evaluación





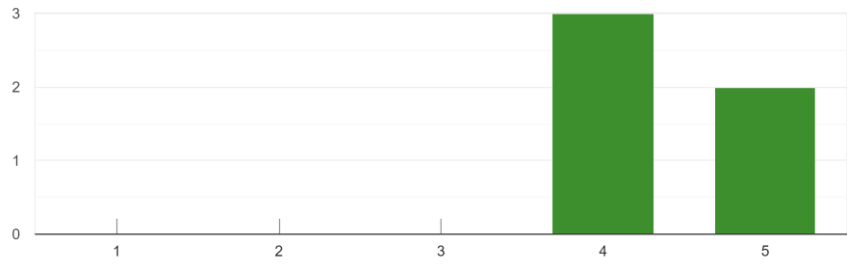
¿Crees que hay alguna operación que no es necesaria?

5 respuestas

No, todas las funciones e apartados tienen un sentido lógico.
negatibo
No.
No
No, todas son necesarias.

¿Encuentras lógica la estructura de navegación?

5 respuestas



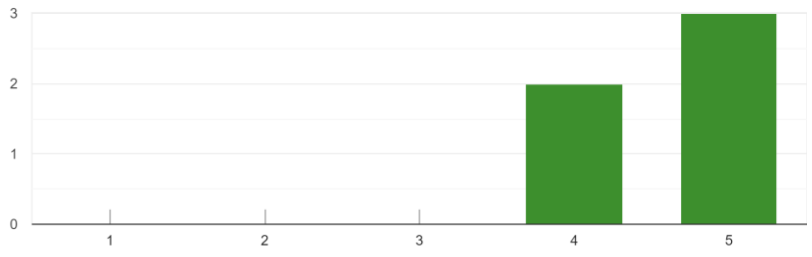
¿Se te ocurre algún modo de simplificar esta tarea?

5 respuestas

Haciendo un apartado principal en la aplicación que fuera una base de datos que recoge las predicciones de todos los usuarios y que muestra, mediante una clasificación las criptomonedas, si son seguras o si, por el contrario, son un posible fraude.
ninguno
No.
No
Está muy simplicada ya.

¿Te sentirías cómodo utilizando la aplicación?

5 respuestas



8. Conclusiones y trabajo futuro

En este apartado se abordarán las conclusiones obtenidas tras los resultados obtenidos de todos los apartados anteriores y, además, se expondrán nuevas líneas de investigación.

8.1. Conclusiones

La fiebre que rodea las criptomonedas está llegando a puntos enfermizos. Miles de transacciones comienzan por *Twitter* o por grupos de *Telegram*, dando credibilidad a cualquier tipo de información. Todo ello da lugar a que la gente se lance compulsivamente a comprar criptomonedas, basándose únicamente en su logotipo. Esto último trae como consecuencia el fraude mediante la suplantación de identidad.

Es complicado detectar si una criptomoneda puede ser fraudulenta o no. Aun así, la aplicación móvil resultante de este proyecto permite, a partir del logo de las criptomonedas capturadas mediante *Twitter* o grupos *Telegram*, obtener información relevante de las mismas. Por lo tanto, en este aspecto, cumple los objetivos mínimos determinados. Gracias a esta nueva forma de verificación mediante el logotipo, se le ofrece una seguridad adicional a este tipo de inversores. Aun así, el rendimiento de los algoritmos utilizados, mediante el uso del *Transfer Learning* debería ser mejor. En este caso, al ser una aplicación educativa, cumple con dichos objetivos.

En el ámbito de la *Inteligencia Artificial*, la primera conclusión que obtengo, es que se necesitan grandes conjuntos de datos para realizar *Transfer Learning* en clasificadores con más de 100 clases. En este sentido, tras probar con más de 4600 imágenes por cada clase, se han obtenido mejores resultados que con los conjuntos de datos menos variados. Aun así, los resultados no mejoran en exceso. El gasto computacional, en cambio, se dispara.

Con respecto al **primer objetivo específico**, que era el de “generar un *dataset* adecuado”, en el trabajo se ha mostrado como se han generado diferentes *datasets,* con el objetivo de experimentar de una manera más exhaustiva. Hubo algunas dificultades en la limpieza de los conjuntos, dado el gran volumen de imágenes que había que tratar, pero, aun así, el objetivo se ha alcanzado correctamente.

Haciendo referencia al **segundo objetivo específico**, que era el de “evaluar todas las alternativas del estado del arte”, se ha razonado correctamente el uso de cada alternativa asociada al ámbito de la *Inteligencia Artificial*. Hubo un problema a resolver con la elección del almacenamiento del modelo en *Firebase,* ya que el límite máximo de almacenaje disponible es de 40MB. Aun así, tras

reducir el número de clases y cambiar el formato del modelo a *TFLite*, se redujo el peso del modelo. Por lo tanto, se solucionó el problema y se da por cumplido el objetivo.

El **tercer objetivo específico**, que era el de "obtener un modelo para la clasificación", ha sido uno de los más costosos de realizar, computacionalmente hablando. Los límites que tiene *Google Colab* junto con la gran cantidad de datos disponibles para entrenar han dificultado mucho este objetivo. Algunos entrenamientos duraban más de 18 horas cuando repentinamente se detenían, por las limitaciones de *Google,* desperdiciando el trabajado realizado hasta el momento. Aun así, se ha conseguido un modelo aceptable, superando el 75% de *Accuracy* mínimamente establecido.

El **cuarto objetivo específico**, que era el de "desarrollar una aplicación móvil", se ha realizado sin problema, debido a la experiencia previa programando en *Android Studio*.

Con respecto al **quinto objetivo específico**, que era el de "desarrollar una página web con los términos y condiciones legales", se ha diseñado una página web en *wordpress* pero no se ha publicado aún. El motivo ha sido la falta de asesoría legal disponible durante el tiempo que se ha realizado el proyecto. Además, las nuevas leyes que regulan la publicidad cripto hacen necesaria la actualización de la página. Por lo tanto, este objetivo queda pendiente de completar.

El **sexto objetivo**, que era el de "evaluar la solución propuesta", se ha realizado correctamente, logrando una puntuación más que aceptable. Además, se han recibido propuestas de mejora que se tendrán en cuenta de cara a la presentación.

Para concluir, en lo referente a la *Inteligencia Artificial*, tras experimentar con las diferentes arquitecturas destacadas del *Transfer Learning* y utilizando distintos conjuntos de datos con distintos niveles de *Data Augmentation*, los resultados no han sido tan buenos como se esperaba. Se necesita una gran cantidad de datos para entrenar este tipo de modelos, es decir, cuando las clases a predecir son mayores a 100. Aun así, los resultados no han llegado al 75% de *Accuracy*. Se podrían generar u obtener más imágenes por cada clase, pero, el costo computacional, con los medios actuales, complicaría mucho el entrenamiento. Por lo tanto, el ámbito de las criptomonedas sigue siendo el medio ideal para seguir investigando y mejorando las técnicas actuales de clasificación de logotipos mediante *Transfer Learning.*

8.2. Líneas de trabajo futuro

En esta sección, se exponen las 3 diferentes líneas futuras de investigación que han surgido del estudio y de los resultados obtenidos en los capítulos anteriores.

La primera línea de investigación se basa en ampliar el conjunto de datos inicial, es decir, en disponer de 40 imágenes más por cada clase, obteniendo un total de 80 imágenes por cada criptomoneda. De esta manera, se podría obtener un conjunto de datos inicial más variado y se podrían poner a prueba, nuevamente, los algoritmos utilizados en este proyecto mediante el *Transfer Learning*. Finalmente, se podrían comparar los resultados obtenidos con los de este trabajo.

La segunda línea de investigación se basa en etiquetar y delimitar los logotipos del conjunto de datos ya disponible en este proyecto. Se plantea cambiar el tipo de problemática a resolver, de clasificación de imágenes a detección y reconocimiento de imágenes. Se podrían etiquetar mediante el software de código abierto *LabelIMG*. De esta forma, se pondrían a prueba los algoritmos utilizados en este trabajo mediante *Transfer Learning*, pero en este caso, para el reconocimiento de logos.

La tercera línea de investigación se centraría en utilizar *Generative Adversarial Networks* (GAN) para generar conjuntos de datos. Es posible entrenar este tipo de *Redes Neuronales* para obtener nuevas imágenes y que parezcan reales. Tras analizar que uno de los problemas generales se basa en la falta de *datasets*, y que esto puede afectar a la generalización de las *Redes Neuronales*, sería interesante poder crear una GAN por cada criptomoneda. De esta manera, cada GAN podría ser entrenada para generar nuevas imágenes de cada clase. Sería interesante comparar los resultados obtenidos, con los *datasets* generados en este proyecto, con otros nuevos *datasets* generados por diferentes GAN. De esta manera, se podría comprobar la utilidad que podrían llegar a tener las GAN en la generación de nuevos conjuntos de datos. Esto podría ser realmente útil para comprobar la diferencia entre los resultados obtenidos con los conjuntos de datos ampliados mediante *Data Augmentation* y los que se amplíen mediante las GAN.

9. Bibliografía

Abu, M. A., Indra, N. H., Halim, A., Rahman, A., Sapiee, A., y Ahmad, I. (2019). A study on image classification based on deep learning and tensorflow. *International Journal of Engineering Research and Technology, 12*(4)

Adidas. (2021). *Dentro del metaverse.* https://www.adidas.es/metaverse

Alake, R. (2021). *Deep learning: GoogLeNet explained.* https://towardsdatascience.com/deep-learning-googlenet-explained-de8861c82765

Angelo, M. D., & Salzer, G. (2021). Tokens, types, and standards: Identification and utilization in ethereum. *International Journal of Data Science and Analytics,* , 1-20. 10.1109/DAPPS49028.2020.00-11

Ante, L. (2021). The non-fungible token (NFT) market and its relationship with bitcoin and ethereum. *SSRN Electronic Journal,* (20)10.2139/ssrn.3861106

Antonio Pierro, G., & Rocha, H. (2019). (2019). The influence factors on ethereum transaction fees. Paper presented at the - *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB),* 24-31. 10.1109/WETSEB.2019.00010

Attal, M. (2022). *¿Qué es el transfer learning?* https://datascientest.com/es/que-es-el-transfer-learning

Aysan, A. F., Khan, A. U. I., y Topuz, H. (2021). Bitcoin and altcoins price dependency. *Risks (Basel), 9*(4), 1-13. 10.3390/risks9040074

Bambrough, B. (2021). *Bitcoin 'Undermining the Dollar'—Hillary clinton issues A surprise crypto warning as el salvador helps the bitcoin price suddenly surge.* https://www.forbes.com/sites/billybambrough/2021/11/21/undermining-the-dollar-hillary-clinton-issues-a-surprise-crypto-warning-as-the-bitcoin-price-suddenly-surges-back/

Billy, B. (2021). *'Sounds like BS'—Elon musk slams radical new tech and boosts the price of dogecoin as bitcoin and ethereum struggle.* https://www.forbes.com/sites/billybambrough/2021/12/02/sounds-like-bs-elon-musk-slams-radical-new-tech-and-boosts-the-price-of-dogecoin-as-bitcoin-and-ethereum-struggle/

Binance. (2021). *Bitcoin exchange | exchange de criptomonedas.* https://www.binance.com

Brummer, C. (2019). *Cryptoassets: Legal, regulatory, and monetary perspectives*. Oxford University Press. 10.1093/oso/9780190077310.001.0001

Cano, J. E. S. (2020). The technological innovation of the blockchain and its impact on the energy sector/la innovacion tecnologica del blockchain y su impacto en el sector energetico. *Panorama Economica, 16*(31), 157.

Cassimiro, G. (2021). *Transfer learning with VGG16 and keras.* https://towardsdatascience.com/transfer-learning-with-vgg16-and-keras-50ea161580b4

CMC Alexandria. (2021). *Rug pull.* https://coinmarketcap.com/alexandria/glossary/rug-pull

CoinMarketCap. (2021). *Cryptocurrency prices, charts and market capitalizations.* https://coinmarketcap.com/

Crypto Logos. (2022). *Crypto logos - cryptocurrency logo files (.SVG & .PNG) download.* https://cryptologos.cc

Ethereum. (2021). *Ethereum whitepaper.* https://ethereum.org

eToro. (2021). *¡El bitcoin ha llegado!* https://www.etoro.com/es/news-and-analysis/etoro-updates/el-bitcoin-ha-llegado/

Feinstein, B. D., & Werbach, K. (2021). The impact of cryptocurrency regulation on trading markets. *J Financ Regul, 7*(1), 48-99. 10.1093/jfr/fjab003

Firebase. (2022). *Usa un modelo de TensorFlow lite para las inferencias con el kit de AA en android.* https://firebase.google.com/docs/ml-kit/android/use-custom-models?hl=es-419

Galvez, J. F., Mejuto, J. C., y Simal-Gandara, J. (2018). Future challenges on the use of blockchain for food traceability analysis. *TrAC, Trends in Analytical Chemistry (Regular Ed.), 107*, 222-232. 10.1016/j.trac.2018.08.011

GanttProject. (2022). *GanttProject home.* https://www.ganttproject.biz/

Gao, B., Wang, H., Xia, P., Wu, S., Zhou, Y., Luo, X., y Tyson, G. (2020). Tracking counterfeit cryptocurrency end-to-end. *Proceedings of the ACM on Measurement and Analysis of Computing Systems, 4*(3), 1-28. 10.1145/3428335

Garcia-Dominguez, M., Dominguez, C., Heras, J., Mata, E., y Pascual, V. (2020). FrImCla: A framework for image classification using traditional and transfer learning techniques. *IEEE Access, 8*, 1. 10.1109/ACCESS.2020.2980798

GlooMaps. (2022). *GlooMaps.* https://www.gloomaps.com/

Google Colab. (2022). *Colaboratory.* https://research.google.com/colaboratory/faq.html#:~:text=How%20long%20can%20notebooks%20run,left%20idle%20for%20too%20long.

Herman, A. (2021). *Will quantum computers burst the bitcoin boom?* https://www.forbes.com/sites/arthurherman/2021/11/09/will-quantum-computers-burst-the-bitcoin-boom/

Hoi, S. C. H., Wu, X., Liu, H., Wu, Y., Wang, H., Xue, H., y Wu, Q. (2015). LOGO-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks. https://arxiv.org/abs/1511.02462

Hou, Q., Min, W., Wang, J., Hou, S., Zheng, Y., y Jiang, S. (2021). FoodLogoDet-1500: A dataset for large-scale food logo detection via multi-scale feature decoupling network. *Proceedings of the 29th ACM International Conference on Multimedia,* 10.1145/3474085.3475289

Howell, S. T., Niessner, M., y Yermack, D. (2020). Initial coin offerings: Financing growth with cryptocurrency token sales. *The Review of Financial Studies, 33*(9), 3925-3974. 10.1093/rfs/hhz131

Jeger, C., Rodrigues, B., Scheid, E., & Stiller, B. (Nov 02, 2020). (Nov 02, 2020). Analysis of stablecoins during the global COVID-19 pandemic. Paper presented at the 30-37. 10.1109/BCCA50787.2020.9274450 https://ieeexplore.ieee.org/document/9274450

Kamps, J., & Kleinberg, B. (2018). To the moon: Defining and detecting cryptocurrency pump-and-dumps. *Crime Science, 7*(1), 1-18. 10.1186/s40163-018-0093-5

Karantias, K., Kiayias, A., & Zindros, D. (Febrero 10-14, 2020). (Febrero 10-14, 2020). Proof-of-burn. Paper presented at the *Financial Cryptography and Data Security : 24th International Conference,* 523-540.

Karimi, M., & Behrad, A. (Mar 2019). (Mar 2019). Logo recognition by combining deep convolutional models in a parallel structure. Paper presented at the 216-221. 10.1109/PRIA.2019.8786032 https://ieeexplore.ieee.org/document/8786032

Kitsantas, T., Vazakidis, A., & Chytis, E. (Jul 2, 2019). (Jul 2, 2019). A review of blockchain technology and its applications in the business environment. Paper presented at the *International Conference on Enterprise, Systems, Accounting, Logistics & Management,*

Konkiewicz, M. (2021). *How to successfully add large data sets to google drive.* https://towardsdatascience.com/how-to-successfully-add-large-data-sets-to-google-drive-130beb320f1a

kumar, K. (2021). *Transfer learning with VGG16 and VGG19, the simpler way!* https://koushik1102.medium.com/transfer-learning-with-vgg16-and-vgg19-the-simpler-way-ad4eec1e2997

Kurtulmus, A. B., & Daniel, K. (2018). Trustless machine learning contracts; evaluating and exchanging machine learning models on the ethereum blockchain. https://arxiv.org/abs/1802.10185

LabelImg. (2022). *LabelImg.* https://github.com/tzutalin/labelImg

Le, N. (2021, Sep 19,). Applying machine learning to solve brand/logo detection problem. https://engineering.tiki.vn/applying-machine-learning-to-solve-brand-logo-detection-problem/

Liebau, D., & Schueffel, P. (2019). Crypto-currencies and ICOs: Are they scams? an empirical study. *SSRN Electronic Journal,* 10.2139/ssrn.3320884

Liu, Y., Tsyvinski, A., y Wu, X. (2019). Common risk factors in cryptocurrency. *SSRN Electronic Journal,* 10.2139/ssrn.3379131

Mehta, P., Lee, A., Lee, C., Balazinska, M., y Rokem, A. (2018b). *Multilabel multiclass classification of OCT images augmented with age, gender and visual acuity data*. Cold Spring Harbor Laboratory. 10.1101/316349

Milton-Barker, A. (2019). *Inception V3 deep convolutional architecture for classifying acute myeloid/lymphoblastic leukemia.* https://www.intel.com/content/www/us/en/developer/articles/technical/inception-v3-deep-convolutional-architecture-for-classifying-acute-myeloidlymphoblastic.html

Miraz, M. H., & Ali, M. (2018). Applications of blockchain technology beyond cryptocurrency. *Annals of Emerging Technologies in Computing (AETiC), 2*(1)

Momtaz, P. P. (2020). Initial coin offerings. *PloS One, 15*(5), e0233018. 10.1371/journal.pone.0233018

Momtaz, P. P. (2021). Security tokens. *SSRN Electronic Journal,* 10.2139/ssrn.3865233

Olasehinde Bisola. (2021). *Coinmarketcap historical data.* https://www.kaggle.com/bizzyvinci/coinmarketcap-historical-data

PancakeSwap. (2021). *Home | PancakeSwap.* https://pancakeswap.finance/

Pérez, H. (2022). *Samsung estrena nueva tienda virtual en el metaverso de decentraland.* https://www.diariobitcoin.com/metaverso/samsung-estrena-nueva-tienda-virtual-en-el-metaverso-de-decentraland/

PyTorch. (2022). *PyTorch.* https://www.pytorch.org

Rafael, G. D. (2021). *El staples center cambia su nombre pro el de una criptomoneda.* https://as.com/meristation/2021/11/18/betech/1637261214_154571.html

Raj, B. (2020). *A simple guide to the versions of the inception network.* https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202

Read, S. (2021). *'Irresponsible' london underground bitcoin advert banned.* https://www.bbc.com/news/business-57254318

RobotFlow. (2022). *Roboflow: Give your software the power to see objects in images and video.* https://roboflow.ai

Romero, N., Gutoski, M., Hattori, L., y Lopes, H. S. (2017). *The effect of data augmentation on the performance of convolutional neural networks*. ABRICOM. 10.21528/cbic2017-51

Roose, K. (2021). *Crypto is cool. now get on the yacht.* https://www.nytimes.com/2021/11/05/technology/nft-nyc-metaverse.html

Rosebrock Adrian. (2021). *PyTorch: Transfer learning and image classification.* https://www.pyimagesearch.com/2021/10/11/pytorch-transfer-learning-and-image-classification/

Roshanzamir[a], A., Weerakkody[a], V., Rana[a], N., Rahmati[b], M., y Shajari[b], M. (2020). Blockchain and image processing to reinforce provenance in the narrative of a handwoven carpet. *International Journal on Advances in Internet Technology, 12*, 61-75.

Ruiz, P. (2018). *Understanding and visualizing DenseNets.* https://towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a

Saha, S. (2018). *A comprehensive guide to convolutional neural networks — the ELI5 way.* https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Sahel, S., Alsahafi, M., Alghamdi, M., y Alsubait, T. (2021). Logo detection using deep learning with pretrained CNN models. *Engineering, Technology & Applied Science Research, 11*(1), 6724-6729. 10.48084/etasr.3919

Samieifar, S., & Baur, D. G. (2021). Read me if you can! an analysis of ICO white papers. *Finance Research Letters, 38*, 101427. https://doi.org/10.1016/j.frl.2020.101427

Sánchez López, A. (2019a). *Development of a food image classification system based on transfer learning with convolutional neural networks.* Unpublished manuscript. https://explore.openaire.eu/search/publication?articleId=doajarticles::df5a1de5386b9e44cf1c4eb8eb4669c5

Sánchez López, A. (2019b). *Publication - development of a food image classification system based on transfer learning with convolutional neural networks.* http://vps161.cesvima.upm.es/en/about-us/people?view=publication&task=show&id=529

Sartipi, F. (2021). Publicizing construction firms by cryptocurrency. *Journal of Construction Materials, 2*(3), 3. 10.36756/JCM.v2.3.1

Schär, F. (2021). Decentralized finance: On blockchain- and smart contract-based financial markets. *Review - Federal Reserve Bank of St.Louis, 103*(2), 153-174. 10.20955/r.103.153-74

Shaikh, R. (2022). *Deploying A deep learning model on mobile using TensorFlow and react.* https://towardsdatascience.com/deploying-a-deep-learning-model-on-mobile-using-tensorflow-and-react-4b594fe04ab

Shaoqing Ren, Kaiming He, Girshick, R., y Jian Sun. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(6), 1137-1149. 10.1109/TPAMI.2016.2577031

Sun, Z. (2021). *Adidas ingresa al metaverso con asociaciones de NFT.* https://es.cointelegraph.com/news/adidas-enters-the-metaverse-with-nft-partnerships

Takimoglu, A. (2021). *What is data augmentation? techniques & examples in 2022.* https://research.aimultiple.com/data-augmentation/

Tensorflow. (2017). *English: Vector logo of tensorflow.* https://commons.wikimedia.org/wiki/File:Tensorflow_logo.svg

TensorFlow. (2021). *Plataforma de extremo a extremo de código abierto para el aprendizaje automático.* https://www.tensorflow.org/?hl=es-419

TokenSniffer. (2021). *Token sniffer.* https://tokensniffer.com/

Tran, K. (2020). *What is PyTorch?* https://towardsdatascience.com/what-is-pytorch-a84e4559f0e3

Google Trends. (2021). Descubre qué está buscando el mundo. https://trends.google.es/trends/?geo=ES

Tsang, S. (2019). *Review: DenseNet — dense convolutional network (image classification).* https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803

Tüzkö, A., Herrmann, C., Manger, D., & Beyerer, J. (2018). (2018). Open set logo detection and retrieval. Paper presented at the 10.5220/0006614602840292 http://publica.fraunhofer.de/documents/N-510038.html

ultralytics. (2022). *Train custom data.* https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data

Wallet, T. (2021). *La mejor billetera de criptomonedas | billetera ethereum | billetera ERC20.* https://trustwallet.com/es/

Wang, J., Min, W., Hou, S., Ma, S., Zheng, Y., y Jiang, S. (2020). LogoDet-3K: A large-scale image dataset for logo detection. https://arxiv.org/abs/2008.05359

Xia, P., wang, H., Gao, B., Su, W., Yu, Z., Luo, X., Zhang, C., Xiao, X., y Xu, G. (2021). Demystifying scam tokens on uniswap decentralized exchange. https://arxiv.org/abs/2109.00229

Yamashita, R., Nishio, M., Do, R. K. G., y Togashi, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights into Imaging, 9*(4), 611-629. 10.1007/s13244-018-0639-9

Young, J. (2021). *Crypto.com buys naming rights to lakers' staples center in a $700 million deal.* https://www.cnbc.com/2021/11/17/cryptocom-buys-naming-rights-to-lakers-staples-center-in-a-700-million-deal-.html

1. Anexos

Se incluye como anexo el artículo científico asociado al trabajo realizado.

1.1. Anexo I. Mejores resultados de los entrenamientos

Este anexo incluye gráficamente los resultados más relevantes de todos los entrenamientos realizados.

1.2. Anexo II. Capturas de pantalla de la aplicación

Este anexo incluye algunas capturas de pantalla de la aplicación elaborada en Android Studio.

1.3. Anexo. Artículo de investigación

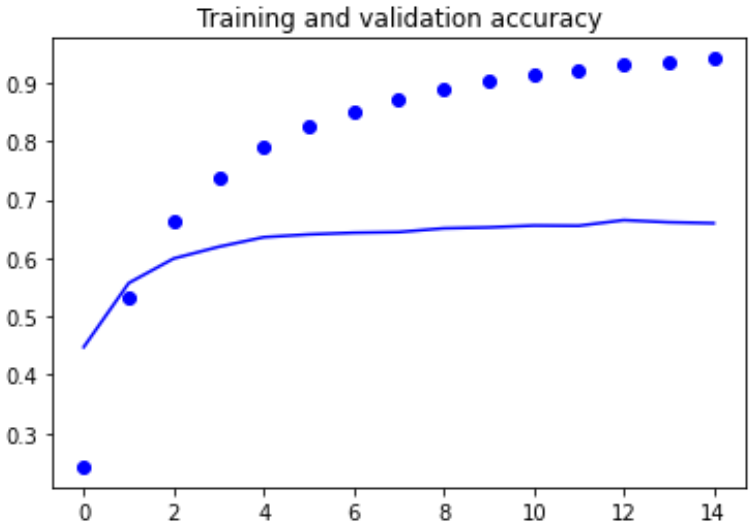
Este anexo incluye un artículo de investigación que resume el trabajo realizado en el proyecto: "*Aplicación móvil para evitar fraudes en criptomonedas mediante la detección de logos".* El artículo dispone de los principales resultados obtenidos.

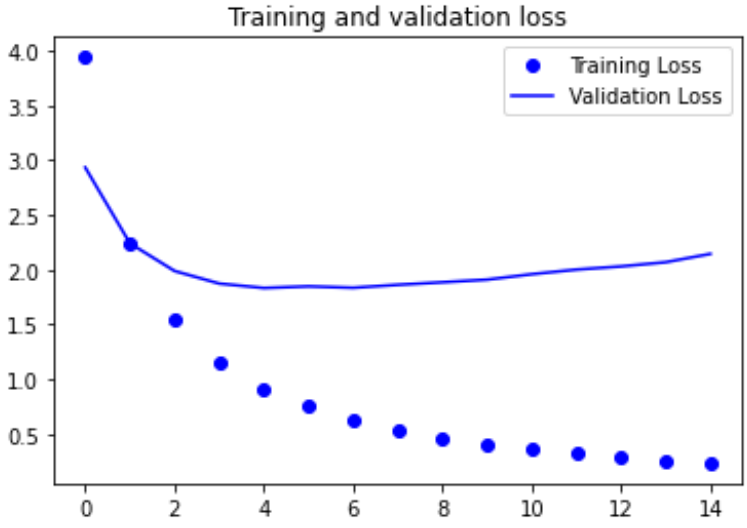
1.1. Anexo I. Mejores resultados de los entrenamientos

En este *Anexo* se mostrarán gráficamente algunos de los mejores resultados en las ejecuciones completadas durante los entrenamientos. Se mostrará la comparativa de las métricas de *Accuracy* y *Loss* entre los conjuntos de entrenamiento y validación. Los mejores resultados fueron obtenidos mediante la arquitectura de *Transfer Learning* VGG16.

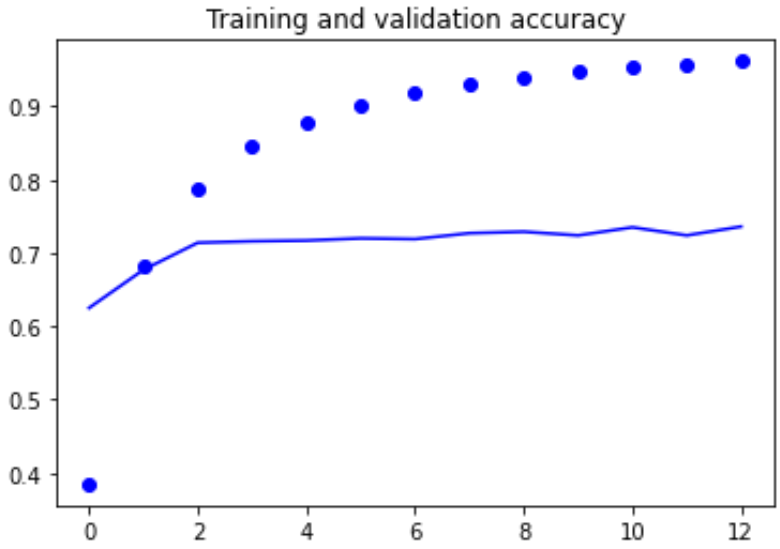
El objetivo de este anexo es el de demostrar, gráficamente, que no existe una mejora realmente tan significativa al reducir a más de la mitad el número de clases a predecir, con la arquitectura VGG16.

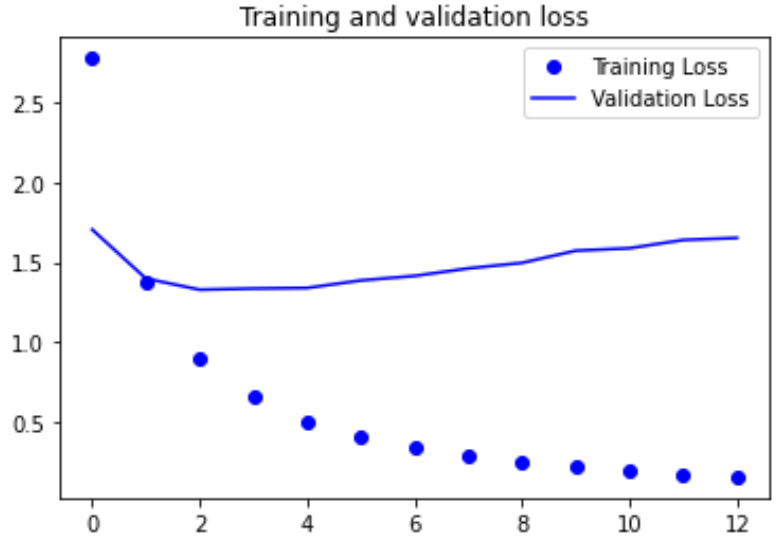
1.1.1. Clasificando 250 criptomonedas (VGG16)





1.1.2.Clasificando 100 criptomonedas (VGG16)

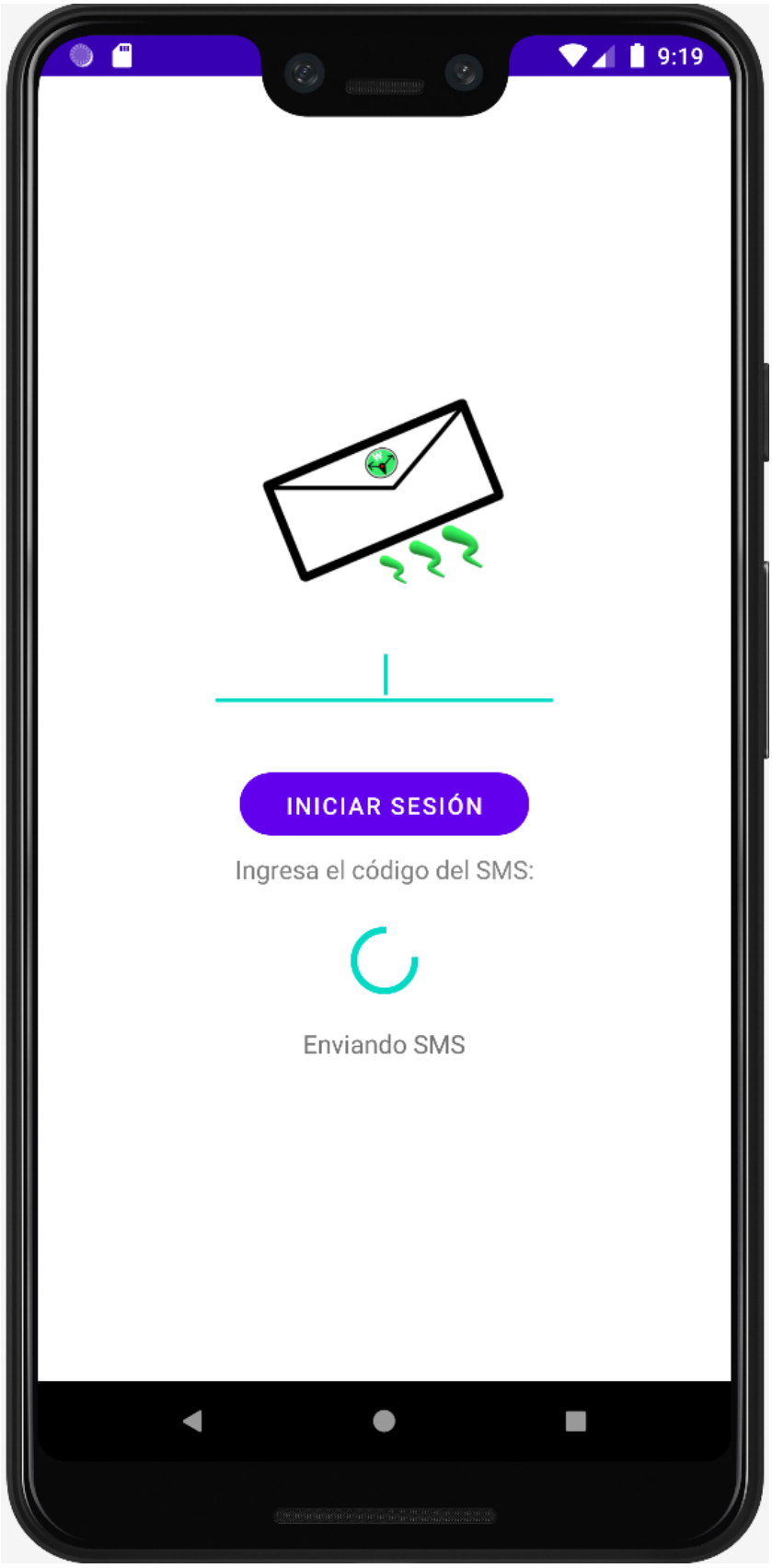




1.2. Anexo II. Capturas de pantalla de la aplicación

En este anexo se incluyen todas las capturas de pantalla de la aplicación:

(Interfaz-Register)



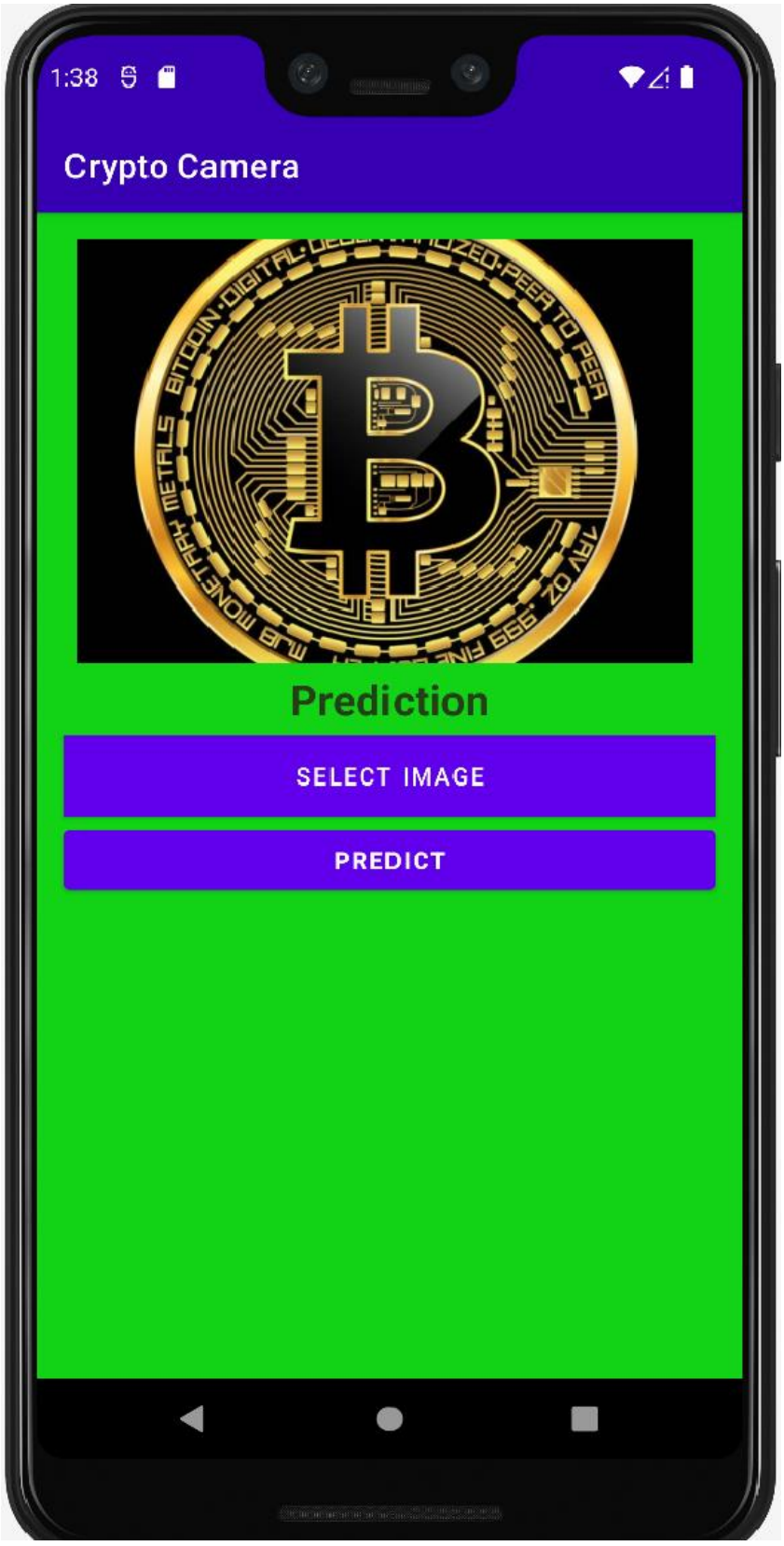
(Interfaz-Login)

(Interfaz-Session)

(Interfaz-Profile)

(Interfaz-Prediction)

(Interfaz-Capture)



(Interfaz-Registro)

Transfer-Learning para evitar fraudes con criptomonedas mediante LD

David Sánchez Alonso

Universidad Internacional de la Rioja, Logroño (España)

Fecha: 02/03/2022

RESUMEN

La detección y clasificación de logotipos de criptomonedas puede ayudar a reducir algunas de las estafas asociadas a la falsificación de *Smart-Contracts*. La gran variedad de criptomonedas que existen, puede ser usada para investigar la creación de un modelo, que detecte y clasifique las criptomonedas más importantes. Este proyecto presenta el desarrollo de una aplicación móvil que converge en un análisis comparativo de algunas de las mejores arquitecturas de *Transfer Learning* (VGG16, VGG19, *InceptionResnetV2* e *InceptionV3*) del estado del arte. Se hacen pruebas con diferentes conjuntos de datos, variando el número de logotipos por clase y con variedades de criptomonedas que van desde las 90, hasta las 250. Se obtuvieron los mejores resultados con la arquitectura VGG16 al clasificar 90 clases, obteniendo un *Accuracy* del 77%. Lo que indica que aún queda mucho por investigar, en el ámbito del *Transfer Learning*, cuando el número de clases comienza a ser significativo.

PALABRAS CLAVE

Cryptocurrency, Logo-Detection, Pump-and-Dump, Shitcoin, Transfer-Learning

I. INTRODUCCIÓN

Las transacciones con criptomonedas están cambiando la manera de invertir de muchas personas en todo el mundo. La fiebre que rodea este tipo de activos digitales llega hasta tal punto que algunas de las empresas y bancos más importantes del mundo han decidido adaptarse a esta nueva revolución. Por ejemplo, *Ana Botín*, presidenta de uno de los bancos sistémicos más importantes del mundo, como es el *Banco Santander*, se posiciona a favor de las criptomonedas y plantea ofrecer productos de inversión cripto a corto plazo [1].

Miles de transacciones comienzan por *Twitter* o por grupos de *Telegram*, pasándose información sobre qué criptomonedas les van a permitir obtener un mejor rendimiento económico. Todo ello da lugar a que la gente se lance compulsivamente a comprar criptomonedas, basándose únicamente en su logotipo, y esto da lugar al fraude mediante la suplantación de identidad.

Existe la necesidad real de añadir formas de verificación para las criptomonedas, el logotipo, dada su importancia, podría ser una de ellas. Además, en lo referente a la *Inteligencia Artificial*, existe un nicho de investigación que engloba la problemática asociada al reconocimiento y clasificación de logotipos, en el que una gran diversidad de clases podría afectar al rendimiento de las mejores arquitecturas de *Transfer Learning* del estado del arte. Por lo tanto, las criptomonedas, teniendo en cuenta la gran diversidad de ellas que existen y sus diferentes representaciones, pueden ser la opción ideal para investigar, analizar y probar este tipo de soluciones.

Se plantea una aplicación que va a permitir, a partir del logo de las criptomonedas capturadas mediante *Twitter* o grupos *Telegram*, obtener información relevante y oficial de las criptomonedas más demandadas. Gracias a esta nueva forma de verificación mediante el logotipo, se le ofrecerá una seguridad adicional a este tipo de inversores.

Entre las arquitecturas para *Transfer Learning* analizadas, algunas de las que mejores rendimientos ofrecen son VGG16, VGG19, *InceptionResnetV2* e *InceptionV3*. Los experimentos documentados se han realizado en un grupo reducido de *datasets* y, en la mayoría de los casos, con una escasa diversidad de clases. Por lo tanto, se propone realizar diferentes pruebas con grandes conjuntos de datos que puedan poner realmente a prueba este tipo de arquitecturas.

II. ESTADO DEL ARTE

En este apartado se va a analizar el estado del arte relacionado con las principales tecnologías de LD (*Logo Detection*), comenzando con una breve introducción para acabar centrando la atención en los logotipos y analizando las carencias y dificultades que entraña este apartado de la Inteligencia Artificial. Se prestará especial atención a las investigaciones más recientes ya a las técnicas y los algoritmos con mejores resultados. Se explicarán las arquitecturas asociadas a estos algoritmos, pero sin profundizar en exceso, ya que, implementarlos individualmente, queda lejos de la perspectiva de este proyecto.

Se analizarán los conjuntos de datos más relevantes y utilizados en LD. Tras ello, se dará paso a las soluciones actuales de la Inteligencia Artificial para la resolución de esta problemática. Además, se mostrarán las limitaciones más más destacadas asociadas a la detección de logos, mostrando las tecnologías que pueden ayudar a paliarlas. Después, se explicará la técnica del *Transfer Learning*, la cual ha revolucionado el sector de la detección de logos, aumentando el rendimiento asociado a la problemática de los mismos.

1. Conjuntos de datos

En este apartado se van a analizar los conjuntos de datos más utilizados para la detección de logos, sus características y, con ello, sus limitaciones. Además, se explicará alguna de las técnicas actuales para la obtención de nuevos *datasets*.

Gran parte de las investigaciones de los últimos años, en la resolución de la problemática de la detección de logos, clasifican correctamente una pequeña cantidad de clases. Este tipo de investigaciones logran buenos resultados, en muchos de los casos, con una precisión cercana al 98%. El uso de *datasets* con un número limitado de clases se debe, principalmente, al límite computacional y al escaso número de conjuntos de datos etiquetados disponibles [11].

Los conjuntos de datos más utilizados en el ámbito de LD se muestran en la **Tabla 1**. En esta tabla se muestra el nombre de los *datasets*, el número de clases que tienen, el número de imágenes totales, el número de objetos totales y su disponibilidad.

Dataset	♯Logos	♯Images	♯Objects	Availability
BelgaLogos [21]	37	10,000	-	Yes
FlickrLogos-27 [22]	27	1,080	4,671	Yes
FlickrLogos-32 [46]	32	8,240	5,644	Yes
Top-Logo-10 [54]	10	700	-	Yes
WebLogo-2M [53]	194	1,867,177	-	Yes
QMUL-OpenLogo [55]	352	27,083	-	Yes
Logos-in-the-Wild [58]	871	11,054	32,850	Yes
Logo-2K+ [60]	2,341	167,140	-	Yes
LogoDet-3K [59]	3,000	158,652	194,261	Yes
MICC-Logos [48]	13	720	-	No
FlickrBelgaLogos [26]	34	10,000	2,695	No
Logo-18 [17]	18	8,460	16,043	No
Logo-160 [17]	160	73,414	130,608	No
Logos-32plus [1]	32	7,830	12,302	No
Video SportsLogo [28]	20	2,000	-	No
CarLogo-51 [63]	51	11,903	-	No
Open Brands [20]	1,216	1,437,812	3,113,828	No
SynthLogo [35]	604	280,000	-	No
PL2K [11]	2,000	295,814	-	No
FoodLogoDet-1500	**1,500**	**99,768**	**145,400**	**Yes**

Tabla. 1. Conjuntos de datos existentes para LD

Cabe destacar que, de los 20 conjuntos de datos etiquetados existentes, tan solo 8, es decir, un 40% del total, contienen más de 200 logotipos diferentes a predecir. Además, de estos 8 conjuntos que tienen más de 200 clases, tan solo 4 contienen la ubicación de los objetos en cada imagen. Lo que supone una limitación a la hora de investigar en conjuntos de datos con gran variedad de clases con objeto sin ubicar. Algunos estudios comentan posibilidad y la necesidad que existe de probar y experimentar con nuevos algoritmos y conjuntos de datos [9].

Además, entre los conjuntos de datos encontrados, no existe ninguno que contenga diversas imágenes sobre criptomonedas. La colección de imágenes más amplia encontrada, es la que se encuentra en la página web *Crypto Logos*. En esta página se encuentran cientos de imágenes de criptomonedas sin tener en cuenta su capitalización, utilidad… Además, cada criptodivisa es representada mediante un único logo, el perteneciente a *CoinMarketCap* [2][3].

2. Métricas destacadas para la evaluación de los modelos

Existen diversas métricas para la evaluación de los modelos en *Inteligencia Artificial*. Se van a explicar algunas de las métricas más utilizadas en el ámbito de los logotipos. Las métricas más comunes varían según el tipo de modelo a evaluar. Se pueden encontrar modelos que clasifiquen imágenes o, también, modelos que detectan los logotipos en la imagen y en qué parte de la misma se encuentran. A continuación, se van a mostrar algunos ejemplos según el tipo de modelo.

2.1. En la clasificación de imágenes

En la clasificación de logotipos se predice el logotipo que aparece en el total de la imagen. Entre las métricas más relevantes en la clasificación de imágenes se encuentran la exactitud (*Accuracy*) y la pérdida (*Loss*) [11].

2.1.1. Exactitud (Accuracy)

La **exactitud**, denominada *Accuracy*, mide el porcentaje de imágenes que el modelo ha acertado. Cuanto mayor sea el coeficiente, entre 0 y 1, mejor funcionará el modelo.

1.1.1. Función de pérdida (Loss)

La **función de pérdida**, denominada *Loss*, evalúa la desviación entre los valores reales de las imágenes y las predicciones realizadas por la red neuronal. Cuanto menor sea el valor de esta función, mayor eficiencia tendrá el modelo asociado a la red neuronal.

2.2. En la detección de objetos en imágenes

En el caso de que se quieran detectar los logotipos que aparecen en una imagen, entre las métricas más utilizadas se encuentran *Average Precission* y *Mean Average Precision*, las cuales se explicarán a continuación.

2.2.1. Average Precision (AP)

AP es el promedio de las predicciones de las clases sobre varios umbrales, es decir, es el promedio de los valores de precisión para diferentes niveles de **recall** El valor resultante comprende entre 0 y 1, siendo 1 el mejor valor de precisión. Es una métrica popular en el ámbito del reconocimiento de objetos [11].

2.2.2. Mean Average Precision (Map)

Comúnmente se le denomina el **precision-recall** de los objetos. Este nombre se debe, principalmente, a que es una de las métricas más populares para medir la precisión de los detectores de objetos. Mide como la red comprende lo importante y elimina la información no deseada. La precisión es más precisa cuando *mAP* es más alta [1]. Su fórmula es la siguiente:

$$\boldsymbol{mAP} = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} (AP_i)$$

En la fórmula de *mAP*, *AP* hace referencia a *Average Precision*, la métrica mostrada anteriormente. Por lo tanto, *mAP* es la media de todos los valores obtenidos de *AP* respecto al total de clases [11].

3. Soluciones actuales de la IA para el reconocimiento de logotipos

En este apartado se explicarán algunos de los algoritmos más destacados y utilizados. En primer lugar, se explicarán algunas de las métricas más relevantes para evaluar el rendimiento de los modelos que se crean a partir de los mejores algoritmos. Es por ello que se tendrán en cuenta los algoritmos que mejores resultados obtengan según las investigaciones más recientes.

Para observar las pruebas realizadas con los diferentes conjuntos de datos, en los últimos años, se dispone de una tabla con algunos de los estudios más relevantes y los modelos de resolución óptimos [11]. En la **Tabla 2** se puede observar cómo, gran parte de los estudios analizados, utilizan el conjunto de datos *FlickrLogo-32*. Las métricas utilizadas para evaluarlos son *Accuracy* y la media de la precisión entre los estudios realizados, *Mean Average Performance (mAP)*. De los 14 estudios mostrados, VGG16 se posiciona como una de las soluciones con mejores resultados en *datasets* con una cantidad elevada de clases, tal y como se puede observar en la **Tabla 2**. Con 160 clases obtiene un *Accuracy* del 85,8% con un tiempo de entrenamiento de 1362 minutos, un tiempo aceptable teniendo en cuenta los demás resultados. Algunos entrenamientos de *R-CNN* ascienden a más de 8000 minutos. VGG16 está clasificado como una solución *Fast R-CNN* [5].

Reference	Year	Approach	Dataset	Performance
[7]	2019	YOLOv3	VLD-3	76.6-98.8
[8]	2017	Faster R-CNN	FlickerLogo	0.52-0.67
[1]	2019	OSLD - SDML	BLAC and Flickr-32	84.5-90.9
[9]	2018	RetinaNet	INbreast - GURO	86-1.00
[10]	2017	Faster R-CNN	FlickrLogos	mAP 51-66%
[11]	2020	Faster R-CNN and YOLOv2	WebLogo-2M	mAP 36.8-46.9
[12]	2017	DenseNet16 -, ResNet101 -, VGG16	FlickrLogos-32 SportsLogos	0.37-0.46
[13]	2017	Scalable Logo Self Training (SLST)	WebLogo-2M	mAP 34.37%
[14]	2007	ANN and Support Vector Machine (SVM), Fisher classifier	Tobacco-800	39.3-84.2%
[5]	2015	RCNN – FRCN – SPPnet	Logos-18 test set Logos-160 test set	81.3-95.2%
[15]	2019	Faster R-CNN, SSD, YOLOv3	FlickrLogos-32, PL2K	0.565 mAP
[6]	2017	CNN	FlickrLogos-32 Logos-32plus	0.1-95.8%
[2]	2016	Fast Region-based Convolutional Networks (FRCN)	ILSVRC, FlickrLogos-32	mAP 54.5-73.74%
[16]	2018	Retina U-Net, Mask R-CNN, Faster R-CNN +, U-Faster R-CNN, + DetU-Net	LIDC-IDRI	mAP 29-50.5%
[17]	2017	Faster R-CNN	FlickrLogo-32 TopLogo-10	mAP 20.5-81.1%
[18]	2020	DenseNet-CNN	FlickrLogo-32	92.8%

Tabla. 2. Conjuntos de datos existentes para LD

A partir de ahora, se van a analizar algunos de los algoritmos más destacados del estado del arte para la detección de logos, añadiendo algunos que puedan resultar relevantes para esta misión.

3.1. VGG16

Pertenece a la familia de algoritmos utilizada en la competición *ILSVRC-2014* por el equipo de VGG. Se encuentra entre las arquitecturas CNN más simples utilizadas en las competencias de *ImageNet*. Debido a su profundidad son lentas de entrenar obteniendo modelos de gran tamaño. En su arquitectura se encuentran capas *convolucionales* activadas por la función *Relu*, capas *max pooling*, capas neuronales ocultas totalmente conectadas y, finalmente la última capa *Softmax* con las clases a predecir. La arquitectura agrega hasta 16 capas para aprender los pesos, y de ahí viene su nombre [7].

3.2. VGG19

La arquitectura de VGG-19 es muy similar a la de la VGG-16. La diferencia principal de VGG-19 radica en el número de capas, agregando hasta 19 capas para aprender los pesos [7].

3.3. Googlenet

GoogleNet es un algoritmo creado por Google, entrenado con *ImageNet*, que consiguió entrenar redes neuronales más pequeñas con mejores resultados, en comparación con las CNN. La red de *GoogleNet* utilizó para la clasificación una nueva variante llamada *Inception*, utilizando R-NN para la detección [4].

3.4. InceptionV3

InceptionV3 se presentó en la misma investigación que la otra versión de su misma familia *InceptionV2*. Se investigaron las posibilidades de mejorar la versión anterior sin retocar mucho los módulos. Este es uno de los motivos por el cual se ha decido centrarse en *InceptionV3* para este trabajo.

La arquitectura de la versión *InceptionV3* tiene como base la red convolucional de sus predecesores, añadiendo varias mejoras como el uso de *Label Smoothing*. Esta técnica intenta mejorar el sobreajuste mediante el uso de la función de pérdida. Además de disponer de diversas capas convolucionales, en vez de darle profundidad a los filtros, se les dio anchura, de esta manera se conseguía mejorar la red sin obtener una pérdida de información a cambio [8].

3.5. InceptionResnetV2

InceptionResnetV2 es un algoritmo de la familia *Inception*, con una arquitectura más profunda que *InceptcionV3* y, según algunos estudios recientes, con mejor un mejor rendimiento en la clasificación.

InceptionResnetV2 dispone de una arquitectura muy profunda. Su arquitectura es mucho más precisa que la de modelos anteriores, aun así, el nivel requerido de memoria no es tan elevado como el esperado.

3.6. Densenet

Denset se creó gracias a la premisa de que mediante una conexión densa se lograsen menos parámetros y con una alta precisión. Este algoritmo utiliza entradas adicionales de todas las capas anteriores, pasándoles sus propios mapas de características. Cada capa recibe el conocimiento colectivo de los procesos anteriores, consiguiendo un número de canales menor y una mayor eficiente computacional [13].

3.7. Yolo

You Only Look Once (YOLO) utiliza *Deep Learning* y las CNN para el reconocimiento de objetos en tiempo real. Algunos estudios recientes obtienen mejores resultados que las *Faster R-CNN*, en el ámbito del reconocimiento de objetos. Estos estudios comienzan a realizar pruebas en conjuntos de datos más amplios y con un número de clases mayor, animando a los investigadores a usar este tipo de *datasets* [15].

Existen diferentes versiones de YOLO, entre las más actuales y destacadas están *YOLOV3*, *YOLOV4* y *YOLOV5*. Aunque sus aplicaciones están más centradas a la detección de objetos en tiempo real junto con su ubicación en la imagen, es por ello que no se profundizará en este algoritmo.

4. *Problemas a resolver de la IA en el reconocimiento de logotipos*

Existen diferentes problemáticas a resolver tales como la escasez en los conjuntos de datos y la dificultad del etiquetado de los mismos, entre otros.

4.1. Escasez de conjuntos de datos

Esta problemática hace que los datos investigados hasta el día de hoy necesiten revisiones y se requieran nuevos estudios para afianzarlos. Existe una necesidad de generar nuevos *datasets*.

4.2. Dificultad en el etiquetado de los conjuntos de datos

En la problemática de reconocimiento de objetos, actualmente, es imprescindible que los conjuntos de datos estén correctamente etiquetados. Cada imagen debe ir con su etiqueta. El problema radica cuando los conjuntos de datos son a nivel de objeto, etiquetar cada objeto es una labor muy laboriosa y que, normalmente, se realiza manualmente.

Los algoritmos de la familia YOLO trabajan a nivel de objeto. Esto supone el tener que disponer, previamente, de un *dataset* etiquetado a este nivel. También existe la posibilidad de etiquetarlos mediante herramientas como *RobotFlow*, *LabelImage…* [14].

5. *Data Augmentation para el aumento de conjunto de datos*

Data Augmentation es un conjunto de técnicas que permite obtener conjuntos de imágenes más grandes y variados. Para conseguirlo trata de variar, voltear, girar… un conjunto de imágenes, con el fin de obtener más imágenes semejantes al conjunto de datos inicial. Existes estudios que demuestran la mejora que ejercen este tipo de técnicas en las *Redes Neuronales Convolucionales.* En estos estudios se analiza el comportamiento de las CNN utilizando *datasets* con y sin *Data Augmentation*. Concluyen con un mejor rendimiento de las CNN mediante la generalización y la variedad del conjunto de datos que ofrecen este tipo de técnicas [10].

6. Transfer Learning para el elevado tiempo de entrenamiento

Transfer Learning (TL) se refiere al conjunto de métodos que consisten en poder transmitir conocimiento, mediante diferentes métodos, para resolver problemas de una tipología similar [1].

En el ámbito del *Machine Learning*, la principal diferencia que existen entre enfoque tradicional y el *TL*, se basa en la reutilización de modelos previamente entrenados por parte del *TL*. En el ámbito de la detección de logotipos, si se cuenta con grandes cantidades de datos y con un número de clases muy variadas, el entrenamiento podría ser inasumible para un equipo doméstico. Y aquí es donde gana en importancia el *Transfer Learning*, puede ayudar a entrenar modelos aprovechando al máximo los recursos disponibles.

7. Despliegue de los modelos

Empresas como IBM, *Google*, *Amazon* y *Microsoft* comienzan a disponer de herramientas para poder implementar soluciones de *Inteligencia Artificial* en la nube. Algunas de estas herramientas pueden alojar los modelos entrenados en sus servidores [12].

III. Objetivos y Metodología

El objetivo general de esta investigación es la de crear una aplicación móvil para evitar los fraudes mediante criptomonedas, y, además, poner a prueba las diferentes arquitecturas de *Transfer Learning*. Se necesita obtener un modelo óptimo que clasifique las criptomonedas más demandadas. De esta manera se pondrán a prueba algunas de las arquitecturas con mejor rendimiento. Se dispone de un conjunto de datos con la siguiente distribución, tal y como se muestra en la **Tabla 3**.

Dataset 1	Entrenamiento	Validación
NºClases (Criptomonedas)	250	250
N.º Imágenes/Clase	26-30	10
Tamaño (GB)	1,4	0,08
Nº Total Imágenes	7115	2500

Tabla. 3. Conjunto de datos inicial

1. *Aumentando el dataset mediante Data Augmentation*

Algunos estudios afirman que un pequeño aumento de los datos de entrenamiento no tiene mucha influencia en el rendimiento del clasificador. Y, además, que un gran aumento de los datos tiene influencia en la buena generalización del modelo [10]. Por lo tanto, se ha decidido generar 3 *datasets* realizando diferentes aumentos. De esta manera, aparte de probar la exactitud de algunos algoritmos del estado del arte, se van a poder comparar los resultados con distintitos niveles de *Data Augmentation*. El conjunto de validación siempre será el mismo.

Para realizarlo de manera controlada, se ha diseñado un algoritmo de *Data Augmentation* que genera 3 tipos de aumento con las imágenes del *Dataset 1*, el que se muestra en la **Tabla 3**. Cada conjunto de datos creado contiene al anterior, pero no al revés. De esta manera, cada conjunto de datos contiene un número de imágenes mayor al anterior.

En todas y cada una de las iteraciones del aumento de imágenes de entrenamiento, las modificaciones que han tenido los nuevos logotipos generados han sido la rotación, el *zoom*, la inversión y el añadido de ruido mediante una función aleatoria. El grado de este tipo de modificaciones ha ido disminuyendo en cada una de las 3 iteraciones. La disminución del ruido es clave para obtener *datasets* mayores, pero sin obtener un ruido excesivo a cambio. A continuación, se detallarán los *datasets* obtenidos, tal y como se muestra en la **Tabla 4**.

	Dataset 2	Dataset 3	Dataset 4
NºClases (Criptomonedas)	250	250	250
N.º Imágenes/Clase	280-290	1100-1200	4600-4700
Tamaño (GB)	10,2	29,6	118,5
Nº Total Imágenes	71.148	298.433	1.170.014

Tabla. 4. Conjuntos de datos de entrenamiento resultantes tras realizar Data Augmentation

Para probar los resultados con diferente número de clases, se ha decido generar otros 3 conjuntos de datos con 100 clases. Estos *datasets* disponen de las mismas imágenes mostradas en la **Tabla 4**, únicamente se ha reducido el número de clases a predecir. Los conjuntos de datos se muestran en la **Tabla 5**.

	Dataset 5	Dataset 6	Dataset 7
NºClases (Criptomonedas)	100	100	100
N.º Imágenes/Clase	280-290	1100-1200	4600-4700
Tamaño (GB)	3,8	12,6	59,8
Nº Total Imágenes	71.148	120.438	470.000

Tabla. 5. Conjuntos de datos de entrenamiento resultantes de 100 clases

Por lo tanto, se van a realizar diferentes pruebas con los diferentes conjuntos de datos y los algoritmos VGG16, VGG19, *InceptionV3* e *InceptionResnetV2*. Se realizarán 3 entrenamientos por cada arquitectura y conjunto de datos, aunque dada la limitación computacional, se irán reduciendo las pruebas según los resultados.

Se comenzará con los conjuntos de datos con mayor variedad de criptomonedas, es decir, con 250 clases y, además, de menor a mayor número de imágenes por clase. Tras realizar dichos entrenamientos y comparar los resultados, se irá disminuyendo el número de clases. De esta manera, se podrán obtener datos sobre el rendimiento de estas arquitecturas con diferente número de clases y, además, con diferente número de imágenes por clase.

Para el entrenamiento, se ha elegido *Google Colab* mediante el uso de GPU. El servidor utilizado dispone de una GPU *NVIDIA-SMI 460.32.03* con 12GB de RAM *y* 166GB de almacenamiento total. El servicio gratuito asigna alrededor de 62GB utilizables y, en la versión Pro, alrededor de las 124GB.

IV. Contribución

Está investigación contribuye a comprobar el rendimiento que tienen ciertas arquitecturas de *Transfer Learning* como VGG16, VGG19, *InceptionV3* e *InceptionResnetV2*, en la clasificación de logotipos con una elevada cantidad de clases.

Además, si los resultados son los esperados, mediante la aplicación móvil desarrollada, se ayudará a que los inversores del mundo cripto puedan contar con mayor seguridad en futuras transacciones.

V. Evaluación y Resultados

Se van a evaluar los mejores modelos obtenidos para el buen funcionamiento del mismo.

1. *Evaluación 1 (250 clases)*

Se ha entrenado el modelo utilizando el *Dataset 2*. El mejor rendimiento lo obtiene el algoritmo VGG16, con un *Accuracy* del 57,2% en aproximadamente 239 minutos, tal y como se muestra en la **Tabla 6**.

D2 (250 clases)	Training Time (Min)	Max (Accuracy)
VGG16	239	57,2%
VGG19	245	56,6%
InceptionResnetV2	243	54,6%
InceptionV3	145	51,07%

Tabla. 6. Resultados utilizando el Dataset 2 con 250 criptomonedas

Los resultados obtenidos mediante el *Dataset 2*, están lejos del *Accuracy* mínimo esperado del 75%. Por lo tanto, se va a entrenar el modelo utilizando el *Dataset 3*, el cual cuenta con una diversidad mayor en cada clase.

Tras realizar varias pruebas con el algoritmo que mejores resultados obtuvo con el *Dataset 2*, el VGG16, se obtiene una mejora hasta llegar a un 65,9% de *Accuracy*. Aun así, sigue lejos del rendimiento esperado, tal y como se muestra en la **Tabla 7**.

D3 (250 clases)	Training Time (Min)	Max (Accuracy)
VGG16	1.110	65,9%

Tabla. 7. Resultados utilizando el Dataset 3 clasificando 250 criptomonedas

2. *Evaluación 2 (100 clases)*

Los resultados de los entrenamientos, con el *Dataset 6* al reducir el número de clases a 100, siguen sin acercarse al mínimo requerido para ser aceptados en este proyecto. Los resultados máximos obtenidos mediante el algoritmo VGG16 se encuentra en la **Tabla 8**.

D6 (100 clases)	Training Time (Min)	Max (Accuracy)
VGG16	453	63,8%

Tabla. 8. Resultados utilizando el Dataset 6 clasificando 100 criptomonedas

A continuación, se comprobará el rendimiento del algoritmo VGG16 con el *dataset* que mayor variedad de logotipos tiene por clase, el *Dataset 7*. El mejor resultado obtenido llega al 74% de *Accuracy*, tal y como se muestra en la **Tabla 9**.

D7 (100 clases)	Training Time (Min)	Max (Accuracy)
VGG16	349	74%

Tabla. 9. Resultados utilizando el Dataset 7 clasificando 100 criptomonedas

Tras numerosas pruebas con este algoritmo y el *Dataset 7*, siendo el que mejores resultados está ofreciendo, se sigue sin llegar al

objetivo esperado. Aun así, se acerca al *Accuracy* mínimo requerido para la aplicación móvil desarrollada.

3. *Evaluación 3 (90 clases)*

Tras comprobar que los mejores resultados en el entrenamiento se obtienen con los conjuntos de datos más variados, se ha decidido realizar una última prueba con 90 clases obtenidas del *Dataset 7*.

El último conjunto de datos creado dispone de 90 clases y una gran variedad de imágenes por clase, tal y como se muestra en la **Tabla 10**.

	Dataset 8
NºClases (Criptomonedas)	90
N.º Imágenes/Clase	4600-4700
Tamaño (GB)	53,5
Nº Total Imágenes	423.000

Tabla. 10. Conjuntos de datos con 90 clases

Los mejores resultados obtenidos con el algoritmo VGG16, entrenando el modelo con el *Dataset 8* y clasificando 90 criptomonedas diferentes, se muestran en la **Tabla 11**.

D8 (90 clases)	**Training Time (Min)**	**Max (Accuracy)**
VGG16	282	77%

Tabla. 11. Resultados utilizando el Dataset 8 clasificando 90 criptomonedas

VI. Discusión

En este apartado, se van a tratar las principales limitaciones actuales que tiene la clasificación de logos en la IA.

En primer lugar, existe una gran limitación a la hora de encontrar conjuntos de datos. Además, el etiquetado de los mismos es una tarea tediosa. La limitación en el número de clases es otro de los problemas a la hora de la detección de imágenes, ya que la mayor parte de los *datasets* utilizados, en el estado del arte, disponen de menos de 200 clases para clasificar. Esto deja al descubierto la problemática que existe en entornos donde las clases a predecir superen estas 200 clases. Sería recomendable poner a prueba el rendimiento de los algoritmos con *datasets* más variados.

Además, los altos tiempos de ejecución de los algoritmos previamente mencionados, con una GPU pueden llegar a durar horas, incluso días. Por lo tanto, cuantas más clases se dispongan a clasificar, mayor número de imágenes se necesitan, incurriendo en un costo computacional cada vez mayor.

VII. Conclusiones

La fiebre que rodea las criptomonedas está llegando a puntos enfermizos. Miles de transacciones comienzan por *Twitter* o por grupos de *Telegram*, dando credibilidad a cualquier tipo de información. Todo ello da lugar a que la gente se lance compulsivamente a comprar criptomonedas, basándose únicamente en su logotipo. Esto último trae como consecuencia el fraude mediante la suplantación de identidad.

Es complicado detectar si una criptomoneda puede ser fraudulenta o no. Aun así, la aplicación móvil resultante de este proyecto permite, a partir del logo de las criptomonedas capturadas mediante *Twitter* o grupos *Telegram*, obtener información relevante de las mismas. Por lo tanto, en este aspecto, cumple los objetivos determinados. Gracias a esta nueva forma de verificación mediante el logotipo, se le ofrece una seguridad adicional a este tipo de inversores. Aun así, el rendimiento de los algoritmos utilizados, mediante el uso del *Transfer Learning* debería ser mejor.

En el ámbito de la *Inteligencia Artificial*, la primera conclusión que obtengo, es que se necesitan grandes conjuntos de datos para realizar *Transfer Learning* en clasificadores con más de 100 clases. En este sentido, tras probar con más de 4600 imágenes por cada clase, se han obtenido mejores resultados que con los conjuntos de datos menos variados. Aun así, los resultados no mejoran en exceso. El gasto computacional, en cambio, se dispara.

Aumentar el *dataset* era otro de los objetivos, en el trabajo se ha mostrado como se han generado diferentes *datasets* mediante *Data Augmentation*, con el objetivo de experimentar de una manera más exhaustiva. Hubo algunas dificultades en la limpieza de los conjuntos dado el gran volumen de imágenes que había que tratar, pero, aun así, el objetivo se ha alcanzado correctamente.

La segunda conclusión que obtengo es que se necesita una gran cantidad de datos para entrenar este tipo de modelos, es decir, cuando las clases a predecir son mayores a 100. Aun así, los resultados no llegan al 75% de *Accuracy*. Tras probar con distintos conjuntos de datos y distintos niveles de *Data Augmentation*, he de decir que los resultados no han sido tan buenos como se esperaba. Por lo tanto, la variedad de clases que ofrece el ámbito de las criptomonedas, sigue siendo el medio ideal para seguir investigando y mejorando las técnicas actuales de clasificación de logotipos mediante *Transfer Learning*.

Referencias

[1] Attal, M. (2022). *¿Qué es el transfer learning?* https://datascientest.com/es/que-es-el-transfer-learning

[2] CoinMarketCap. (2021). *Cryptocurrency prices, charts and market capitalizations.* https://coinmarketcap.com/

[3] Crypto Logos. (2022). *Crypto logos - cryptocurrency logo files (.SVG & .PNG) download.* https://cryptologos.cc

[4] Garcia-Dominguez, M., Dominguez, C., Heras, J., Mata, E., y Pascual, V. (2020). FrImCla: A framework for image classification using traditional and transfer learning techniques. *IEEE Access, 8*, 1. 10.1109/ACCESS.2020.2980798

[5] Hoi, S. C. H., Wu, X., Liu, H., Wu, Y., Wang, H., Xue, H., y Wu, Q. (2015). LOGO-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks. https://arxiv.org/abs/1511.02462

[6] Karimi, M., & Behrad, A. (Mar 2019). (Mar 2019). Logo recognition by combining deep convolutional models in a parallel structure. Paper presented at the 216-221. 10.1109/PRIA.2019.8786032 https://ieeexplore.ieee.org/document/8786032

[7] kumar, K. (2021). *Transfer learning with VGG16 and VGG19, the simpler way!* https://koushik1102.medium.com/transfer-learning-with-vgg16-and-vgg19-the-simpler-way-ad4eec1e2997

[8] Raj, B. (2020). *A simple guide to the versions of the inception network.* https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202

[9] Romero, C. (2021). Ana patricia botín, la séptima mujer más poderosa del mundo. http://forbes.es/forbes-w/129003/ana-patricia-botin-la-septima-mujer-mas-poderosa-del-mundo/

[10] Romero, N., Gutoski, M., Hattori, L., y Lopes, H. S. (2017). *The effect of data augmentation on the performance of convolutional neural networks*. ABRICOM. 10.21528/cbic2017-51

[11] Sahel, S., Alsahafi, M., Alghamdi, M., y Alsubait, T. (2021). Logo detection using deep learning with pretrained CNN models. *Engineering, Technology & Applied Science Research, 11*(1), 6724-6729. 10.48084/etasr.3919

[12] Shaikh, R. (2022). *Deploying A deep learning model on mobile using TensorFlow and react.* https://towardsdatascience.com/deploying-a-deep-learning-model-on-mobile-using-tensorflow-and-react-4b594fe04ab

[13] Tsang, S. (2019). *Review: DenseNet — dense convolutional network (image classification).* https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803

[14] ultralytics. (2022). *Train custom data.* https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data

[15] Wang, J., Min, W., Hou, S., Ma, S., Zheng, Y., y Jiang, S. (2020). LogoDet-3K: A large-scale image dataset for logo detection. https://arxiv.org/abs/2008.05359