Universidad Internacional de La Rioja (UNIR)

Escuela de Ingeniería

Máster en Análisis y Visualización de Datos Masivos

Análisis y predicción de la evolución del precio de las criptomonedas

Trabajo Fin de Máster

presentado por: Cristian García Díaz

Director: Luis Pedraza Gómara

Ciudad: Barcelona
Fecha: 18/09/2018

Agradecimientos

A mi familia, mis amigos y Eva por su paciencia y tolerancia, a Luis Pedraza que gracias a sus sugerencias y comentarios me han ayudado a encauzar este proyecto, a mis mentores Albert Prades y Elisa Palacios que gracias a su contagiosa motivación y sus indicaciones me mostraron el camino que me conduciría hasta el momento presente, en definitiva, a todas esas y esos gigantes a los que a sus hombros vamos, en especial a ti Alejandro que estoy seguro que eres feliz donde quiera que estés.

Resumen

El objetivo de la investigación es estudiar la estimación del precio de las criptomonedas empleando modelos predictivos utilizando Python sobre la plataforma de ciencia de datos más popular, Anaconda. Se lleva a cabo un análisis previo mediante la identificación, la selección, la visualización y el análisis estadístico de los datos. Se realiza una correlación entre las diferentes criptomonedas obteniendo como resultado que los precios de las criptomonedas se relacionan cada vez más unas con otras a medida que transcurre el tiempo. Posteriormente, se seleccionan los indicadores relevantes de Bitcoin para la implementación de los modelos predictivos. Los métodos empleados son la regresión múltiple y aprendizaje automático. Se comparan los resultados obtenidos y se encuentra que el modelo de la regresión múltiple Ordinary Least Squares (OLS) obtiene un mejor resultado que los modelos predictivos implementados con técnicas de aprendizaje automático de redes neuronales.

Palabras Clave: criptomonedas, predicción de series temporales, regresión múltiple, aprendizaje automático, red neuronal.

Abstract

The objective of the research is to study the price estimation of cryptocurrencies using predictive models using Python on the most popular data science platform, Anaconda. A preliminary analysis is carried out through the identification, selection, visualization and statistical analysis of the data. A correlation is made between the different cryptocurrencies obtaining as a result that the prices of cryptocurrencies are increasingly related to each other as time passes. Subsequently, the relevant Bitcoin indicators are selected for the implementation of predictive models. The methods used are multiple regression and automatic learning. The obtained results are compared and it is found that the multiple regression model Ordinary Least Squares (OLS) obtains a better result than the predictive models implemented with automatic neural network learning techniques.

Keywords: cryptocurrencies, time series prediction, multiple regression, machine learning, neural network.

Índice de contenido

Índice de figuras

Índice de Códigos

Índice de tablas

1 Introducción

Las criptomonedas son relativamente nuevas, su tecnología ampliamente desconocida y su valor altamente volátil. La problemática consiste en la estimación del precio de las criptomonedas con el objetivo de obtener una mayor rentabilidad. Para ello, se profundiza en entender qué son las criptomonedas, la tecnología en la que se sustentan, para que sirven, cómo funcionan, como se intercambian, las ventajas y desventajas.

Para dar solución al problema se utiliza una combinación de la tecnología de *Python* sobre la plataforma de ciencia de datos más popular, *Anaconda*. Se identifican y capturan los datos de diferentes fuentes para realiza un análisis previo y seleccionar los indicadores más relevantes para la implementación de los modelos predictivos. En el primer modelo se emplea la técnica de regresión múltiple *Ordinary Least Squares (OLS).* En el segundo modelo se utiliza la técnica de aprendizaje automático, concretamente la red neuronal. Para el segundo modelo se realizan diferentes implementaciones aplicando tres redes neuronales distintas. La primera red neuronal es *Long Short Term Memory* (*LSTM*) de una capa, la segunda *LSTM2* con dos capas *LSTM* conectadas de forma secuencial y la tercera *Gate Recurrent Units (GRUs)*. Finalmente, se evalúan y comparan los resultados obtenidos para la extracción de las conclusiones y establecer futuras líneas de trabajo.

1.1 Motivación

Surge la atracción por descubrir cómo funciona la tecnología de las criptomonedas, analizar la evolución del precio y estimar su precio. Se trata de un problema relevante debido al desconocimiento existente de su tecnología, a las posibles futuras aplicaciones y al empleo de técnicas de la inteligencia artificial como son la regresión lineal y las redes neuronales.

1.2 Planteamiento del trabajo

La solución adoptada para solventar el problema de la predicción del precio de las criptomonedas se trata, en primer lugar, de conocer y entender el funcionamiento de las criptomonedas. En segundo lugar, de obtener datos útiles, seleccionar, analizar y visualizar los indicadores que probablemente influyen en el precio. En tercer lugar, implementar dos

modelos predictivos. El primero aplicando la técnica de regresión múltiple y el segundo aplicando redes neuronales. Posteriormente, comparar y evaluar los resultados obtenidos.

1.3 Estructura del trabajo

En el capítulo del **Contexto** se explican las criptomonedas y su tecnología desde una perspectiva técnica para entender su funcionamiento. Se analiza las bases criptográficas en las que sustentan, la descentralización del sistema y los intercambios de criptomonedas, las ventajas e inconvenientes y por último el planteamiento de la problemática.

En el apartado del ¡Error! No se encuentra el origen de la referencia. se analizan diversos artículos de distintos autores con el objetivo de identificar qué factores influyen en el precio de las criptomonedas y qué metodologías, técnicas y algoritmos emplean.

En el capítulo de **Objetivos y metodología de trabajo** se definen los objetivos, tanto el general como los específicos y se describe la metodología a seguir para dar solución al problema.

El apartado de **Desarrollo específico de la contribución** se centra en alcanzar los objetivos que se han definido en el capítulo anterior. Se explican qué tecnologías se utilizan, la obtención de los datos, cómo obtenerlos, tratarlos y analizarlos. Después, se explica la implementación de los modelos predictivos y se evalúan los resultados obtenidos.

Por último, en el capítulo **Conclusiones y trabajo futuro** se hace un breve resumen del problema y solución obteniendo las conclusiones del trabajo. Por último, se describen futuras líneas de trabajo.

2 Contexto

Satashi Nakamoto, el seudónimo del creador de *Bitcoin*, publicó en 2008 un documento [1] definiendo un protocolo de sistema de criptomonedas llamado *Bitcoin*. Se trata de un proyecto de código abierto con una comunidad que lo mantiene. Después, han ido apareciendo progresivamente otras criptomonedas basadas en el mismo protocolo que *Bitcoin,* pero con diferentes características.

Primero se comienza definiendo que es el término criptomoneda. Según el diccionario de Oxford se define como “una moneda digital en que se utilizan técnicas de cifrado para regular la generación de unidades de moneda y verificar transferencias de fondo, operando independientemente de un banco central”. El diccionario de Cambridge define el término como “una moneda digital producida por una red pública, en lugar de cualquier gobierno, que utiliza la criptografía para asegurarse de que los pagos se envían y se reciben de forma segura”. El mismo diccionario define moneda digital como “una forma de efectivo digital comprado de una compañía particular para pagar bienes y servicios en Internet”. Por lo tanto, una criptomoneda se considera como un caso particular de una moneda digital.

Estas definiciones ayudan a dar una idea general, sin embargo, para comprender realmente qué tiene de especial *Bitcoin* y la demás criptomonedas se debe profundizar en los detalles y en el funcionamiento a nivel técnico. Para ello, se va a centrar la explicación en la criptomoneda pionera en aplicar diversas tecnologías de una forma específica. Se va a utilizar el término de *Bitcoin* para hacer referencia el protocolo y a la red mientras que se utilizará *bitcoin* para referirse a la criptomoneda. La información referente al capítulo de contexto, es decir al funcionamiento de *Bitcoin* y de las tecnologías subyacentes, como por ejemplo *Blockchain,* ha sido obtenida de [2].

En los siguientes apartados se va explicar la base criptográfica fundamental en la que sustenta *Blockchain* y las demás tecnologías, la descentralización y el sistema, los intercambios de criptomonedas, las ventajas e inconvenientes y por último el planteamiento de la problemática existente en el momento de invertir en criptomonedas.

2.1 Base criptográfica

Todos los tipos de monedas necesitan controlar el suministro de nuevas monedas y aplicar varias medidas de seguridad para evitar falsificaciones o movimientos fraudulentos. En las

monedas fiduciarias, las organizaciones como los bancos, controlan la incorporación de nuevas unidades monetarias y se encargan de añadir medidas de seguridad para evitar la falsificación del dinero físico. La aplicación de estas medidas eleva la seguridad del dinero, sin embargo, no hacen que sea imposible de falsificar.

Las criptomonedas también tienen medidas de seguridad específicas para evitar que las personas puedan alterar el estado del sistema. En [2] utilizan el siguiente ejemplo: Si Alice convence a Bob de que le pagó una moneda digital, no debería ser capaz de convencer a Carol de que le pagó la misma moneda. Este problema es conocido como el problema de doble gasto y el sistema *Bitcoin* utiliza el *Blockchain* para solventarlo.

La diferencia con las monedas fiduciarias es que las reglas de seguridad de las criptomonedas deben aplicarse tecnológicamente y sin depender de ninguna autoridad central. Como la palabra indica, las criptomonedas utilizan la criptografía como mecanismo para codificar de forma segura las reglas en el propio sistema. Para entender las reglas y la tecnología de las criptomonedas, se debe profundizar en las bases criptográficas en las que se sustenta el sistema de criptomonedas. A continuación, se verán las funciones criptográficas hash, la tecnología *blockchain* y las firmas digitales.

2.1.1 Función criptográfica hash

Se comienza con el primer concepto que es la función criptográfica hash. Primero se verá cómo surgió. La idea era encontrar un mecanismo que evitará el envío de correos no deseados por correo electrónico. La primera solución fue la idea propuesta por los criptógrafos Dwork y Naor en 1992 [3].

La solución se basaba en que cada vez que se enviará un correo electrónico, el ordenador del emisor debía resolver un problema computacional que tardaba unos segundos. Para después, adjuntar la solución en el correo electrónico. El programa de correo electrónico del destinatario simplemente ignoraba todo el correo electrónico que no tuviera adjunta la solución al problema computacional.

Una idea similar fue descubierta posteriormente por Adam Back en 1997 en una propuesta llamada Hashcash [4]. Estos problemas computacionales debían tener algunas propiedades específicas para ser un elemento útil de disuasión de correo no deseado.

- En primer lugar, debería ser imposible para un emisor de correo no deseado resolver un problema computacional para cada correo electrónico no deseado y adjuntar la

solución a cada correo electrónico. Para garantizar esto, el problema debía ser específico para cada correo electrónico, debía depender del remitente y el receptor.

- Segundo, el receptor debería ser capaz de verificar fácilmente la solución del problema computacional sin tener que repetir el proceso de resolviendo el problema computacional.
- En tercer lugar, cada problema computacional debía ser totalmente independiente de los demás. Es decir, que si se resolvía un problema computacional no disminuía la cantidad de tiempo que tomaría resolver otro problema computacional.
- Finalmente, ya que el hardware mejora con el tiempo y resolver cualquier problema computacional se realiza cada vez más rápido, los destinatarios deberían ser capaces de ajustar la dificultad de las soluciones de problema computacional que aceptarán. *Bitcoin* se beneficia de esta idea y utiliza funciones criptográficas hash para diseñar estos problemas computacionales.

Función criptográfica hash

Una función es una función hash general cuando cumple las siguientes tres características:

- La entrada puede ser una cadena de cualquier tamaño.
- Produce una salida de tamaño fijo.
- Es eficiente computacionalmente, es decir, para una cadena de entrada, se puede averiguar cuál es la salida de la función hash en un período de tiempo razonable. Para calcular el hash de una cadena de n bits debe tener un tiempo de ejecución que sea $O(n)$.

Propiedades de la función hash

Para que una función hash sea criptográficamente segura y pueda utilizarse en el ámbito de las criptomonedas, tiene que tener las siguientes tres propiedades:

1. Resistente a colisiones.
2. Ocultación.
3. Facilidad de desconcierto.

Propiedad 1: La función debe ser resistente a colisiones. Una colisión ocurre cuando para dos entradas distintas se produce una misma salida. Una función hash $H()$ es resistente a colisión si nadie encuentra una colisión. Formalmente:

Resistente a colisiones(Collision resistance): Una función hash H es resistente a colisiones si no se cumple que: dados dos valores, x e y, donde $x \neq y$, sea $H(x) = H(y)$.

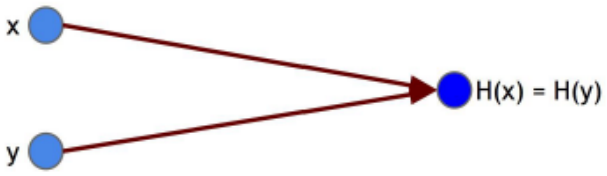

Figura 1: Una colisión hash [2, Fig 1.1].

En la ***Figura 1*** se puede ver la entrada de una función hash con dos valores distintos x e y. Se produce la misma salida. Se debe tener que la propiedad es que nadie encuentre una colisión y no que no existan colisiones. En realidad, se conoce que las colisiones existen. Es decir, el espacio de entrada de una función hash es cualquier longitud de caracteres mientras que el espacio de salida es una longitud fija de caracteres. El espacio de entrada es más grande que el de salida, por lo tanto, debe haber cadenas de entrada que se mapeen con la misma cadena de salida. En la ***Figura 2*** se muestra el número posible de entradas excede el número posible de salidas. Debe haber para una salida de la función hash más de una entrada.

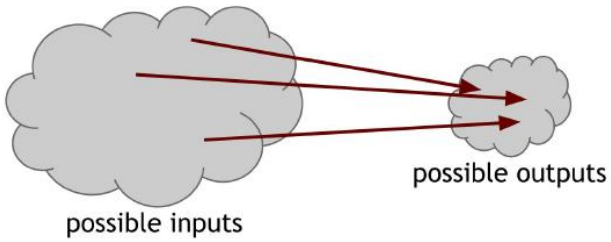

Figura 2 Posibles entradas y salidas de una función hash [2, Fig 1.2].

Existen métodos para encontrar colisiones en funciones hash. Por ejemplo, para una función hash de 256 bits de salida: escoger como entrada $2^{256} + 1$ de valores distintos, de los cuales se debe calcular el valor hash de cada uno de ellos y comprobar si existen 2 valores iguales. Como se eligen más entradas que salidas posibles, algunos pares deben colisionar cuando se aplica la función hash.

Este método garantiza encontrar una colisión. Pero si escogemos entradas aleatorias $2^{130} + 1$, hay un 99.8% de posibilidades de que al menos dos de ellos vayan a colisionar. Este algoritmo para encontrar colisiones funciona, pero el problema es que se tarda mucho tiempo en resolverlo. Para una función hash de 256 bits de salida, se tendría que calcular la función hash $2^{256} + 1$ veces en el peor de los casos, y 2^{128} veces en promedio. Si un dispositivo realiza

10.000 hashes por segundo, tardaría más de 10^{27} años en calcular 2^{128} hashes. Y esto es muchísimo tiempo.

Un hash es como si se tratará de un resumen de longitud fija de un mensaje. Más adelante se mostrarán aplicaciones donde es útil utilizar funciones hash como resumen de un mensaje.

Propiedad 2: La propiedad de ocultación afirma, que si se da un resultado de la función hash $y = H(x)$, no hay posibilidad de conocer cuál fue la entrada x. Formalmente:

Ocultación(Hiding): Una función hash H se oculta si: cuando se elige un valor secreto r de una distribución de probabilidad que tiene una entropía mínima alta, luego dado $H\ (r \parallel x)$ es imposible encontrar x.

En teoría de la información, la entropía mínima es la medida para conocer si es predecible un resultado. Una entropía mínima alta significa que la distribución (por ejemplo, de una variable aleatoria) está muy dispersa.

Por ejemplo, si el valor secreto r se elige de forma aleatoria entre todas las cadenas de 256 bits de longitud, la probabilidad de elección de cualquier cadena es $½^{256}$, es decir un valor infinitesimalmente pequeño.

Propiedad 3: *Puzzle friendliness*. Es una propiedad amigable con los problemas computacionales. Formalmente:

Puzzle friendliness: Una función hash H es *puzzle friendliness* con los problemas computacionales, si para cada posible salida de n bits del valor y, si k se elige de una distribución con la entropía mínima alta, por lo tanto no es factible encontrar x tal que $H(k\|x) = y$ en el tiempo significativamente menor que 2^n.

Lo que significa es que si alguien quiere como objetivo de una función hash obtener un valor de salida particular y, si hay parte de la entrada que se elige de forma aleatoria, es muy difícil encontrar otro valor que llegue a ese objetivo.

SHA-256

Se han mostrado las tres propiedades generales de las funciones hash y las tres propiedades adicionales de seguridad. Existen muchas funciones hash, un caso particular es la que utiliza principalmente *Bitcoin*. La función hash *SHA-256*. Esta función utiliza la trasformación *Merkle-Damgard*. La función hash resistente a colisiones se llama función de compresión. Se ha demostrado que si la función de compresión es resistente a colisiones la función hash también lo es.

En la transformación de *Merkle-Damgard* se supone que la función de compresión como entrada puede tener entradas de longitud m y produce una salida de longitud menor a n. La entrada de la función hash puede ser de cualquier tamaño, con lo que la entrada se divide en bloques de longitud $m - n$. El funcionamiento es que toma como entrada cada bloque $m - n$ junto con la salida del bloque anterior a la función de compresión. La longitud de entrada será $(m - n) + n = m$, que es la función de entrada de la función de compresión.

Para el primer bloque en el cual no hay salida de la función de compresión se utiliza un vector de inicialización (IV). Este número se reutiliza para cada llamada a la función hash. La salida del último bloque es lo que devuelve como salida la función hash. En SHA-256 utiliza la función de compresión que toma como entrada 768 bits y produce salidas de 256 bits. El tamaño de cada bloque es de 512 bits.

En la ***Figura 3*** se puede observar que SHA-256 utiliza la transformación Merkle-Damgard para activar la función de compresión resistente a colisiones con entradas de la función hash de cualquier longitud. La entrada esta forzada a que su longitud sea un múltiplo de 512 bits.

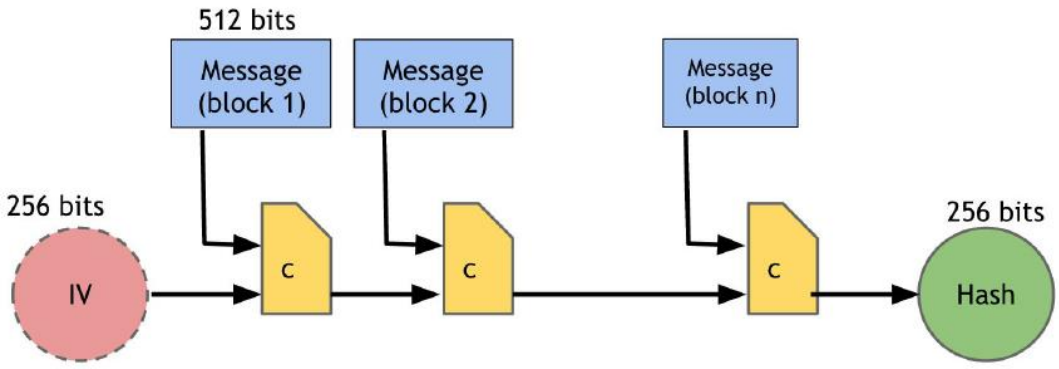

Figura 3:Función hash SHA-256(simplificada) [2, Fig 1.3].

2.1.2 Blockchain

El siguiente concepto es la idea detrás de Blockchain se remonta a un documento [5] de Haber y Stornetta en 1991. Se dio como solución al problema de doble gasto. Su propuesta era un método de sellado seguro de tipo marca de tiempo para documentos digitales. El objetivo de la marca de tiempo es conocer cuándo se creó un documento. Más importante aún, el sellado de tiempo permite conocer con precisión el orden de creación de estos documentos. El requisito de seguridad es que la marca de tiempo de un documento no se pueda cambiar una vez creado el documento.

En el esquema de Haber y Stornetta, hay un servicio de sellado de tiempo en el cual los clientes envían mensajes o documentos con marca de tiempo. Cuando el servidor recibe un documento, firma el documento junto con la marca de tiempo actual y también añade un enlace o un puntero al documento anterior, y se emite un "certificado" con esta información. El puntero en cuestión es un puntero especial que se vincula a un dato en lugar de a una ubicación. Eso significa que, si los datos cambian, el puntero se invalidará automáticamente.

Lo que esto logra es que el certificado de cada documento garantiza la integridad del contenido del documento anterior. De hecho, cada certificado soluciona toda la historia de documentos y certificados hasta ese momento. Si suponemos que cada cliente en el sistema realiza un seguimiento de al menos unos pocos certificados, sus propios certificados de documentos y los de los documentos anteriores y siguientes, colectivamente los participantes pueden asegurar que toda la historia de los documentos permanece inalterable hasta ese momento y se conserva el orden de los documentos. A continuación, en ***Figura 4*** se puede ver que para crear un certificado para un documento, el sellado de la marca del tiempo incluye un puntero hash que enlaza al certificado del documento anterior, la marca del tiempo actual y firma los elementos juntos.

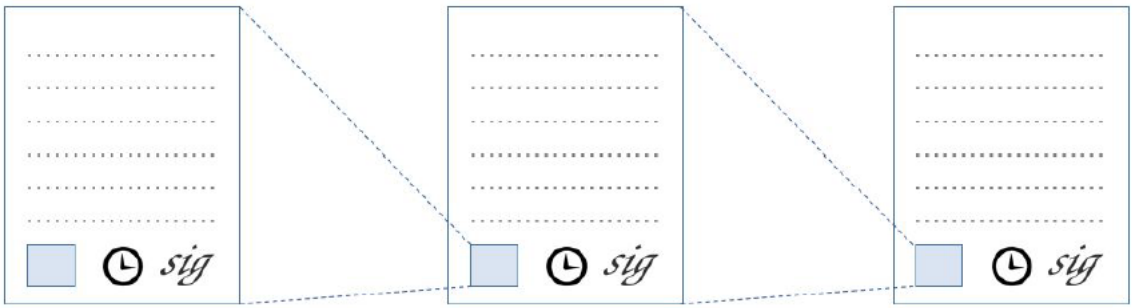

Figura 4: Sellado de marcas de tiempo [1, Fig 4].

Un documento posterior propuso una mejora de la eficiencia. En lugar de vincular los documentos de forma individual, se pueden agrupar en bloques y unir los bloques entre sí formando una cadena de bloques o *blockchain*. Dentro de cada bloque, los documentos de nuevo están unidos entre sí, pero en una estructura de árbol en lugar de linealmente. Esto disminuye la cantidad de comprobación necesaria para verificar que un documento en particular aparece en un punto particular en la historia del sistema. Esta estructura de datos es el esqueleto de la cadena de bloques de *Bitcoin*. En la ***Figura 5*** se pueden ver las flechas representan punteros hash y las líneas verticales representan intervalos de tiempo.

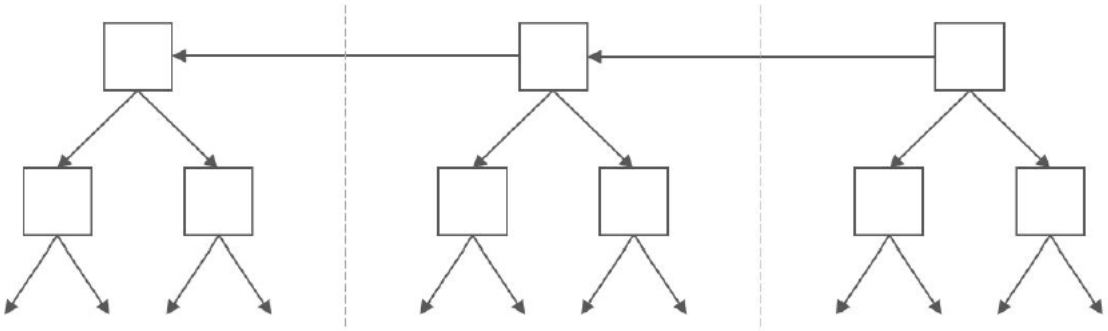

Figura 5: Eficiencia del sellado de marcas de tiempo [1, Fig 5].

Bitcoin refina la eficiencia de una forma sutil pero importante. *Bitcoin* utiliza el protocolo visto anteriormente, el Hashcash, para retrasar los nuevos bloques que se añaden a la cadena de bloques. Esta modificación tiene consecuencias favorables para la seguridad de *Bitcoin.*

Ya hay necesidad de servidores confiables, en su lugar los eventos son guardados por una colección de nodos no confiables llamados mineros. Cada minero realiza un seguimiento de los bloques, en lugar de depender de los usuarios para hacerlo. Cualquiera puede convertirse en minero al resolver el problema computacional para crear bloques. *Bitcoin* confia en los punteros hash para garantizar la integridad de los datos. Finalmente, las marcas de tiempo no son de mucha importancia en *Bitcoin* ya que el objetivo del sistema es registrar el orden relativo de las transacciones de forma inalterable. Los bloques en *Bitcoin* no se crean en un tiempo predeterminado. El sistema asegura que se crea un bloque nuevo cada 10 minutos en promedio, pero hay una variación de tiempo considerable entre bloques adyacentes. Más adelante profundizaremos sobre este tema.

Bitcoin combina la idea de usar problemas computacionales para regular la creación de nuevas unidades de moneda con la idea de sellado de marca de tiempo para registrar en un libro de transacciones y solucionar el problema de doble gasto.

Claves hash y Estructura de datos

Para entender que es *Blockchain* antes se debe conocer que es un puntero hash. Un puntero hash es una clave producida por un hash criptográfico que almacena el valor de cierta información. Un puntero general apunta a un lugar de memoria y es una manera de recuperar la información mientras que un puntero hash además proporciona una manera de verificar que la información no ha cambiado. En la ***Figura 6*** se puede ver que un puntero hash es un puntero o clave producida por una función hash criptográfica que almacena el valor de cierta información.

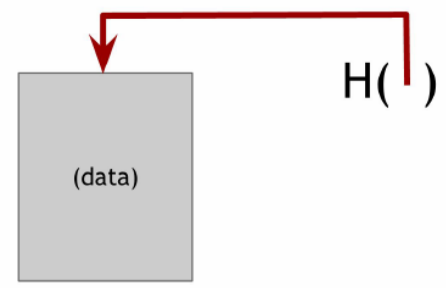

Figura 6: Puntero hash [2, Fig 1.4].

Una lista de punteros hash enlazados entre sí es lo que se conoce como *Blockhain.* Una lista enlazada regular tiene una serie de bloques, donde cada bloque tiene datos y un puntero que apunta al bloque anterior. En *Blockchain* el puntero que apunta al bloque anterior es un puntero hash. Cada bloque apunta al bloque anterior y también contiene el resumen del valor de la información del bloque con lo que permite verificar que el valor del bloque permanece inalterable. En la cabecera de cada bloque hay un puntero hash que apunta al bloque de datos anterior. En la ***Figura 7*** se observa una cadena de bloques es una lista enlazada con punteros hash, donde cada bloque tiene un puntero hash que apunta al bloque anterior.

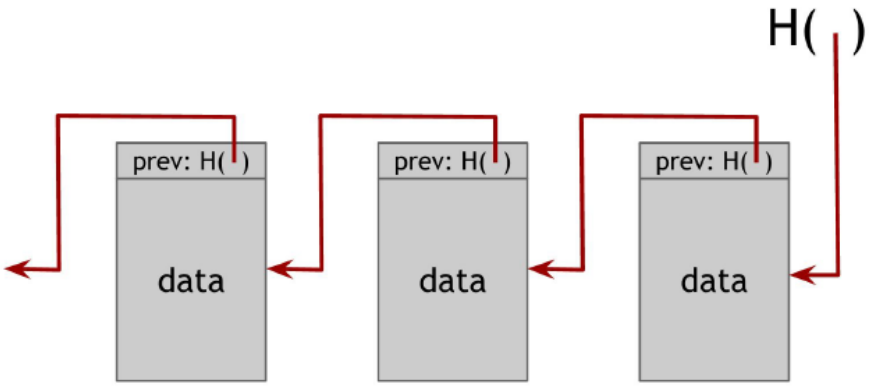

Figura 7: Blockchain [2, Fig 1.5].

Bitcoin utiliza *Blockchain* para tener un registro inalterable. Es una estructura de datos que permite almacenar una gran cantidad de datos y si alguien altera los datos históricos del registro se detecta fácilmente que la información ha sido alterada. Para entender porque *Blockchain* tiene la propiedad de que los datos almacenados no puedan ser modificados se va a mostrar el siguiente ejemplo.

Se puede imaginar que un individuo malicioso tiene como objetivo alterar los datos que se encuentran en un bloque en medio de la cadena de bloques. Para lograr ese objetivo el individuo malicioso cambia los datos de un bloque k. Puesto que los datos del bloque han sido

modificados, el hash de todo el bloque k no va a coincidir. Como se ha visto anteriormente, esta estadísticamente demostrado que la función hash es resistente a colisiones.

Por lo tanto, se detectará la inconsistencia entre los datos nuevos del bloque k y el puntero de hash en el bloque $k+1$. El individuo malicioso puede intentar modificar el hash del siguiente bloque y puede continuar con esta estrategia, sin embargo, fracasará cuando llegue a la cabecera de la lista siempre que se guarde el puntero hash al principio de la lista. Esto permite que un individuo malicioso no pueda modificarlo. De esta manera se detecta si alguien ha alterado la información de un bloque de la cadena.

De tal forma es posible construir una cadena de bloques que tenga tantos bloques como se quiera, con los punteros hash que apuntan al bloque anterior es posible ir retrocediendo de uno a otro hasta llegar al primer bloque llamado génesis.

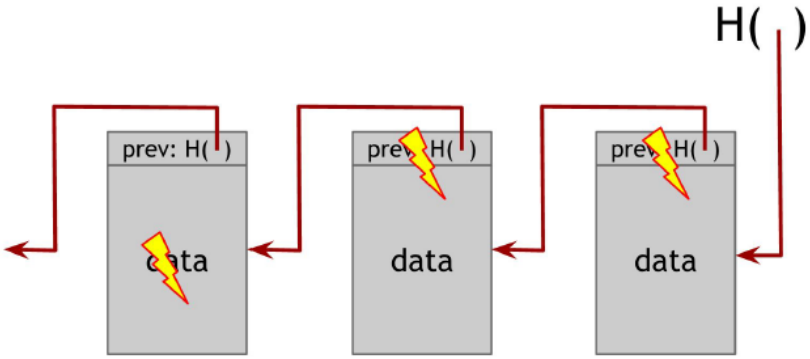

Figura 8. Blockchain a prueba de manipulaciones [2, Fig 1.6].

En la **Figura *8*** se puede ver que si un individuo malicioso modifica los datos de un bloque de la cadena de bloques sucederá que el puntero hash del siguiente bloque será incorrecto. Si se guarda la cabecera de la lista, aunque el individuo malicioso modifique todos los punteros para que sean consistentes con los datos modificados, el puntero de la cabecera será incorrecto y se detectará la modificación.

Estructura de Árbol Merkle

Bitcoin utiliza la estructura de datos mediante punteros hash en un árbol binario. Esta estructura se conoce como Árbol de Merkle, su creador fue Ralph Merkle [6]. Le estructura es la siguiente. Se supone que hay varios bloques que contienen datos, estos bloques son los nodos hoja del árbol, es decir, los nodos que no tiene hijos. Estos nodos se agrupan en pares de dos y para cada par, se crea una estructura de datos que tiene dos punteros hash, uno para cada bloque. Esta estructura es el siguiente nivel del árbol.

A su vez, se agrupan en pares de dos, y para cada par se crea una nueva estructura de datos que contiene el hash de cada uno. Sería el siguiente nivel del árbol. Realizaremos esta operación hasta llegar al nodo raíz del árbol, es decir, al nodo superior del árbol. En la **Figura *9*** se puede ver la estructura de datos Árbol de Merkle, los bloques de datos son agrupados en pares y el hash de cada uno de estos bloques es almacenado en el nodo padre. Los nodos padres son agrupados en pares y sus hashes guardados en el siguiente nivel del árbol. Esto continúa en todo el árbol hasta llegar al nodo raíz.

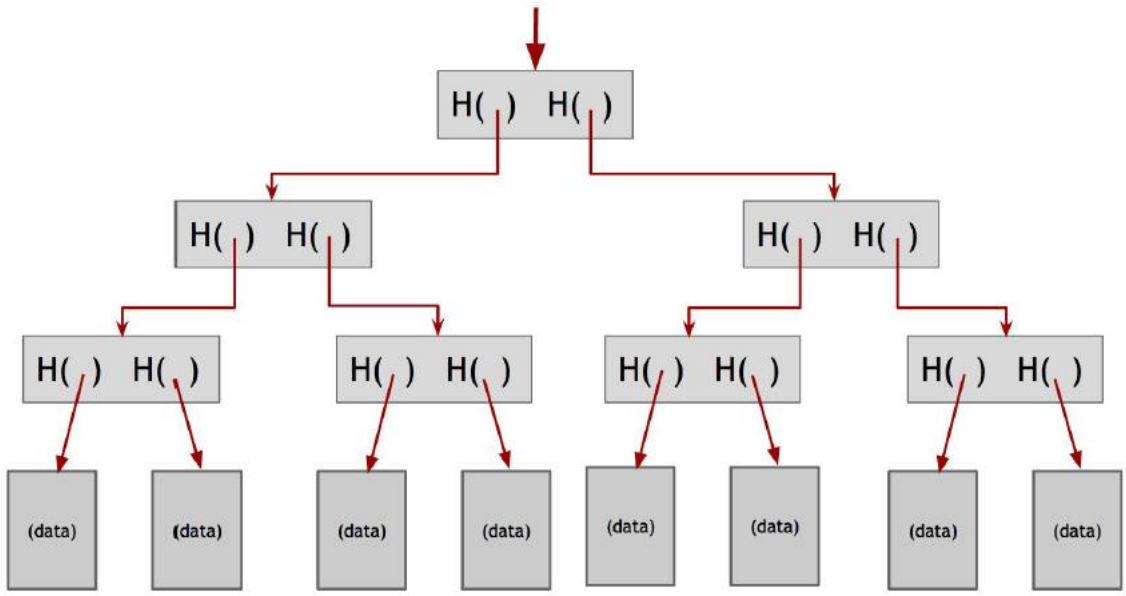

Figura 9. Árbol de Merkle [2, Fig 1.7].

Al igual que anteriormente, se guarda únicamente el puntero hash del nodo raíz del Árbol de Merkle. Con esta estructura se puede recorrer cualquier nodo de la lista asegurando que los datos no han sido modificados. Como se ha explicado anteriormente, si un individuo malicioso manipula algún bloque en la parte inferior del árbol, sucederá que el puntero hash que está en el nivel superior no coincida con el que está guardado. Así pues cualquier intento de modificación de datos se detectará simplemente con recordar el puntero hash del nodo raíz.

Una propiedad que ofrece el Árbol de Merkle es que a diferencia de la cadena de bloques nos permite conocer si una información concreta es miembro de un Árbol de Merkle. De este modo, se mostrará la ruta desde el bloque de datos hasta el nodo raíz. El resto del Árbol de Merkle se ignora debido a que únicamente con este camino es suficiente para comprobar los hashes de todo el camino hasta la raíz del árbol.

Si hay n nodos en el árbol y se quiere verificar un bloque concreto solo deben mostrarse la ruta que sigue ese bloque. Puesto que para cada paso se debe calcular el hash del bloque secundario, el tiempo es aproximadamente $log(n)$ para verificarlo.

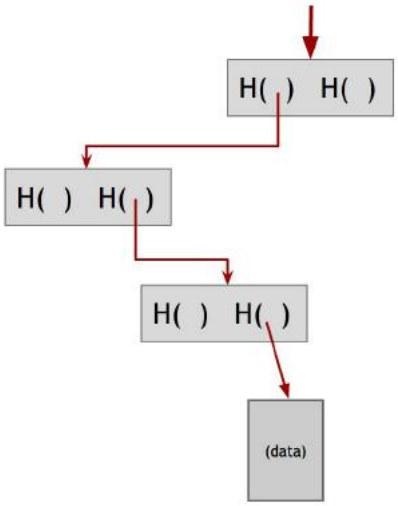

Figura 10. Proof of membership [2, Fig 1.8].

En la **Figura *10*** se observa que para comprobar que un bloque de datos está incluido en el árbol, solamente se necesita mostrar la ruta del bloque de datos hasta el nodo raíz.

2.1.3 Firmas digitales

Para continuar, en este apartado se explican las firmas digitales. Una firma digital es un el equivalente digital a una firma escrita en papel. Se quiere conseguir que las firmas digitales tengas dos propiedades iguales que las firmas escritas en papel. Primero, solamente una persona puede realizar su firma, sin embargo, cualquier persona puede verificar que sea válida. En segundo lugar, la firma debe vincularse a un mensaje concreto para que la firma no pueda utilizarse en un mensaje diferente.

Para ver que propiedades posee la firma digital primero se verá el esquema de firma digital que consiste en tres algoritmos.

$(sk, pk) := generateKeys(keysize)$ El método $generateKeys$ tiene como parámetro el tamaño de una clave y genera un par de claves. La clave secreta sk, utilizada para firmar mensajes y la clave pública pk, utilizada para verificar mensajes. Cualquier persona que conozca la clave pública pk puede verificar la firma realizada con la clave privada sk.

$sig := sign(sk, message)$ El método $sign$ tiene como parámetro la clave privada sk y un mensaje $message$ y genera una firma para el mensaje $message$ con la clave privada sk.

$isValid := verify(pk, message, sig)$ El método $verify$ tiene como parámetros la clave pública pk, el mensaje $message$ y la firma sig. Devuelve un valor booleano $isValid$, que será $true$ si sig es una firma válida para el mensaje $message$ con la clave pública pk. En cualquier otro caso el valor devuelto será $false$.

Como requisito se deben cumplir las siguientes dos propiedades:

Propiedad 1: Las firmas válidas sean verificables.

$$\boldsymbol{verify}\,(\,pk\,, message\,, \boldsymbol{sign}\,(\,sk\,, message\,))\;==\;\boldsymbol{true}$$

Propiedad 2: Las firmas no se puedan falsificar.

Se puede observar que $generateKeys$ y $sign$ pueden ser algoritmos aleatorios. Concretamente $generateKeys$ debería ser aleatorio para que al generar pares de claves a diferentes personas fueran diferentes. Por otra parte, $verify$ debe ser determinista.

La primera propiedad es que las firmas válidas sean verificables, es decir, si Alice firma un mensaje con su propia clave secreta y luego Bob intenta validar esa firma sobre ese mensaje utilizando la clave pública de Alice, la firma se debe validar correctamente.

La segunda propiedad es que las firmas no pueden ser falsificadas. Es decir, que, si un individuo malicioso conoce la clave pública de otra persona y puede ver la firma de la otra persona en otros mensajes, no es posible que pueda falsificar su firma sin conocer la clave privada.

En consecuencia, el esquema de firma digital, independientemente del algoritmo que utilice el individuo malicioso, es no falsificable debido a que la probabilidad de forzar una firma en un mensaje sin conocer la clave privada es tan pequeña que nunca sucederá en la práctica.

Hay una serie de preocupaciones a lo hora de poner este mecanismo de firma digital en práctica:

- En primer lugar, el algoritmo elegido debe tener una buena aleatoriedad como es el caso de *Bitcoin*. Es importante no subestimar esta característica, ya que una mala aleatoriedad hará que el algoritmo sea inseguro.
- Segundo, hay un límite en el tamaño del mensaje que se puede firmar porque los esquemas de firma digital operan sobre cadenas de bits de longitud limitada. Existe una forma de superar esta limitación y es, en lugar de firmar el mensaje, firmar el hash criptográfico del mensaje. Si se utiliza una función hash criptográfica de 256 bits de salida es posible firmar mensajes de cualquier tamaño, siempre que el esquema de firma digital permita firmar mensajes de 256 bits. Como se ha comentado anteriormente es seguro utilizar el hash del mensaje ya que es resistente a colisiones.
- En tercer lugar, si se firma un hash criptográfico, la firma protege toda la estructura. No únicamente ese hash criptográfico sino todo a lo que apunta la cadena de punteros hash. Por lo tanto, si se firma el hash criptográfico que se encuentra al final del

Blockchain, el resultado es que se estaría firmando digitalmente la cadena de bloques completa.

ECDSA

Bitcoin utiliza un esquema concreto de firma digital que se llama *Elliptic Curve Digital Algorithm (ECDSA). ECDSA* se puede ver [7] y es un estándar del gobierno de *EE.UU.* Se trata de una actualización del algoritmo que se utilizaba anteriormente llamado *Digital Signatura Algorithm (DSA*). A lo largo de años se han realizado análisis de estos algoritmos y se cree que son seguros.

Concretamente, *Bitcoin* utiliza *ECDSA* sobre la curva elíptica "secp256k1" para proporcionar de forma estimada 128 bits de seguridad. Significa que para romper el algoritmo se necesitaría de promedio realizar 2^{128} operaciones criptográficas de clave simétrica para obtener la clave de la función hash correcta. Eso son muchas operaciones.

A continuación, se muestra el tamaño de las siguientes claves de *Bitcoin*:

Clave privada: 256 bits

Clave pública, sin comprimir: 512 bits

Clave pública, comprimida: 257 bits

Mensaje a firmar: 256 bits

Firma: 512 bits

Aunque *ECDSA* técnicamente puede firmar mensajes de 256 bits de longitud, no es un problema debido a que los mensajes siempre son claves hash antes de ser firmados. Con lo se puede firmar un mensaje de cualquier tamaño de forma eficiente.

Claves públicas como identidades

Para entender el esquema de firma digital se puede escoger una clave pública, es decir, una del clave de verificación pública del esquema de firma digital y se puede comparar a la identidad de una persona. Si se observa un mensaje con una firma, verificado correctamente bajo una clave pública pk, se puede pensar que la clave pública pk estaría diciendo el mensaje. Desde este punto de vista, la clave pública pk es una identidad. Para que alguien pueda hablar en nombre de la identidad pk debe de conocer la clave secreta sk correspondiente.

Una consecuencia de las claves públicas como identidades es que una persona puede crear una nueva identidad siempre que quiera. Si se genera un nuevo par de claves nuevas pk y sk, a través de $generateKeys$ en el esquema de firma digital, la nueva clave pública pk

corresponderá a la nueva identidad mientras que la clave secreta sk correspondiente será la que permita hablar en nombre de la nueva identidad pk.

Si se utiliza el hash de pk como identidad porque las claves públicas son grandes, por tanto para verificar que un mensaje proviene de la identidad pk, se deberá verificar:

1. Que pk concreto es su identidad
2. Que el mensaje se verifica bajo la clave pública pk.

Gestión de la identidad descentralizada

Con este esquema se obtiene un sistema con identidades descentralizadas. En lugar de tener que ir a una autoridad central para registrarse, una persona se puede registrar como usuario del sistema por su cuenta. No se necesita un nombre de usuario. Si se quiere una nueva identidad se puede generar una nueva identidad en cualquier momento y se pueden realizar tantas identidades como se deseen.

Bitcoin utiliza la gestión de identidad descentralizada de esta manera. Estas identidades se llaman direcciones en terminología de *Bitcoin* y en otras criptomonedas. Por tanto, cuando se haga referencia a dirección realmente será un hash de una clave pública.

2.2 Descentralización

El protocolo de *Bitcoin* permite mantener el registro de las transacciones o *Blockchain,* además de la autoridad sobre qué transacciones son válidas y la creación de nuevos *bitcoins* de forma descentralizada. Es un concepto importante en *Bitcoin* la descentralización. Casi ningún sistema es totalmente centralizado o descentralizado. En el caso de *Bitcoin*, el protocolo es descentralizado, sin embargo, el lugar donde se pueden intercambiar *bitcoins* en otras divisas, aplicaciones que permiten administrar *Bitcoins* pueden estar centralizados o descentralizados en diferentes grados.

En primer lugar, la red *peer-to-peer* está casi totalmente descentralizada, ya que cualquier persona podría ejecutar un nodo *Bitcoin*. Es posible conectarse en línea y descargar fácilmente un cliente de *Bitcoin* y ejecutar un nodo en su dispositivo electrónico. Actualmente hay varios miles de nodos.

Segundo, la minería *Bitcoin* que cualquier persona podría realizar, aunque se requiere un alto coste de capital. Por esta razón hay un alto grado de centralización o concentración de poder, en el ecosistema de minería de *Bitcoin*.

Un tercer aspecto son las actualizaciones del software que ejecutan los nodos de *Bitcoin*, y esto influye en cómo y cuándo cambian las reglas del sistema. En la práctica, la mayoría de los nodos ejecutan la implementación de referencia y sus desarrolladores son confiables para la comunidad. A continuación, se explicará el consenso distribuido, el sistema de incentivos, la minería y el sistema *Bitcoin*.

2.2.1 Consenso distribuido

El consenso distribuido tiene varias aplicaciones. La aplicación tradicional es la confiabilidad en sistemas distribuidos. Existen sistemas que miles o incluso millones de servidores, que en conjunto forman una gran base de datos distribuida que registra todas las acciones que ocurren en el sistema. Cada pieza de información debe ser almacenada en varios nodos diferentes, y los nodos deben estar sincronizados con el estado del sistema.

En el protocolo de consenso distribuido existen n nodos, donde cada uno tiene un valor como entrada. Algunos de estos nodos son maliciosos o defectuosos. Un protocolo de consenso distribuido tiene las siguientes dos propiedades:

1. Debe llegar a un acuerdo de valor común junto con todos los nodos honestos.

2. El valor debe haber sido generado por un nodo honesto.

En el contexto de *Bitcoin*, un nodo es un miembro de la red *peer-to-peer* de *Bitcoin.* Cuando Alice quiere pagar a Bob, lo que hace es transmitir una transacción a todos los nodos de *Bitcoin* que componen la red *peer-to-peer*. En **Figura *11*** se puede ver que *Alice* para pagar a Bob transmite una transacción a todos los nodos de la red *peer-to-peer* de *Bitcoin*.

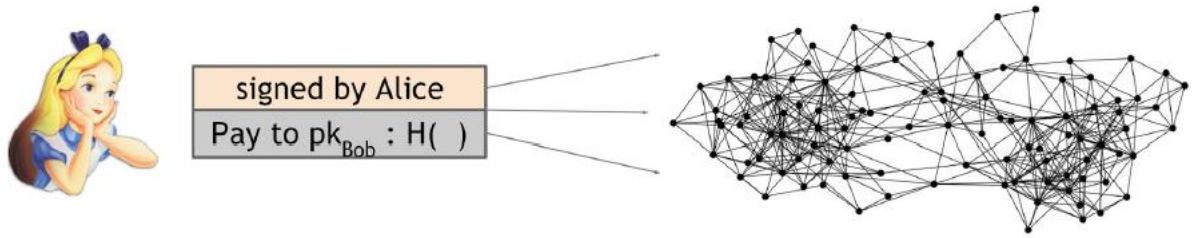

Figura 11. Transmisión de una transacción [2, Fig 2.1].

Puesto que hay una variedad de usuarios que están transmitiendo transacciones a la red, los nodos deben acordar exactamente qué transacciones se realizaron y el orden en que ocurrieron. Esto dará como resultado un único libro de contabilidad global para el sistema. En *Bitcoin*, el consenso es bloque por bloque.

En cualquier punto dado, todos los nodos en la red *peer-to-peer* tienen un libro de contabilidad que consiste en una secuencia de bloques y una lista de transacciones en la que todos los nodos han llegado a un consenso.

Además, cada nodo tiene un conjunto de transacciones pendientes que aún no han sido incluidas en *Blockchain*. Para estas transacciones, el consenso no ha sucedido, por tanto, cada nodo puede tener una versión ligeramente diferente del conjunto de transacciones pendientes. En práctica, esto ocurre porque la red *peer-to-peer* no es perfecta, por lo que algunos nodos pueden haber recibido una transacción que otros nodos no.

Los nodos llegan a un consenso de la siguiente manera. En intervalos regulares, aproximadamente cada 10 minutos, cada nodo en el sistema propone su propio conjunto de transacciones pendientes para ser el siguiente bloque de la cadena de bloques. Luego, los nodos ejecutan un protocolo de consenso, donde la entrada de cada nodo es la propia lista de transacciones pendientes para el bloque propuesto.

Existe la posibilidad de que algunos nodos puedan ser maliciosos e introducir transacciones no válidas en sus bloques, pero podemos suponer que otros nodos serán honestos. Si el protocolo de consenso tiene éxito, la salida será válida siempre que el bloque sea válido. Otra posibilidad es que alguna transacción pendiente válida no haya sido incluida en el bloque, sin embargo, esto no es un problema. Si una transacción válida no entro en el bloque actual, simplemente esperaría y entraría en el siguiente bloque.

En el protocolo de consenso distribuido en *Bitcoin* se deben tener en cuenta una serie de problemas adicionales. Como, por ejemplo, que los nodos puedan ser completamente maliciosos, que la red peer-to-peer sea imperfecta, que haya fallos de conexión, la gran latencia en el sistema que se distribuye a traves de Internet. No se entrará en este nivel de detalle. Cuando se tenga una sólida comprensión técnica de cómo funciona *Bitcoin*, se tendrá una garantía de estabilidad y seguridad de *Bitcoin.*

Algoritmo de consenso de Bitcoin

Este algoritmo se simplifica en la medida en que asume la capacidad de seleccionar un nodo aleatorio de una manera que no es vulnerable a los ataques de Sybil [8].

1. Las transacciones nuevas se transmiten a todos los nodos.

2. Cada nodo recoge nuevas transacciones en un bloque.

3. En cada ronda, un nodo aleatorio puede emitir su bloque.

4. Otros nodos aceptan el bloque solo si todas las transacciones en él son válidas.

5. Los nodos expresan su aceptación del bloque al incluir su hash en el siguiente bloque que crean.

2.2.2 Incentivos

En *Bitcoin* se introduce la idea de incentivos, algo novedoso en el protocolo de consenso distribuido. Es un mecanismo para incentivar a los participantes para actuar de forma honesta. Existen dos mecanismos de incentivos separados en *Bitcoin*. El primero es la **recompensa** por bloque, es decir, una recompensa para el nodo que crea un bloque, este incluye una transacción especial en el bloque que es la creación de *bitcoins*.

Esta recompensa que en el inicio de *Bitcoin* eran 50 *bitcoins*, actualmente son 12,5 puesto que cada 210,000 bloques se reduce la recompensa a la mitad. Un bloque se genera aproximadamente cada 10 minutos de promedio. Es decir, que aproximadamente cada cuatro años la tasa de creación de *bitcoins* se reduce a la mitad. Es la única manera de introducir nuevos *bitcoins* en el sistema. De esta manera tenemos un número máximo de *bitcoins* que son 21 millones. En **Figura *12*** se puede ver La evolución de la recompensa por bloque generado se reduce a la mitad aproximadamente cada 4 años.

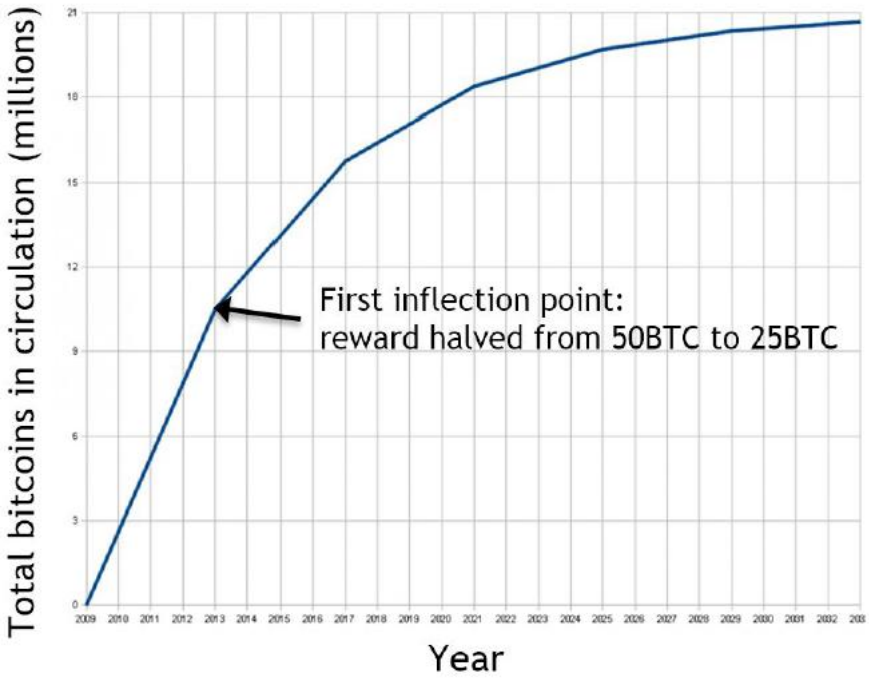

Figura 12. Evolución de la recompensa por bloque [2, Fig 2.4].

El segundo incentivo es la **tasa de transacción**. Al hacer una transacción existe una ligera diferencia entre el valor que se emite y el valor menor se recibe. Esta diferencia es la tarifa de transacción que se incluye como recompensa para el nodo que crea el bloque. De tal modo

que, se supone que se crea un bloque en el que están recogidas n transacciones, luego la suma de las tarifas de esas n transacciones se pagan a la dirección *Bitcoin* que genera el bloque.

2.2.3 Minería

Todavía quedan algunos problemas con el mecanismo de consenso distribuido. El primero, es elegir un nodo aleatorio para la creación de bloques nuevos. En segundo lugar, se ha propuesto un nuevo problema al proporcionar a los nodos estos incentivos para la participación. El sistema puede volverse inestable, ya que los incentivos causan que todos los nodos quieran ejecutar un nodo *Bitcoin* con la esperanza de capturar algunas de estas recompensas. Y un tercero es una versión aún más complicada de este problema, que es que un adversario podría crear una gran cantidad de nodos maliciosos para tratar de modificar el proceso de consenso.

Todos estos problemas están relacionados y todos tienen la misma solución, *proof-of-work*. La idea clave detrás *proof-of-work* es conseguir la selección de un nodo para la generación de un bloque sea un proceso aleatorio y asegurarse de que nadie pueda monopolizarlo. Dicho de otro modo, *proof-of-work* es un mecanismo donde los nodos compiten entre ellos utilizando la potencia de cálculo.

Bitcoin logra la solución utilizando un problema computacional basado en funciones hash. Para crear un bloque, el nodo que propone ese bloque es necesario que encuentre un *nonce* específico. En criptografía, el término *nonce* se utiliza para referirse al valor que solamente puede utilizarse una vez.

De modo que se concatena el *nonce*, el hash anterior y la lista de transacciones que comprende ese bloque. Se calcula el hash de la cadena completa y, si esa salida hash cumple con la condición objetivo, se generará un nuevo bloque. En este caso, el *nonce* tendrá que satisfacer la siguiente desigualdad:

$$H(nonce \,||\, prev\_hash \,||\, tx \,||\, tx\ H \,||\, \ldots \,||\, tx) \;<\; target$$

La idea detrás de este mecanismo es que sea moderadamente difícil de encontrar un *nonce* que cumpla la condición objetivo. Si la función hash, satisface la propiedad explicada anteriormente de *Puzzle friendliness,* la única forma de resolver este problema computacional hash consiste en probar *nonce* por *nonce* hasta dar con uno que cumpla la condición objetivo.

Con este mecanismo se elimina la opción de elegir nodos al azar, en cambio los nodos compiten entre ellos de forma independiente. Así pues, un nodo encentrará de forma aleatoria

un *nonce* que cumpla la condición objetivo y será ese nodo el que proponga el siguiente bloque. De esta manera el sistema es completamente descentralizado.

Se conoce la minería como el proceso de intentar resolver el problema computacional hash. En él intervienen una serie de nodos, llamados mineros, que compiten entre sí de forma independiente para conseguir generar el siguiente bloque *Blockchain*. Este proceso se mantiene debido a los incentivos del protocolo de consenso distribuido comentados anteriormente.

Hay tres propiedades importantes para los problemas computacionales hash.

1. Dificultad de computación: Deben ser difícil de calcular. A finales del 2014 el nivel de dificultad era de 10^{20} hashes por bloque. En otras palabras el tamaño de espacio objetivo es de solo $1 / 10^{20}$ del espacio de salida de la función hash.

2. Coste parametrizable: Cada 2016 bloques se recalcula la condición objetivo, de tal manera que el promedio entre bloques sucesivos de la red *Bitcoin* sea de aproximadamente 10 minutos. En otras palabras, a 10 minutos por bloque con 2016 bloques, el recalculo de la dificultad de computación se realiza cada 2 semanas.

3. Fácil de verificar: Aunque un nodo haya realizado 10^{20} hashes hasta obtener el nonce adecuado que cumple la condición objetivo debe ser fácil de verificar el resultado por cualquier otro minero.

2.2.4 Sistema Bitcoin

El objetivo de la red peer-to-peer *Bitcoin* es propagar todas las nuevas transacciones y los bloques nuevos a todos los nodos *Bitcoin*. Sin embargo, la red no es perfecta y por tanto la seguridad del sistema *Bitcoin* proviene de *Blockchain* y del protocolo de consenso distribuido.

Cuando se dice que una transacción está incluida en *Blockchain*, realmente significa que la transacción ha logrado numerosas confirmaciones de diferentes nodos. No hay un número fijo de confirmaciones necesarias para incluir la transacción en un bloque. Sin embargo, cuantas más confirmaciones haya recibido del protocolo de consenso distribuido una transacción más probabilidades tiene de que esa transacción se incluya en *Blockchain*.

Después con los problemas computacionales hash y la minería obtenemos solución a los problemas identificados en un consenso distribuido. Los mineros, los nodos que compiten en la creación de nuevos bloques de *Blockchain,* intentan resolver el problema computacional. Al solucionarlo, son recompensados por el esfuerzo con el incentivo por la generación del nuevo bloque y con el incentivo de las tasas de transacciones.

Un punto sutil en el concepto de minería es el siguiente: Alice y Bob son dos mineros diferentes y Alice tiene 100 veces más potencia de cálculo que Bob. Esto no significa que Alice siempre gane la carrera contra Bob para encontrar el próximo bloque. En cambio, Alice y Bob tienen un índice de probabilidad de encontrar el próximo bloque, en la proporción de 100 a 1. A largo plazo, Bob encontrará, en promedio, el uno por ciento del número de bloques que Alice encuentra.

Por otro lado, si la minería fuera muy rentable, más hardware de minería entraría en la red. Provocando el aumento de la tasa de hash que repercutiría en un aumentar de la dificultad de *proof-of-work*, y la recompensa esperada de cada minero se reduciría.

El conceso distribuido es muy importante en *Bitcoin*. En una moneda tradicional, el consenso se determina con la tasa de cambio de la moneda. En *Bitcoin* sucede lo mismo, se necesita llegar a un consenso sobre el valor de *Bitcoin*. Sin embargo en *Bitcoin*, además, es necesario un consenso sobre *Blockchain*. Por ejemplo, Alice posee una cierta cantidad o número de *bitcoins*. Realmente significa que en la red *peer-to-peer Bitcoin,* concretamente en *Blockchain*, la suma de direcciones *Bitcoin* de Alice son el número de *bitcoins* que posee.

En este punto ya se conocen las bases criptográficas, la descentralización en *Bitcoin*, las identidades en *Bitcoin*, cómo se propagan y validan las transacciones, la red peer-to-peer *Bitcoin*, qué es y cómo se utiliza *Blockchain* para lograr el consenso, cómo funcionan los problemas computacionales hash y el proceso de minería. Estos conceptos proporcionan una base sólida para entender el sistema *Bitcoin*.

2.3 Intercambios de criptomonedas

Las criptomonedas se utilizan como medio de intercambio de valor. Primero se va a ver como se almacenan y administrar los *bitcoins*, para después ver los servicios de terceros.

Para realizar intercambios primero veremos la forma más simple de almacenar *bitcoins.* Para poder realizar una transacción de *Bitcoins* se necesita conocer la clave privada y la clave pública. En la práctica almacenar *bitcoins* hay que almacenar y administrar la clave secreta de *Bitcoin*. Para saber cómo almacenar y administrar *bitcoins* se va a tener en cuenta tres objetivos.

1. Disponibilidad: Gastar *bitcoins* propios cuando se quiera.

2. Seguridad: Asegurar de que nadie más puede gastar sus propios *bitcoins*.

3. Conveniencia: La administración de claves debería ser sencillo de gestionar.

El método de administración de claves más simple es almacenarlos en un archivo en su propio dispositivo local como por ejemplo un teléfono inteligente. Esto es ideal para la comodidad ya que permite realizar transacciones de *bitcoins* con solo presionar algunos botones.

Sin embargo, no es tan bueno para la disponibilidad. Es decir, si el dispositivo falla y se tiene que formatear el disco, o si el archivo se vuelve corrupto, las claves se pierden, y también los *bitcoins* asociados a esas claves. De igual modo para la seguridad. Si se pierde el dispositivo o alguien lo roba o se infecta con malware, pueden copiar sus claves y enviar todos los *bitcoins* a sus propias direcciones *Bitcoin*.

En otras palabras, almacenar sus claves privadas en un dispositivo local como un teléfono inteligente es muy similar a llevar dinero en la cartera o en el bolso. Por consiguiente, normalmente lo que se hace es almacenar un poco de información en la cartera y mantener la mayor parte del dinero en otro lugar. Existen servicios de terceros que ayudan al almacenamiento y la administración de *bitcoins*.

2.3.1 Wallets

Si se quiere almacenar *bitcoins* localmente, se usaría un *wallet*, que es un software para realizar un seguimiento de todas sus monedas, administrar todos los detalles de sus claves y tiene una buena interfaz de usuario.

Para realizar transacciones con *bitcoins*, se necesita una forma de intercambiar la dirección a la que se enviarán los *bitcoins*. Hay dos maneras de codificar las direcciones *Bitcoin*: puede ser por una cadena de texto codificada en base 58 o por un código QR. Para codificar una dirección en una cadena de texto, se cogen los bits de la clave y se convierten de un número binario a un número base 58. Luego, se utiliza un conjunto de 58 caracteres para codificar cada dígito como un carácter; esto se llama notación base58.

Se codifica de esa manera porque ese es el número que obtenemos cuando incluimos las letras mayúsculas y minúsculas. No se incluyen en la base58 caracteres que den lugar a confusión. Por ejemplo, la letra mayúscula 'O' y el cero no se incluyen en esta notación porque son muy parecidos. Un ejemplo siguiente muestra la primera dirección *Bitcoin* en recibir la recompensa de generar el bloque génesis codificada en base58.

$$\mathbf{1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa}$$

El segundo método de codificar una dirección *Bitcoin* es con un código QR. Un código de barras bidimensional. La ventaja de un código QR es que puede tomar una fotografía con un teléfono inteligente y el software *wallet* puede convertir automáticamente el código de barras a la dirección de *Bitcoin* correspondiente.

Figura 13. Código QR de una dirección *Bitcoin* [2, Fig 4.1].

2.3.2 Wallets online

Es una cartera digital que el propietario puede administrar, donde se almacena la información en La Nube, y se accede a ella usando una interfaz web o una aplicación en un dispositivo digital. Algunos ejemplos de servicios populares de *wallet* en línea son Coinbase.com [9] o blockchain.com [10].

Si se está dispuesto a utilizar estos servicios de terceros se debe confiar en el código que se ejecuta en el navegador o en la aplicación. Ya que debe ser lo suficientemente seguro para que nadie consiga filtrar las claves o contraseñas almacenadas.

Una ventaja es que si se accede vía página web no es necesario instalar nada en el dispositivo y se puede tener una sola billetera a la que acceda desde la página web o desde su teléfono, y simplemente funcionará porque el *wallet* está ubicado en La Nube.

Por otra parte, si el sitio o las personas que operan en ese servicio resultan ser maliciosos es posible que los *bitcoins* almacenados en el *wallet* online puedan ser robados. Finalmente, se otorga una confianza a estos servicios para administrar los *bitcoins*.

2.3.3 Intercambios de bitcoins

Para entender los intercambios de *bitcoins*, debemos conocer cómo operan los bancos tradicionales. Si una persona va a un banco a depositar dinero, el banco se compromete a devolver ese dinero más tarde. Realmente, el banco coge ese dinero y lo invierte, aunque normalmente el banco se asegura de tener el suficiente dinero como para pagar la demanda de dinero retirado.

Los intercambios de *bitcoins* funcionan de manera similar a los bancos. Aceptan depósitos de *bitcoins,* y al igual que un banco se comprometen a devolver la demanda más tarde. También se pueden realizar enviar moneda fiduciaria como euros, dólares o yenes mediante

transferencias de cuenta bancaria. El servicio de intercambio se compromete a devolver cualquiera o ambos tipos de moneda bajo demanda.

El funcionamiento es que buscan un cliente que quiera comprar *bitcoins* por euros y por otra parte buscan a otro cliente que quiera vender *bitcoins* por euros y si hay un precio aceptable se realiza la transacción.

Si suponemos que una persona tiene en la cuanta 5.000 euros y tres *bitcoins* y quiere comprar 2 *bitcoins*, de tal modo que utiliza el servicio de intercambio y define una orden de comprar por 580 euros cada uno. El servicio de intercambio encuentra a otra persona dispuesto a vender 2 *bitcoins* por lo que la transacción sucede. Después del intercambio la primera persona tendrá 5 *bitcoins* y 3840 euros.

Lo que hay que tener en cuenta es que esta transacción involucra a 2 personas en el mismo intercambio y no ocurrió ninguna transacción en *Blockchain* de *Bitcoin*. Antes el compromiso del servicio de intercambio era “le daremos 5000 euros y 3 *BTC*” y ahora ha cambiado y es “le daremos 3840 euros y 5 *BTC*”. Realmente es un cambio de compromiso y no repercute en la economía del euro ni a través del *Blockchain* de *Bitcoin*.

2.3.4 Mercado de intercambio

Por cambio de moneda se hace referencia a intercambiar *bitcoins* por monedas fiduciarias como euros, dólares, yenes u otras. Se han visto servicios de terceros que permiten estos intercambios ahora se quiere ver desde el punto de vista de un mercado, qué tamaño tiene, como se opera y su economía. Lo primero que se tiene que entender es como funciona un mercado entre dos monedas fiduciarias como el euro y el dólar. El precio fluctúa según la cantidad de gente que quiera comprar euros frente a la cantidad de gente que quiera comprar dólares en un día en concreto.

Existen páginas web como bitcoincharts.com [11] que muestran la tasa de cambio con varias monedas fiduciarias en diferentes servicios de intercambio diferentes. Si se visita el sitio web se puede ver que hay muchas operaciones en curso, y los precios fluctúan en tiempo real a medida que se realizan los intercambios.

Por ejemplo, en localbitcoins.com [12], se puede especificar su ubicación y que desea comprar *bitcoins* con efectivo. Existen personas dispuestas a intercambiar *bitcoins* en lugares concretos y se especifica la cantidad y el precio. Después es posible ponerse en contacto para organizar una reunión y realizar el intercambio en cualquier lugar público. Una razón por la que alguien quisiera obtener *bitcoins* en persona en lugar de a través de internet es porque una transacción en un lugar público puede considerarse anónima. En cambio, una transacción a través de una

cuenta en un servicio de intercambio normalmente se requiere la presentación de una identificación emitida por el gobierno debido a la regulación bancaria.

Nombre del servicio	URL del servicio
Bisq	https://bisq.network
Binance	https://www.binance.com
Bitstamp	https://www.bitstamp.net
Bittrex	https://bittrex.com
Bitwage	https://www.bitwage.com
Coinmama	https://www.coinmama.com
Hodl Hold	https://hodlhodl.com
Kraken	https://www.kraken.com
Local Bitcoins	https://localbitcoins.com
Poloniex	https://poloniex.com

Tabla 1. Lista de los principales servicios de intercambio *bitcoins* [13].

2.3.5 Oferta y demanda

Como cualquier mercado, el mercado el intercambio de *Bitcoins* coincide en que los compradores quieren comprar y los vendedores que quieren realizar la acción contraria. Es un gran mercado que mueve varios millones de euros y dólares. Es lo suficientemente grande como para que una persona que quiere entrar en el mercado puede comprar o vender, al menos una cantidad modesta, siempre podrá encontrar una contraparte. El precio del mercado se establece según la oferta y la demanda.

Se va a entrar más en detalle. La oferta de *bitcoins* es el número de *bitcoins* que es posible adquirir en uno de estos mercados, y es igual al número de *bitcoins* que están en circulación actualmente. Este número es fijo. Como se ha visto anteriormente el suministro de *bitcoins* llegará a un límite de 21 millones y actualmente a mediados del 2018 hay unos 17 millones de *BTC*. Donde un *bitcoin* puede tener un valor con 8 decimales de precisión y el valor más pequeño posible es 0.00000001 *bitcoins* y se denomina Satoshi.

Existen dos tipos de demanda de *bitcoins.*

- El primero es una manera de realizar intercambios de valor al igual que las monedas fiduciarias mientras que el segundo es como método de inversión. En el primer caso suponemos que Alice quiere transferir una cantidad de dinero a Bob y deciden utilizar

Bitcoin para realizar la transferencia. Se supone que ninguno de los dos quiere conservar los *bitcoins* a largo plazo. Por lo que Alice compraría *bitcoins* por euros y haría la transferencia de *bitcoins* a Bob. Bob al recibir la transferencia realizaría el cambio de *bitcoins* a euros. Lo que sucede desde un punto de vista de demanda de *bitcoins* es que, durante el tiempo que dura la transacción los *bitcoins*, estos tienen que ser retirados de la circulación. Este hecho crea demanda de *bitcoins*.

- El segundo caso es una demanda como una inversión. Alguien compra *bitcoins* en un momento dado con la esperanza que en un futuro su precio aumente y al venderlos obtengan beneficios. Si estas personas compran y mantienen sus *bitcoins*, estos quedan fuera de circulación. Debido a lo cual si el precio es bajo, muchas personas querrán comprar *bitcoins* como inversión mientras que si el precio es alto la demanda no será tan alta.

Modelo simple de comportamiento del mercado *Bitcoin*

Ahora se realizará un modelo simple de comportamiento del mercado para observar el movimiento de las transacciones y el efecto que podría tener en el precio de los *bitcoins*. Empezaremos definiendo algunos parámetros.

$\boldsymbol{T}$ es el valor total medio de las transacciones de *bitcoins* por todos los que participan en el mercado. Este valor se mide en euros por segundo. Por simplicidad se supone que las personas que desean realizar estas transacciones tienen en mente un cierto valor en euros de las transacciones, o de alguna otra moneda fiduciaria que se traduce en euros. De tal manera, hay una cierta cantidad de euros por segundo de transacciones en *bitcoins*.

$\boldsymbol{D}$ es la duración del tiempo que los *bitcoins* deben estar fuera de circulación mientras se realiza una transacción. Ese es el tiempo desde que el pagador compra los *bitcoins* hasta cuando el receptor puede venderlos nuevamente al mercado. Se mide en segundos.

$\boldsymbol{S}$ es el suministro total de *bitcoins* que están disponibles para esta compra. Suponemos que están disponibles, actualmente alrededor de 17 millones menos aquellos que las personas sostienen como inversiones a largo plazo. En otras palabras, estamos hablando de las *bitcoins* que circulan y están disponibles para las transacciones.

Finalmente, $\boldsymbol{P}$ es el precio de *bitcoin*, medido en euros.

Ahora se realizan algunos cálculos. Primero se calcula cuántos *bitcoins* estarán disponibles para dar servicio a las transacciones cada segundo. Hay $\boldsymbol{S}$ *bitcoins* disponibles en total y debido durante un tiempo de $\boldsymbol{D}$ segundos se retiran de la circulación, tenemos de promedio

cada segundo la fracción de *bitcoins* S/D que son los que volverán a estar disponibles para realizar transacciones cada segundo**.** Esta es la oferta.

En la demanda, el número de *bitcoins* por segundo que se necesitan para realizar las transacciones es de $1/P$ euros. Por lo tanto, T/P es la cantidad de *bitcoins* por segundo que se necesitan para cubrir la demanda.

Si se observa un segundo en particular, hay una oferta de S/D y una demanda de T/P . En este mercado, como la mayoría de los mercados, el precio fluctuará para alinear la oferta con la demanda.

Si la oferta es más alta que la demanda, hay *bitcoins* que no se venderán, por lo que las personas que venden *bitcoins* estarán dispuestas a reducir su precio de venta para poder venderlos. De acuerdo con la fórmula T/P para la demanda, cuando el precio baja, la demanda aumenta. Por esta razón la oferta y la demanda tienden a alcanzar el equilibrio.

Por otro lado, si la oferta es menor que la demanda, significa que hay personas que quieren comprar *bitcoins* no pueden obtenerlos porque no hay suficientes *bitcoins*. Por lo cual estas personas tendrán que pujar más para obtener sus *bitcoins* porque habrá mucha competencia por una oferta limitada de *bitcoins*. Por lo tanto, el precio aumenta lo que significa que la demanda se reducirá hasta que haya un equilibrio. Es decir, la oferta debe ser igual a la demanda, obteniendo la siguiente fórmula:

$$\frac{S}{D} = \frac{T}{P}$$

Lo que nos da para el precio:

$$P = \frac{T \cdot D}{S}$$

Se podría simplificar aún más suponiendo que D**,** la duración para la cual se necesita mantener un *bitcoin* fuera de circulación mientras se realiza una transacción no cambia. Además suponemos que el suministro S tampoco cambia, o al menos cambia lentamente con el tiempo. Lo que significa que el precio es proporcional a la demanda. Con lo cual, si la demanda se duplica, el precio de los *bitcoins* se duplicaría.

Se observa que suministro total S incluye solo los *bitcoins* que no se mantienen como inversión. Si más personas están comprando *bitcoins* como inversión, S se reducirá y por tanto P aumentará. Es decir, que si hay más demanda como inversión, el precio del *bitcoin* aumentará.

Este es un modelo simplificado del mercado. Para tener un modelo más completo se deberá tener en cuenta la actividad de los inversores. Es decir, los inversores comprarán *bitcoins*

cuando crean que el precio aumentará en un futuro por lo que se deberán incluir en el modelo las expectativas de los inversores. Y estas expectativas son la demanda que se espera en un futuro.

Como conclusión, existe un mercado entre *bitcoins* y euros, y entre *bitcoins* y otras monedas fiduciarias. Ese mercado tiene suficiente liquidez que puede comprar o vender en cantidades modestas. El precio se rige por la oferta y la demanda que hace que el precio del *bitcoin* fluctúe.

Es conocida la volatilidad del valor del bitcoin y de las criptomonedas, se debe a que las personas especulan con ellas. En 2009 el valor técnico de un bitcoin era de 0 € mientras que a finales de 2017 el valor de un bitcoin llego a superar los 17.000 €. Con estas cifras se puede observar la gran fluctuación que existe en el precio.

Es posible realizar modelos económicos para obtener una idea sobre como la oferta y la demanda interactúan en este mercado y de esta manera poder predecir lo que el mercado podría hacer. Se deberá tener en cuenta como estimar variables desconocidas, como por ejemplo la demanda futura de *bitcoins.*

2.4 Tipos de criptomonedas

Bitcoin es solo un miembro (aunque muy importante) del ecosistema de criptomonedas. Existen otras criptomonedas similares a *Bitcoin* llamadas *alternative coins* o *altcoins.*

Como *Bitcoin* es *open source,* significa que muchos desarrolladores pueden utilizar su código fuente, realizar modificaciones y lanzar sistemas de criptomonedas alternativos. A menudo son ramificaciones o bifurcaciones que tiene el código fuente de *Bitcoin* y generan otras criptomonedas. Desde la creación del *Bitcoin* han surgido varios tipos de criptomonedas con características y protocolos distintos. Por ejemplo: Ethereum, Ripple, Bitcoin Cash, Litecoin entre muchas otras. En [14] se puede ver un listado de las actuales.

2.4.1 Altcoins o criptomonedas

Cada *altcoin* necesita una historia que contar. Normalmente un *altcoin* cambia algunos parámetros de la configuración de *Bitcoin* como por ejemplo el tiempo promedio entre bloques, los incentivos, la tasa de creación de monedas. Es posible que haya diferentes detalles técnicos como, por ejemplo, añadir un lenguaje de scripting para añadir seguridad o expresar diferente tipo de transacciones. Otra posibilidad es que la minería funcione de manera distinta.

A menudo las *altcoins* tienen detrás un proyecto que las respalda, una comunidad y se desarrollan en base a unos temas específicos.

Lo importante de un *altcoin* es que la comunidad o las personas acepten esta criptomoneda y decidan utilizarla, de lo contrario no tendrá ningún valor en el mercado porque nadie quiere utilizar las esa criptomoneda y no tendrá seguridad porque no habrá mineros. Lo que necesita una criptomoneda es una buena narrativa para que las personas puedan confiar y la utilicen porque piensen que va a ser valiosa en un futuro.

2.4.2 Ethereum y Smart Contracts

Muchas criptomonedas proponen agregar aplicaciones específicas. Se podría pensar, en lugar de lanzar un nuevo sistema para admitir las nuevas aplicaciones, crear una criptomoneda que pueda admitir cualquier aplicación en un futuro. Es decir que tenga lenguaje de programación completo de Turing. Con esto permitiría especificar cualquier aplicación que sea posible especificar por cualquier otro dispositivo.

Como se puede ver en el documento propuesto por Vitalik Buterin en 2013 [15] *Ethereum* es un sistema ambicioso de criptomonedas que provee un lenguaje de programación completo de Turing para escribir ”*scripts*” o “*contracts*”. Gas es el coste que tiene una operación en *Ethereum* y *Eher* es la unidad de gas para realizar transacciones donde la unidad mínima es 1 wei. Un *Ether* equivale a 10^{18} *wei.*

El modelo de programación de contratos inteligentes es el siguiente. El termino de contrato inteligente se utiliza para describir el uso de sistemas informáticos con el objetivo de hacer cumplir los contratos. En [2] se puede ver el siguiente ejemplo de contrato inteligente, una máquina expendedora sería como un contrato mecánico inteligente que impone un acuerdo entre el comprador y el propietario de la máquina.

En *Ethereum* un contrato es un programa que reside en la cadena de bloques. Cualquiera puede crear un programa en *Ethereum,* a cambio de una pequeña tarifa y cargando el código del programa en una transacción especial. Este contrato está escrito en *bytecode* y se ejecuta por en una máquina virtual especial de *Ethereum*, llamada *EVM*. Una vez cargado, el contrato reside en la cadena de bloques. Tendrá sus propios fondos y otros usuarios pueden hacer llamadas de procedimiento a través de la *API* que abra el programa y pueda enviar y recibir dinero.

Un ejemplo de contrato inteligente se muestra en la siguiente figura. Se implementa con el código de programación de alto nivel de *Ethereum* que es *Solidity*. Este contrato implementa

un registro de tipo nombre/valor o lo que es lo mismo un registro de nombres con valores, en el que cada nombre tiene asignado un único valor.

El contrato define una variable de datos que es *registryTable*, será una cadena de 32 bytes de asignación para las claves públicas. Inicialmente, se asigna a la cada cadena una dirección nula 0x0000000000 ... 000. Este contrato también define un único punto de entrada, que es la función *ClaimName*. Esta función acepta un solo argumento nombre de 32 bytes como entrada.

Lo primero que hace el contrato es asegurarse que el valor de entrada es mínimo de 10 *wei,* siendo *wei* la unidad monetaria más pequeña en *Ethereum*. Si no se cumple la condición, el mensaje finaliza con un error y no se toman medidas. Si cumple, y el nombre aún no se ha sido registrado, se le asigna permanentemente el valor de la dirección que llamó a esa función. En el **Código 1** se puede observar un contrato inteligente simple de *Ethereum* implementado con el lenguaje de programación *Solidity*.

```
contract NameRegistry {
      mapping(bytes32 => address) public registryTable;
      function claimName(bytes32 name) {
            if (msg.value < 10) {
                  throw;
            }
            if (registryTable[name] == 0) {
                  registryTable[name] = msg.sender;
            }
      }
}
```

Código 1. **Un contrato inteligente simple de Ethereum [2, Fig 10.4].**

Esto es lo que el contrato puede hacer en unas pocas líneas de código. Tal y como está hecho este contrato está pensado para cuando se envíe dinero se retire de la circulación para siempre. Sin embargo, se podrían añadir otro requerimiento que permitiera, por ejemplo, retirar el dinero. Para ello habría que implementar una segunda función para poder retirar el dinero. En esta función debería verificarse que la persona que quiere retirar el dinero sea la misma que lo ha depositado. Cualquiera puede llamar a un contrato de *Ethereum*, solo que las llamadas se firman para que se pueda identificar con seguridad quien está llamando.

Aunque en este ejemplo de código no aparezcan en *Ethereum* se pueden realizar bucles. Esto puede plantear un problema de que existan bucles infinitos. Con lo que es necesario un mecanismo que permita limitar los contratos que tardan mucho tiempo en ejecutarse. Este mecanismo se llama gas. Es decir, que la ejecución de cada instrucción en la máquina virtual

de *Ethereum* cuesta una cierta cantidad de gas. Diferentes operaciones cuestan diferente cantidad de dinero. Y una transacción tiene un coste fijo de gas también.

Se puede pensar en Ethereum como si se volar en una aerolínea en la que pagas por subir a bordo de un avión y se pagan extras por los demás servicios que adquieras. El gas se paga mediante la criptomoneda de *Ethereum* llamada *Ether*. En esencia, *Ethereum* proporciona un servicio en el que dos o más partes anónimas pueden acordar un contrato de forma segura.

Ethereum consta con diferencias respecto a *Bitcoin,* como, por ejemplo. el tiempo de creación de bloques es cada 12 segundos, también utiliza un protocolo alternativo para el consenso distribuido llamado GHOST entre otras. Esto hace que las aplicaciones que se pueden desarrollar sobre el sistema *Ethereum* sean varias, aunque las aplicaciones financieras pueden ser las más interesantes para implementar.

En resumen, *Ethereum* es ecosistema de criptomonedas complejo con características diferentes al protocolo *Bitcoin* que con la incorporación de los contratos inteligentes lo sitúan como un sistema prometedor.

2.5 Ventajas, desventajas y futuro de las criptomonedas

Como se comenta en [16] existen opiniones enfrentadas respecto al futuro de las criptomonedas y a *Bitcoin* en concreto. La visión de las personas que están a favor del uso de las criptomonedas está respaldada debido a que facilitan la transferencia de fondos entre dos partes sin intermediarios, constan de seguridad mediante claves públicas y privadas. Las tarifas por cada transacción son mínimas lo que permite a los usuarios evitar las tarifas elevadas que cobran la mayoría de los bancos. Además de que muchos países han empezado a regularizar *Bitcoin* como moneda válida. En especial, los países que quieres deshacerse del dinero efectivo tiene un enfoque a favor de las criptomonedas. Otro argumento de los optimistas del uso de las criptomonedas es la capitalización del mercado ya que se ha vuelto grande y poderoso, por lo que prohibirlo sería costoso para cualquier país.

Por otro lado, las personas en contra de las criptomonedas afirman que son muy volátiles y que pueden usarse para blanquear dinero o financiar actividades ilegales. Tymoigne (2015), por ejemplo, no está entusiasmado y comenta sus razones de que los *bitcoins* no sean una moneda fiable. Señala que muestran una alta volatilidad en los precios y que el valor en efectivo de un bitcoin es cero. Además, comenta que carece de un organismo central y que no la respalda ninguna entidad financiera.

2.5.1 Ventajas de las criptomonedas

En [17] Ivaschenko explica las ventajas y desventajas de *Bitcoin*.

1. **Código abierto para el minado de criptomonedas.** Por ejemplo, la información de las transacciones es pública, sin embargo, no hay información personal de emisor o receptor de la transacción.

2. No hay inflación. La cantidad máxima de monedas es de 21 millones de *bitcoins.* Ya que no depende de ninguna entidad central no hay fuerzas políticas ni corporaciones capaces de cambiar la cantidad máxima por lo tanto no hay posibilidad de que exista la inflación en el sistema.

3. Sistema *peer-to-peer*. La red no tiene un servidor central responsable de todas las operaciones. Todos los *wallets* forman parte de la red *Bitcoin*. Las transacciones se realizan mediante el consenso distribuido por miles de nodos. Ni bancos ni gobiernos pueden controlar las transacciones.

4. Posibilidad de transacciones ilimitadas. Cada persona titular de un *wallet* puede realizar una transacción de cualquier cantidad a cualquier otra dirección *Bitcoin* en cualquier lugar a través de la red. La transacción no se puede controlar ni evitar.

5. Sin límites. Las transacciones realizadas no se pueden cancelar. Los *bitcoins* no se pueden falsificar o copiar. Estas características garantizan la seguridad e integridad del sistema.

6. Costes mínimos por transacción. No se pagan comisiones a bancos ni organizaciones. El coste de la transacción es más reducido que cualquier otro. Equivale al 0.1% del monto de la transacción. Los costes por transacción se cargan al *wallet* de los mineros en *bitcoins*.

7. Descentralización. No existe una autoridad central de la red, la red se distribuye entre todos los participantes. Con lo que una autoridad central no puede modificar las reglas del sistema. Y aunque una parte de la red se desconecte, el sistema sigue funcionando de manera estable.

8. Sencillo de usar. Muchas veces abrir una cuenta en una entidad financiera es tedioso mientras que abrir un *wallet* en *Bitcoin* se realiza en unos minutos sin comisiones.

9. Anónimo. El sistema es anónimo y transparente. Cualquier persona o empresa puede realizar un número ilimitado de transacciones sin proporcionar más información que la dirección de *Bitcoin*.

10. Transparencia. El sistema almacena el historial de todas las transacciones que han ocurrido en *Bockchain*. Para un anonimato mayor se utiliza una dirección *Bitcoin* por cada transacción.

11. Velocidad de transacción alta. La capacidad de enviar dinero a cualquier parte y a cualquier persona se realiza en cuestión de minutos después que el sistema procese la transacción.

12. El valor es únicamente propiedad del propietario. La verificación del uso correcto del *wallet* es responsabilidad del propietario. El propietario tiene una clave privada y una clave pública que es la dirección *Bitcoin.* Nadie más puede retirar los *bitcoins* [18].

13. No se puede cometer fraude. Actualmente la mayoría de compras por internet se realizan con las tarjetas de crédito. Los clientes ingresas los datos de sus tarjetas en diferentes páginas web inseguras que a menudo son robadas. Las transacciones de *bitcoins* utilizan dos claves, la pública y la privada. La transacción debe estar firmada por las claves privadas para que se pueda comprobar quien realizó la transacción.

14. La posibilidad de invertir fondos es un recurso transparente y rentable.

2.5.2 Desventajas de las criptomonedas

1. Alta volatilidad. El valor de los *bitcoins* tiene muchos altibajos dependen directamente de las declaraciones que anuncian los gobiernos de diferentes países. Esta gran fluctuación del precio crea problemas a corto plazo.

2. Grandes riesgos de invertir en criptomonedas que deberían considerarse en el medio y largo plazo.

En [16] opinan que la lista de desventajas es mucho más larga y que está relacionado con el blanqueo de dinero, el terrorismo y otras actividades ilegales. Sin embargo, aunque es muy difícil de predecir muchos académicos y profesionales de este tema afirman que el futuro de las criptomonedas es brillante ya que eliminaría las barreras comerciales y los intermediarios, disminuyendo el coste de las transacciones. Esto provocaría un aumento del comercio y de la economía. Por otra parte, se debe considerar el alto riesgo existente debido a la volatilidad y la falta de un respaldo de una entidad central [19].

2.5.3 Futuro de las criptomonedas

Como conclusiones de las ventajas y desventajas en [16] opinan que *Bitcoin* y las otras criptomonedas tienen el potencial de reemplazar a los métodos de pago tradicionales. Pero para lograr esto deben solventar una serie de desafíos como los problemas reglamentarios. Es improbable que suceda en un corto periodo de tiempo, aunque los bancos deberían analizar

detenidamente la tecnología de las criptomonedas como una nueva forma de transferir la propiedad del valor a largo plazo.

En [2] opinan que la tecnología subyacente de *Bitcoin* es profunda e interesante y se basa en principios sólidos. El futuro de *Bitcoin* y de las criptomonedas es un tema incierto, sin embargo, hay entusiastas que contemplan una revolución tecnológica. Es una posibilidad que la descentralización como sistema monetario puede hacer reconsiderar otras instituciones centralizadas como son las instituciones centralizadas, las acciones, los votos, los inmuebles, la propiedad intelectual entre otras.

Otra posibilidad es la interconexión entre diferentes ecosistemas de criptomonedas. Es decir, que desde las transacciones de una cadena de bloques se puedan referir a ciertas otras transacciones de otra cadena de bloques distinta. Esto provocaría aplicaciones nuevas.

Los optimistas afirman que la tecnología de las criptomonedas tiene muchas posibilidades comerciales y como por ejemplo el sistema financiero, la economía e incluso la política en todo el mundo mientras que los pesimistas afirman que en algún momento estallará y sufrirá un colapso inevitable y espectacular.

2.6 Conclusiones del análisis del contexto e identificación del problema

Actualmente la tecnología está omnipresente en la sociedad. Este hecho modifica la conducta y las rutinas humanas. Las personas tienen objetivos en la vida los cuales se obtienen mediante un intercambio de valor. Para lograr estos objetivos las personas necesitan atesorar una cantidad de valor concreto para poder intercambiarlo y conseguir el objetivo deseado.

Es ese aliciente, el de intentar conseguir la mayor cantidad de valor, el motivo del que las personas inviertan en mercados como el de las criptomonedas. Estos mercados son volátiles y de riesgo, pero si se dan las condiciones necesarias es posible conseguir una rentabilidad muy alta.

La idea es invertir en una o varias criptomonedas en un momento determinado con la esperanza de que en un futuro el valor aumente obteniendo un mayor rendimiento. Esta estrategia se puede hacer a corto, mediano o largo plazo. Por contrapartida, es un mercado muy volátil y se debería invertir el dinero que se esté dispuesto a perder. Aun conociendo las

desventajas las personas, inversores, empresas y compañías están invirtiendo en criptomonedas debido a sus ventajas.

La problemática consiste en la estimación del precio para averiguar el mejor momento de invertir con el objetivo de obtener una mayor rentabilidad. Por lo que el primer paso para solventar esta problemática es identificar los factores que influyen en el precio de las criptomonedas. En el siguiente capítulo se observan diferentes trabajos realizados sobre este tema.

3 Estado del arte

En este capítulo se van a analizar los diversos artículos y las diferentes perspectivas adoptadas por sus autores con la motivación de identificar qué factores influyen en el precio de las criptomonedas aplicando distintas metodologías, técnicas y algoritmos. La mayoría de estudios se centran en el precio de *Bitcoin*.

3.1 Técnicas de descomposición de señales

Se comienza con dos artículos que emplean un planteamiento novedoso ya que utilizan técnicas de procesamiento de señal para intentar descubrir las posibles características ocultas que influyen en los datos del precio de *Bitcoin*.

En [20] "*What drives Bitcoin Price?"* se emplea la técnica llamada *Empirical Mode Decomposition (EMD)* utilizada para descomponer las señales no estacionarias y con ruido. Concretamente, se descompone el precio del *Bitcoin* en funciones *Intrinsic Mode Function (IMF)*. Estas funciones se clasifican en diferentes escalas de tiempo y corresponden a frecuencias altas (a corto plazo), medias (a medio plazo) y bajas (a largo plazo). Se observan peculiaridades en las funciones posiblemente debido a factores ocultos que afectan el precio del *Bitcoin*. para cada función *IFM* se realiza una correlación de Pearson, Kendall obteniendo como resultado que los factores más influyentes en el precio del *Bitcoin* parecen ser los componentes a largo plazo.

En [21] *"What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis"* se adopta la técnica Wavelet Coherence Analysis. Con esta técnica se pretende descubrir la evolución de las relaciones en el tiempo entre los diferentes factores y también conexiones a corto y largo plazo. El objetivo es determinar los posibles factores que influyen en el precio del *Bitcoin* son factores fundamentales, especulativos, técnicos y sí existe la influencia del mercado chino.

En [20], [21] y [22] coinciden en que *Bitcoin* posee características de un activo especulativo, sin embargo, es probable que los factores de largo plazo sean los más influyentes en el precio de *Bitcoin*. Componentes fundamentales como el suministro total de *bitcoins*, el precio de *Bitcoin* y el uso del sistema en el comercio. Además en [21] se observa que el precio del *bitcoins* impulsa el interés de los inversores, que la dificultad de la red es demasiado alta como para que un usuario normal obtenga beneficios de la minería, que parece ser una inversión no segura y que no se han encontrado evidencias significativas de que el mercado chino influya

en el americano. Finalmente, se comenta que *Bitcoin* es un activo único debido a que posee propiedades de un activo financiero estándar y uno especulativo.

3.2 Técnicas de Inteligencia Artificial

El siguiente artículo [23] *"Cryptocurrencies Prices Forecasting With Anaconda Tool Using Machine Learning Techniques"* tiene como objetivo identificar los criterios afectan al precio de las criptomonedas y su uso para la previsión de los precios. Se utilizan las siguientes técnicas de inteligencia artificial, concretamente de aprendizaje automático. Y los algoritmos utilizados son una regresión lineal múltiple con la configuración sklearn, *random forest* con 100 árboles, una red neuronal *Long short-term memory* (*LSTM*) con 50 neuronas de capa oculta y 100 épocas de entrenamiento.

La regresión lineal muestra la relación entre las variables y como se afectan entre ellas, el algoritmo *random forest* utiliza el método *bagging (Bootstrap Aggregating)* para crear grupos de árboles de decisión con datos aleatorios y la red neuronal *LSTM* se utiliza para aprender conocimiento a largo plazo.

Los criterios que se definen como conjunto de datos a estudiar son el número total de *bitcoins* minados, el nivel de dificultad, el volumen de transacciones, la percepción de la sociedad del valor de las criptomonedas y el precio del *Bitcoin*. El 70% de los datos se utilizan para aprendizaje y el 30% restante para test.

Como resultado del experimento se encuentra una selección de criterios que pueden explicar más del 70% de la variación de los precios de las criptomonedas utilizando una combinación de *Multiple Linear Regression*, *Random Forests*, y algoritmos *LSTM* implementados con *Python* en la herramienta *Anaconda*.

En [24] *"Price Fluctuations and the Use of Bitcoin: An Empirical Inquiry"* el objetivo es identificar las características y la información de precios, la inversión y el uso comercial de *Bitcoin.* Se emplean las técnicas de regresión ordinarias y de Tobit. Como resultado se obtiene la popularidad, los sentimientos de los informes periodísticos y el número total de transacciones afecta al precio de *Bitcoin*. Además, este documento es el primero en realizar una encuesta global a comerciantes que han adoptado esta tecnología y se muestra datos como las características de la compañía, el uso de otros métodos de pago, el conocimiento de los clientes sobre *Bitcoin*, etc…

En [25] *"Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis"* el objetivo es comprobar si los datos de Twitter están relacionados con las criptomonedas y si pueden ser utilizados para desarrollar una estrategia comercial de cirptomonedas. Se utilizan técnicas de aprendizaje automático supervisado y se aplica clasificación de texto y análisis de sentimiento en Twitter sobre *Bitcoin*.

El enfoque es aplicar algoritmos de aprendizaje supervisado como la regresión logística, Naive Bayes y *Support Vector Machine* (*SVM*). El objetivo de cada algoritmo es predecir si el precio de *Bitcoin* aumentará o disminuirá en un período de tiempo establecido analizando los Tweets. Se utiliza un riguroso análisis de errores para garantizar que se utilicen entradas precisas en cada paso del modelo. Se obtiene una exactitud de predicción final de hora a hora y día que excede el 90%.

Primero se utiliza el clasificador Naive Bayes obtuvo el mejor resultado de todos los algoritmos de clasificación de texto al lograr una exactitud diaria de 95% y una precisión de hora a hora de 76.23%. Después, se utiliza la API de textprocessing.com para calcular la negatividad, neutralidad y positividad para cada tweet. Utilizando las etiquetas positivas, neutrales y negativas como vectores de características para Naive Bayes (Bernoulli y Multinomial) se obtiene una mejoría con el 86.00% de exactitud diario y una precisión de hora a hora de 98.58%. Se obtiene un conjunto de 8061 palabras de disminución de precio y 8852 de aumento de precio.

3.3 Técnicas de regresión

Los dos siguientes artículos muestran un enfoque aplicando regresión lineal. En [22] *"The economics of BitCoin price formation"* se utiliza la técnica basada en *Vector autoregression (VAR)* para identificar los efectos causales entre el precio de *Bitcoin* y sus factores. Este artículo no está de acuerdo con estudio anteriores ya que afirman que existen deficiencias porque el análisis de los factores que influyen en el precio del *Bitcoin* se realizaba por separado y no se consideraban las interacciones entre ellos.

Se estiman cuatro conjuntos de modelos econométricos identificados en estudios anteriores para explicar el precio del *Bitcoin*. El primer modelo sobre la interacción entre la oferta y la demanda de *Bitcoin*, el segundo sobre el atractivo para los inversores para realizar transacciones con *Bitcoins*, en el tercero el impacto de los desarrollos macroeconómicos y financieros mundiales. Y en el último la interacción entre los tres modelos anteriores.

Están en concordancia [20], [21] y [22] ya que confirman que los propios factores fundamentales del mercado de *Bitcoin* tienen un impacto importante sobre el precio. Esto implica que el precio del *Bitcoin* podría explicarse con un modelo económico estándar.

En [22] se han encontrado evidencias significativas de que las opiniones de Wikipedia influyen sobre el precio de *Bitcoin,* que la nueva información optimista afecta al precio del *Bitcoin* de manera positiva. Finalmente, en contra de lo comentado en estudios previos, se afirma que el desarrollo macro-financiero global no impulsa el precio de *Bitcoin*.

El objetivo de [26] *"Forecasting Cryptocurrencies Financial Time Series"* es analizar la predicción de las criptomonedas comparando varios modelos predictivos univariante y multivariante. La técnica aplicada es un conjunto de predictores criptográficos basados en *Dynamic Model Averaging* (*DMA*) para combinar *Dynamic Linear Models* (*DLM*) y otros modelos *Vector Autoregressive* (*VAR*). Las criptomonedas analizadas son *Bitcoin*, *Ethereum*, *Ripple* y *Litecoin*. Se obtiene resultados significativos con el estimador puntual de *Bitcoin* y *Ethereum* cuando se utilizan combinaciones de modelos univariantes y con el estimador de densidad para todas las criptomonedas cuando se basan en modelos multivariantes en un cierto tiempo.

En el siguiente documento [27] *"On the return-volatility relationship in the Bitcoin market around the price crash of 2013"* se realiza un análisis antes del desplome de precios de *Bitcoin* que hubo en diciembre del 2013. En él se analiza la relación entre el rendimiento de los precios y la volatilidad en el mercado. La técnica adoptada se basa en un marco *Generalized Autorregresive Conditional Heterocedasticity* (*GARCH*) asimétrico y se obtiene como resultado que, en el periodo anterior al colapso, *Bitcoin* poseía la propiedad de refugio seguro. Sin embargo, después esta propiedad desaparece.

El método empleado en [28] *"What Factors Give Cryptocurrencies Their Value: An Empirical Analysis"* es un modelo de regresión múltiple de mínimos cuadrados y se utilizan los datos de las 66 criptomonedas más utilizadas del mercado para identificar qué afecta a su valor. Las variables que se utilizan son: número total de *bitcoins*, cantidad de incentivos al generar un bloque, tiempo de generación de un bloque, dificultad, algoritmo utilizado y precio.

Como resultado se obtienen dos modelos y se encuentra que aproximadamente el 84 % del valor relativo puede explicarse por tres variables: la potencia de cálculo, el número total de criptomonedas y el algoritmo utilizado para generar los bloques. Además, se afirma que las tasas de creación de unidades influyen en el precio y que existen otros factores subjetivos para determinar el precio de mercado que aún no se han identificado y que es probable que haya un factor especulativo, así como la tendencia a acumular monedas minadas que desempeñará un papel adicional en la formación del precio, pero que es más difícil de cuantificar y medir.

En este documento [29] *"Do Cryptocurrencies and Traditional Asset Classes Influence Each Other?"* se estudia las conexiones e influencias entre las criptomonedas y los activos tradicionales como productos básicos, divisas, acciones y valores financieros. Se evalúa las relaciones y las asimetrías entre estos activos. La metodología propuesta permite medir las asimetrías de volatilidad. Se obtiene como resultado que *Bitcoin* no es relevante para otros mercados. Esta conclusión [29] está en línea con [22]. Sin embargo, se encuentra evidencias de que el mercado de *Bitcoin* puede influir en otros activos. Por ejemplo, los resultados muestran una caída del oro después de una caída de precio del *Bitcoin*. Por tanto, el mercado *Bitcoin* podría ser más relevante para otros mercados financieros de lo que se creía anteriormente.

Por último, en [30] *"The Hidden Predictive Power of Cryptocurrencies: Evidence from US Stock Market"* se analiza la relación entre la incertidumbre en los precios de las acciones y los precios de las criptomonedas. Se plantea la hipótesis de que para pronosticar el precio de las acciones un modelo predictivo basado el estimador de Westerlund Narayan tiene el potencial de ser más preciso para pronosticar los precios de las acciones que el modelo tradicional *Ordinary Least Squares* (*OLS*). Se exploran las criptomonedas *Bitcoin*, *Ethereum*, *Litecoin* y *Ripple*.

Se obtienen evidencias de que el modelo predictivo utilizando el estimador de Westerlund Narayan es más preciso para previsión de los precios de las acciones en los Estados Unidos que el modelo tradicional basado en *Autoregressive Integrated Moving Average* (*ARIMA)* y *Autoregressive Fractionally Integrated Moving Average* (*ARFIMA*).

3.4 Teoría neutralista de la evolución molecular

En el artículo [31] *"Evolutionary dynamics of the cryptocurrency market"* se utiliza un enfoque basado en la teoría neutralista de la evolución molecular que no captura toda la complejidad de *Bitcoin*, sin embargo, se observa la buena coincidencia de la imagen que surge de los datos sugiere que algunas propiedades a largo plazo del mercado de criptomonedas se pueden justificar con hipótesis simples. Se realiza un análisis exhaustivo de la dinámica del mercado de criptomonedas analizando 1469 criptomonedas.

Se ha identificado que la capitalización de mercado total de las criptomonedas se encuentra en aumento y muestra un crecimiento exponencial, que la cuota de mercado de *Bitcoin* está disminuyendo y que la de las demás criptomonedas está aumentando. Además, el número de criptomonedas que surgen y que desaparecen se mantiene estable y es aproximadamente unas siete a la semana. Se observa los cambios en la legislación, técnicos y sociales probablemente impacten directamente sobre el mercado de las criptomonedas. Por ejemplo,

en abril del 2017, en Japón se empezó a tratar *Bitcoin* en dólares estadunidenses, mientras que en febrero de 2017 un cambio de regulación en China desplomó el precio aproximadamente en 86 €. De forma similar, el aumento de la capitalización del mercado probablemente atraerá una mayor especulación.

3.5 Conclusiones del estado del arte

Se han visto diferentes perspectivas y se han aplicado diferentes enfoques para intentar determinar factores que influyen en el precio de las criptomonedas y si existen relaciones entre esos factores. Las principales técnicas adoptadas han sido novedosas como el procesamiento de señal empleado en [20] aplicando *Empirical Mode Decomposition (EMD)* o el *Wavelet Coherence Analysis* aplicado en [21]. Otros enfoques basados en técnicas de regresión como por ejemplo los adoptados en [22] y en [26] con *Vector autoregression (VAR)*, una regresión de Tovit en [24], una regresión múltiple de mínimos cuadrados [28] y [30]. Finalmente se ha empleado técnicas de inteligencia artificial en [23] y en [25], entre otras.

A continuación, se recopilan los resultados y las conclusiones de los artículos explicados anteriormente.

- Se afirma en [21] que *Bitcoin* es un activo único debido a que posee características propias de un activo financiero estándar y de uno especulativo. Además, se observa que el precio del *bitcoins* impulsa el interés de los inversores, que posiblemente sea una inversión no segura.
- En [20], [21], [22] y [28] coinciden en que el precio de *Bitcoin* posee características de un activo especulativo, sin embargo, es probable que los factores de largo plazo sean los más influyentes en el precio de *Bitcoin* y que se podría explicar con un modelo económico estándar.
- Se muestra la evolución de la capitalización del mercado en [31], un aumento con un crecimiento exponencial, que la cuota de mercado de *Bitcoin* está disminuyendo y la de las demás criptomonedas esta aumentado. Además, que en un futuro inmediato y de medio plazo, los cambios en la legislación, técnicos y sociales probablemente impacte sobre el mercado de las criptomonedas.
- Las características propias de *Bitcoin* influyen en su precio como se puede ver en [28] que la tasa de creación de moneda influye en el precio y que existen factores subjetivos que aún no han sido identificados y que es probable que haya un factor especulativo, así como la tendencia a acumular monedas minadas que desempeñará un papel adicional en la formación del precio, pero que es difícil de cuantificar y medir.

- En el análisis realizado en [27] se identifica que, antes del periodo correspondiente al colapso que hubo en el precio de *Bitcoin* en 2013 tenía la propiedad de refugio seguro. Sin embargo, después de ese momento esa propiedad desaparece.
- En [22] y [24] coinciden en que la información publicada afecta al precio de *Bitcoin*. En [24] se muestra que la popularidad, los sentimientos de las noticias periodísticas y el número total de transacciones afectan al precio de *Bitcoin*. En [22] se muestra que la información optimista afecta al precio del *Bitcoin* de manera positiva y que las opiniones de Wikipedia sobre el precio de *Bitcoin* podría ser una evidencia del comportamiento especulativo a corto plazo de los inversores. Además en [22] se encuentra que el desarrollo macro-financiero global no afecta al precio de *Bitcoin*. Por otra parte, en [29] afirman que *Bitcoin* no es relevante para otros mercados pero que hay evidencias de que puede influir en otros activos como el oro.
- Se obtiene en [23] una selección de criterios que pueden explicar más del 70% de la variación de los precios de las criptomonedas utilizando una combinación de *Multiple Linear Regression*, *Random Forests*, y algoritmos *LSTM* implementados con *Python* en la herramienta *Anaconda*. Y en [25] se ha identificado que hay un conjunto de palabras corresponden a la disminución de precio y otro conjunto al aumento. Por último, en [30] se obtiene que hay evidencia de que el modelo predictivo basado en criptomonedas es más preciso para previsión del comportamiento de los precios de las acciones en los Estados Unidos.

Es un hecho, cuanto mejor se logre conocer qué factores influyen en el precio y qué relaciones existen entre ellos, será posible determinar con mayor exactitud la predicción del precio de las criptomonedas. Es esta la razón, por la que los artículos explicados anteriormente adoptan diferentes metodologías, técnicas y algoritmos.

Hay que ser consciente de la dificultad del problema que se va a tratar tal y como se explica en [28] existen factores subjetivos que aún no han sido identificados y que probablemente sean difíciles de cuantificar y medir. Se conoce que es un problema complejo y para solventarlo se van a adoptar el un enfoque basado en las ideas de los artículos [23] y en [25]. En ambos artículos utilizan técnicas de inteligencia artificial, concretamente en aprendizaje automático para lograr descubrir características ocultas en los datos.

4 Objetivos y metodología de trabajo

A continuación, se define el objetivo general, los objetivos específicos y la metodología que se va a seguir para dar solución al problema identificado.

4.1 Objetivo general

Este trabajo tiene como objetivo **establecer una comparativa entre dos técnicas de predicción de la evolución del precio de las criptomonedas: la regresión lineal múltiple y la red neuronal**.

4.2 Objetivos específicos

Es necesario disgregar el objetivo general en otros más específicos para planificar y acotar el trabajo, y finalmente comprobar si se han alcanzado. Los objetivos específicos son los siguientes:

1. **Buscar y seleccionar las tecnologías** oportunas para implementar funciones, métodos y algoritmos que permitan realizar el análisis de datos que se propone.
2. **Analizar los datos de diferentes fuentes e identificar los datos útiles**. Este es un objetivo prioritario ya que para realizar un correcto análisis se necesitan datos útiles y disponibles. Este objetivo se subdivide en otros más concretos que son:
    - Buscar y conocer las diferentes fuentes de datos que existen para poder capturar los datos históricos.
    - Entender la manera de recuperar los datos y definir funciones para la correcta obtención de los mismos.
    - Obtener indicadores de las características principales del mercado de *Bitcoin.*
    - Seleccionar la información importante disponible para realizar un correcto análisis.
    - Almacenar los datos.
3. **Realizar un análisis y visualización de los indicadores *Bitcoin*.** Este objetivo se desglosa en:
    - Analizar y procesar los datos para poder aplicar los métodos estadísticos.
    - Visualizar y analizar la evolución del precio del *Bitcoin* y de las principales criptomonedas desde que aparecieron. En el caso de *Bitcoin*, los últimos 9 años.
    - Visualizar y analizar otros indicadores relevantes de *Bitcoin*.

- Realizar una correlación entre *Bitcoin* y las criptomonedas con mayor capitalización del mercado para identificar si existen relaciones entre ellas y si el precio de una criptomoneda influye en el precio de las otras.

4. **Implementar dos modelos predictivos empleando técnicas de inteligencia artificial.**
   - **Implementar una regresión lineal multiple** concretamente *Ordinary Least Squares (OLS).*
   - **Implementar tres redes neuronales,** la primera de tipo *Long Short Term Memory* (*LSTM*) de una capa, la segunda LSTM2 con dos capas *LSTM* conectadas de forma secuencial y la tercera de tipo *Gate Recurrent Unit (GRU)*.
5. **Evaluar y comparar los resultados obtenidos** de los modelos predictivos mediante el *Mean Absolute Error (MAE)*.
6. **Establecer futuras líneas de trabajo**.

4.3 Metodología del trabajo

Se va a describir cada una de las fases de la investigación, como una secuencia de pasos permitiendo alcanzar los objetivos de la investigación.

A continuación, se enumeran los diferentes pasos de la metodología explicada.

1. **Búsqueda y selección de tecnologías.** Se realiza una búsqueda de las tecnologías empleadas actualmente para este tipo de análisis de datos y se escogen las que posiblemente sean las más oportunas para solventar el problema.
2. **Identificación de las fuentes para obtener datos.** Se identifican las diferentes fuentes de datos disponibles.
3. **Realización de pruebas para obtener los datos.** Se realizan pruebas para la obtención de los datos de diferentes fuentes.
4. **Almacenamiento de los datos.** Se almacenan los datos para poder realizar el procesado.
5. **Selección de los datos relevantes.** Se realiza una selección de los datos importantes.
6. **Análisis previo y visualización de los datos.** Se analizan los datos disponibles de *Bitcoin* y de las criptomonedas con mayor capitalización del mercado. Se transforman los datos y se aplica un análisis estadístico.
7. **Selección de algoritmos de inteligencia artificial.** Se selecciona los algoritmos de inteligencia artificial que se van a utilizar.
8. **Implementación de los modelos predictivos.** Se implementan los modelos predictivos.

9. **Evaluación y comparación de los resultados obtenidos.** Se ejecutan los modelos predictivos y se comparan los resultados obtenidos.
10. **Conclusiones y posibles mejoras.** Se obtienen las conclusiones y se describen posibles líneas de trabajo futuras.

5 Desarrollo específico de la contribución

En este apartado se explica el desarrollo necesario para conseguir los objetivos propuestos en el capítulo anterior. Se elige la tecnología para realizar el análisis, la obtención, el tratamiento, la visualización de los datos y la implementación de los modelos predictivos. Se establece la comparativa entre las dos técnicas de inteligencia artificial para la predicción de la evolución del precio de las criptomonedas: la regresión lineal múltiple y la red neuronal. Los datos de entrada se pueden ver en **Tabla *8*** y son los mismos para ambos modelos.

El modelo predictivo de regresión lineal múltiple se emplea el estimador *Ordinary Lest Square (OLS).* Este método de mínimos cuadrados permite conocer la ecuación de regresión y sus componentes de manera que minimiza la suma de las diferencias cuadráticas de los valores observados y de los valores calculados o predichos.

El modelo de red neuronal es modelo computacional que trata de imitar el funcionamiento del cerebro humano aprendiendo a partir de la experiencia. Es una técnica de aprendizaje supervisado dentro del campo de la inteligencia artificial. Se implementan tres modelos de red neuronal. La primera de tipo *Long Short Term Memory* (*LSTM*) de una capa, la segunda *LSTM2* con dos capas *LSTM* conectadas de forma secuencial y la tercera de tipo *Gate Recurrent Unit (GRU)*.

Por último, se comparan los resultados obtenidos de cada modelo predictivo empleando el error *Mean Absolute Error (MAE).*

A continuación, se describen los contenidos de cada una de las secciones.

- **5.1 Tecnología**. Se describen las tecnologías utilizadas. El lenguaje de programación *Pyhton* sobre la plataforma de ciencia de datos, *Anaconda*.
- **5.2 Obtención de datos**. Se enumeran las diferentes APIs, se seleccionan *Quandl* y *Poloniex* como fuente de datos y se explica el modo de obtener los datos.
- **5.3 Análisis previo, tratamiento y visualización de los datos**. Se realiza un análisis del precio medio de *Bitcoin*, un análisis del precio medio de las principales criptomonedas, una correlación del precio medio de *Bitcoin* y las principales criptomonedas y una visualización de los indicadores de *Bitcoin*.
- **5.4 Desarrollo de la comparativa**. Se describen la implementación del modelo predictivo de regresión múltiple y el modelo predictivo de la red neuronal.
- **5.5 Resultados**. Se comentan los resultados obtenidos.

5.1 Tecnología

Para el desarrollo de este estudio se utiliza el lenguaje de programación *Python* [32] sobre la plataforma *Anaconda* [33]. Se ha escogido este lenguaje de programación debido a que permiten realizar análisis estadístico, transformaciones, realizar peticiones *APIs*, almacenar los datos, implementar modelos predictivos, crear visualizaciones de los datos de forma interactiva y definir funciones propias, en definitiva, resulta un lenguaje potente y versátil para este tipo de problemas. Se ha escogido la plataforma *Anaconda* debido a que permite simplificar la gestión de los entornos y de las librerías de programación.

5.1.1 Pyhton

Es un lenguaje de programación interpretado de alto nivel que fue creado en 1990 por Guido van Rossum [34]. La filosofía de su creador ha sido transmitida al lenguaje de programación como se puede ver en el siguiente párrafo [35].

"To be effective, an idea must be expressed as a computer program, using a programming language. The language that is best to express an idea will give the team using that language a key advantage, because it gives the team members — people! — clarity about that idea."

Se trata de un lenguaje multiplataforma, orientado a objetos y con tipo de datos dinámico. Soporta múltiples paradigmas de programación, imperativo, funcional, procedural, posee una autogestión de la memoria y tiene una colección de librerías estándar muy extensa. Es de propósito general, lo que significa que está diseñado para ser aplicado en varias áreas. Python tiene una licencia de código abierto y esta administrado por la organización *Python Software Fundation.*

Es por su diseño, su filosofía y sus características las razones por las cuales hoy en día es un lenguaje tan popular y utilizado. La versión utilizada en este proyecto es *Python* es la 3.6.4. y las librerías utilizadas son demasiadas para enumerarlas pero se pueden ver en los diferentes archivos *.ipynb en [36].

5.1.2 Anaconda

La plataforma *Anaconda* se trata de una distribución de código libre que permite programar con los lenguajes de programación *Python* y *R* [37]. El software fue desarrollado en 2012 por la empresa *Continuum Analytics*, actualmente llamada *Anaconda Inc*.

Es una plataforma para realizar aplicaciones relacionadas con *Data Science* aplicando *Artificial Intelligence* (*IA*) y *Machine Learning (ML)* como por ejemplo procesamiento de datos *Big Data, Predictive Analytics o Deep Learning.* Permite simplificar los entornos, administrar e implementar las versiones de los paquetes y de las librerías de programación.

Es recomendable trabajar con entornos virtuales ya que, si se trabaja en múltiples proyectos de Python en la misma máquina, resulta muy útil para mantener separadas las librerías, paquetes y módulos de los diferentes proyectos. *Anaconda* permite crear diferentes entornos virtuales para almacenar todo el *software* referente a un proyecto para mantenerlo organizado y separado de otros proyectos. La versión utilizada en este trabajo es *Anaconda3* que incluye una gran variedad de paquetes y librerías instaladas por defecto.

En la **Figura *14*** se puede ver el esquema general de su arquitectura. La base es *conda*, el paquete de ciencia de datos, gestor de entornos virtuales y que permite descargar los módulos y paquetes necesarios. Sobre *conda* se pueden observar las librerías de ciencia de datos que pueden ser *Integrated Development Environment (IDEs)* como *Jupiter*, librerías científicas y analíticas como *numpy* o *pandas,* librerías de visualización como *plotly* y librerías de aprendizaje automático como *tensorflow* o *keras*. *Conda* y las librerías de ciencia de datos se encapsulan en *Anaconda Project* y se accede mediante *Anaconda Navigator*, una interfaz gráfica para instalar y editar diferentes aplicaciones, gestionar entornos virtuales y acceder a los recursos de la comunidad.

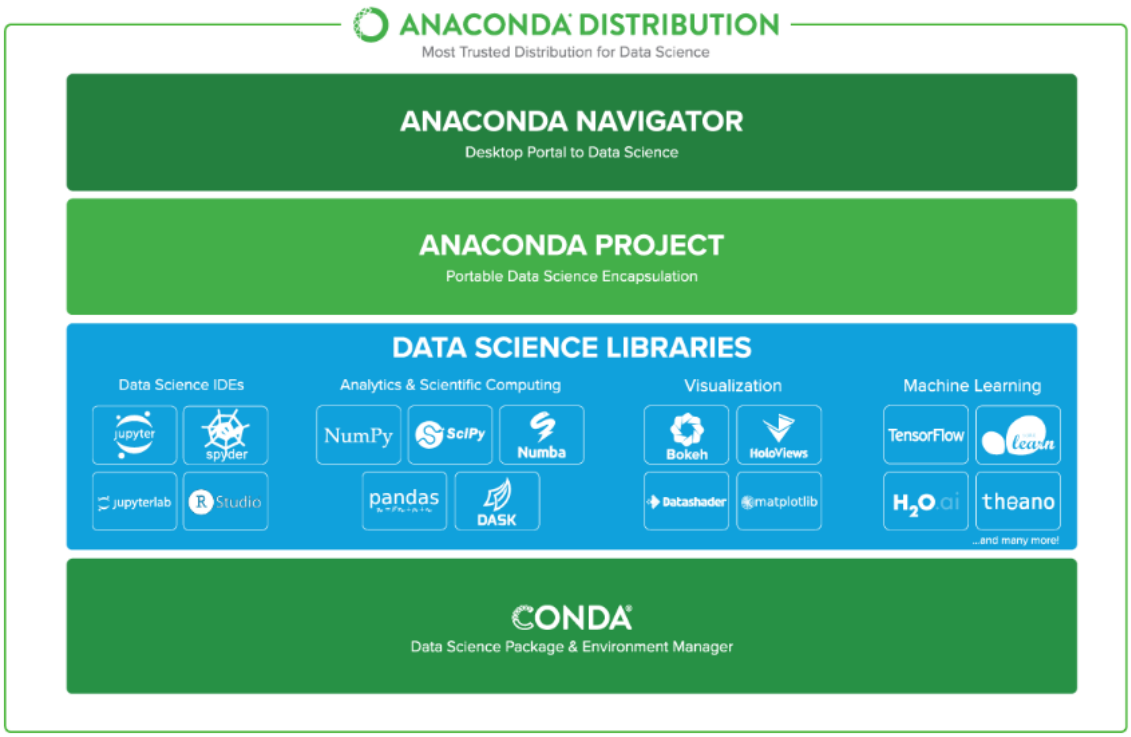

Figura 14. Esquema general de la arquitectura de *Anaconda Distribution* [33].

Actualmente se ha convertido en un estándar para desarrollar aplicaciones relacionadas con *Data Science*, ya que actualmente cuenta con más de 6 millones de usuarios, tiene más de 1.000 paquetes de código abierto, existe mucha documentación y cuenta con el soporte de la comunidad.

5.2 Obtención de datos

Una vez escogida la tecnología que se va a utilizar, es necesario obtener datos útiles para poder trabajar. Lo recomendable es capturar datos de diferentes fuentes, disponibles y comprobar que la información sea útil para el análisis que se quiere realizar. En este caso, los datos se capturan mediante peticiones que se realizan a las *Applications Programming Interface (APIs)* de los servicios de intercambio de criptomonedas.

Se han identificado diferentes servicios de intercambio que proporcionan información de los indicadores, de las transacciones, del mercado, de los intercambios y datos históricos de las criptomonedas. A continuación se describe brevemente distintos servicio de intercambio.

Bitpay es un proveedor de servicios de pago de *Bitcoin* y ofrece servicios de procesamiento de pago de *Bitcoin* y *Bitcoin Cash*.

Blockchain es un servicio para explorar los bloques de *Bitcoin*. Proporciona gráficos, estadísticas e información detallada de los datos de *Bitcoin*.

CoinAPI es un servicio que proporciona una *API* para acceder a los datos históricos y en tiempo real del mercado de criptomonedas.

Coinbase puede generar direcciones, realizar transacciones y devolver datos históricos de *Bitcoin*, *Bitcoin Cash*, *Litecoin* y *Ethereum*.

Coinmarketcap es un servicio para ver las posiciones de las criptomonedas en base a su capitalización de mercado.

Coingy permite interactuar con cuentas *coingy* y cuentas de intercambio para actualizar saldos, programar y cancelar pedidas, establecer y cancelar alertas.

Cryptocompare permite obtener datos de criptomonedas, datos históricos de volumen y de explorador de bloques.

Poloniex es una plataforma de intercambio de criptomonedas que proporciona una API pública para recuperar información de diferentes criptomonedas.

Quandl es una plataforma de datos financieros, económicos y alternativos que sirve para los profesionales de la inversión. El servicio API es accesible a través de librerías de R, Python, Matlab, Maple y Stata.

Kraken es una plataforma de intercambio que proporciona un servicio

En la

Tabla *2* se muestra una lista de los diferentes servicios de intercambio que proporcionan *APIs.*

Nombre del servicio	URL del servicio
Bitpay	https://bitpay.com/api
Blockchain	https://www.blockchain.com/api/
CoinAPI	https://www.coinapi.io/
Coinbase	https://developers.coinbase.com/
Coinmarketcap	https://coinmarketcap.com/es/api/
Coinigy	https://www.coinigy.com/bitcoin-api/
Cryptocompare	https://www.cryptocompare.com/api/#-api-data-toppairs-
Poloniex	https://poloniex.com/support/api/
Quandl	https://blog.quandl.com/api-for-bitcoin-data
Kraken	https://www.kraken.com/help/api

Tabla 2. Lista de los servicios *APIs* de criptomonedas.

Todos los servicios enumerados proporcionan *APIs* donde cada uno tiene diferentes características. Por ejemplo, *kraken* y *coinbase* proporcionan *APIs* para que desarrolladores puedan implementar aplicaciones de intercambio, otros servicios como *Bitpay* que son de pago, otros como *Cryptocompare* donde el usuario gratuito tiene límites de peticiones muy bajos.

Se han escogido como fuente de obtención de datos los servicios de intercambio de ***Quandl* y *Poloniex***. Se ha escogido Quandl para obtener los datos de los indicadores de *Bitcoin* y *Poloniex* para obtener los indicadores del precio de otras criptomonedas. Se podría añadir ***Blockchain*,** ya que Quandl realmente por debajo está utilizando esta otro servicio. *Quandl* debido a que proporciona información detallada de los indicadores de *Bitcoin* y porque proporciona un módulo de *Python* que facilita la obtención de datos. *Poloniex* por la amplia oferta de intercambio de criptomonedas.

Cada *API* tiene parámetros diferentes para realizar las peticiones y dependiendo el tipo de petición que se realice devuelve una información u otra. Por lo que se implementan diferentes funciones para obtener los datos. En algunas *APIs* como *Quandl* se requiere una clave de

identificación para realizar las peticiones, esto es así porque algunas de ellas están limitadas a realizar un máximo de n peticiones al día. Si se quiere superar ese límite, habría que pagar.

Existe una página web [38] muy interesante de un ingeniero de Google llamado Patrick Triest que muestra cómo capturar los datos de criptomonedas y realiza un análisis de los precios de criptomonedas. Es ideal para iniciarse en este tipo de análisis. Se ha utilizado código de su Github personal [39].

En este trabajo se van a ver ejemplos prácticos para conocer cómo funcionan las *APIs* de *Quandl* y *Poloniex*. A continuación, se va a mostrar unos ejemplos de funciones en *Python* que realizan las peticiones a las *APIs* para obtener los datos y para almacenar los datos en ficheros.

5.2.1 Quandl

El primer ejemplo es de la plataforma *Quandl* para datos financieros, económicos y alternativos. Los datos son accesibles a través de su *API* que se puede acceder mediante diferentes lenguajes de programación como *R* o *Python*. La plataforma proporciona un módulo propio que tiene funciones implementadas en *Pyhton*.

Para obtener los datos del precios de *Bitcoin* a través de la API Quandl , primero se importa el módulo y después, como se puede ver en el **Código *2,*** se define la función *get_quandl_data*. La función comprueba si existe el fichero con los datos en una ruta concreta predefinida, si lo encuentra, lee ese fichero y carga los datos. De lo contrario, realiza la petición a la *API* de *Quandl* para descargar los datos y almacenarlos en un archivo. La función tiene un parámetro de entrada que es *quandl_id.* Este parámetro define el tipo de información que se quiere recuperar.

```
# Se define una función Quandl para cargar los datos
"""pickle --> para no descargar de nuevo los mismos datos"""
"""La función devuelve un Dataframe Pandas"""

def get_quandl_data(quandl_id):
    """Se almacena un fichero .pkl como cache de los datos"""
    cache_path ='.\cryptocurrency_analysis_files\{}.pkl'
                        .format(quandl_id).replace('/','-')
    try:
        f = open(cache_path,'rb')
        df = pickle.load(f)
        print('Dataset {} cargado del cache'.format(quandl_id))
    except (OSError,IOError)as e:
        print('Descargando {} de Quandl'.format(quandl_id))
        df = quandl.get(quandl_id, returns="pandas")
        df.to_pickle(cache_path)
        print('Cargado {} de {} en el
cache'.format(quandl_id,cache_path))
    return df
```

Código 2. Definición de la función *get_quandl_data* [36].

Para adquirir los datos del precio de *Bitcoin* del *exchange* Kraken se llama a la función *get_quandl_data* con el parámetro de entrada *'BCHARTS/KRAKENUSD'* como se puede ver en **Código *3.***

```
# Se realiza una petición del precio de BTC del exchange Kraken
btc_usd_price_kraken = get_quandl_data('BCHARTS/KRAKENUSD')

# Se muestra la cabecera del Dataframe
btc_usd_price_kraken.head()
```

Código 3. Llamada a la función *get_quandl_data* y muestra de los datos [36].

Una vez ejecutado el código, se obtienen los datos correspondientes con la estructura de datos de tipo *Dataframe*. En la **Tabla *3*** podemos observar que tenemos, la fecha, la apertura y el cierre del mercado ese día, el máximo y mínimo, el volumen en *bitcoins*, el volumen de circulación y el precio medio.

Date	Open	High	Low	Close	Volume (BTC)	Volume (Currency)	Weighted Price
2014-01-07	874.67040	892.06753	810.00000	810.00000	15.622378	13151.472844	841.835522
2014-01-08	810.00000	899.84281	788.00000	824.98287	19.182756	16097.329584	839.156269
2014-01-09	825.56345	870.00000	807.42084	841.86934	8.158335	6784.249982	831.572913
2014-01-10	839.99000	857.34056	817.00000	857.33056	8.024510	6780.220188	844.938794
2014-01-11	858.20000	918.05471	857.16554	899.84105	18.748285	16698.566929	890.671709

Tabla 3. Dataframe de los datos históricos del precio de *Bitcoin* [36].

5.2.2 Poloniex

El segundo ejemplo es capturar los datos de la plataforma *exchange Poloniex*. Los datos son accesibles a través de su *API*. Para capturar los datos, se definen las funciones: *get_json_data* para para almacenar y cargar los datos y *get_crypto_data* para definir las peticiones que se van a realizar a la *API Poloniex*.

En el **Código *4*** se puede ver la definición de la función *get_json_data* que consta del parámetro *json_url* que es la petición a la *API* y el parámetro *cache_path* que es la ruta donde se almacenan los datos descargados. La función *get_json_data* comprueba si existe el fichero con los datos en una ruta concreta predefinida, si lo encuentra, lee ese fichero y carga los datos. De lo contrario, se realiza la petición a la *API* de *Poloniex* para descargar los datos en formato *json* y se almacenan en un archivo.

```
# Se define una función get_json_data para cargar los datos de la API
Poloniex
"""pickle --> para no descargar de nuevo los mismos datos"""
"""La función devuelve un Dataframe Pandas"""

def get_json_data(json_url,cache_path):

cache_path='.\cryptocurrency_analysis_files\{}.pkl'.format(cache_path)
    """Descargamos en cache los datos en formato json"""
    try:
        f = open(cache_path,'rb')
        df = pickle.load(f)
        print('Dataset {} cargado del cache'.format(json_url))
    except (OSError,IOError) as e:
        print('Descargando datos {} mediante la API
Poloniex'.format(json_url))
        df = pd.read_json(json_url)
        df.to_pickle(cache_path)
        print('Cargado {} de {} en el cache'.format(json_url,cache_path))
    return df
```

Código 4. Definición de la función *get_json_data* [36].

En **Código *5*** se puede ver que se definen el conjunto de parámetros que forman la petición de la *API* y después se define la función *get_crypto_data* que concatena los parámetros para crear la *URL* que corresponde a la llamada a la *API*.

```
# Se define la función genera las peticiones vía HTTP a Poloniex API y se
llamará a la función get_json_data para guardar los datos obtenidos.
base_url =
'https://poloniex.com/public?command=returnChartData&currencyPair={}&star
t={}&end={}&period={}'
start_date = datetime.strptime('2015-01-01', '%Y-%m-%d')
end_date = datetime.now()
# Periodos válidos: '15m': 900, '5m': 300, '30m': 1800, '4h': 14400,
'2h': 7200, '1d': 86400
period = 86400

def get_crypto_data(poloniex_pair):
    '''Captura de los datos de criptomonedas de la API Poloniex'''
    json_url =
base_url.format(poloniex_pair,start_date.timestamp(),end_date.timestamp()
,period)
    data_df = get_json_data(json_url,poloniex_pair)
    data_df = data_df.set_index('date')
    return data_df

# URL de ejemplo:
https://poloniex.com/public?command=returnChartData&currencyPair=BTC_ETH&
start=1420066800.0&end=1483225200.0&period=86400
```

Código 5. Definición de la función *get_crypto_data* [36].

Después, en el **Código *6*** se realiza una estructura de datos con un diccionario de criptomonedas donde se llama a la función *get_crypto_data* una vez por cada criptomoneda y se incorporan los datos en un único *Dataframe*. Como la mayoría de criptomonedas no se pueden comprar directamente en euros o en dólares, lo usuarios normalmente compran *bitcoins* y después los intercambian por otras criptomonedas. Por esta razón se descargan las ratios de cambio de *Bitcoin* a otras criptomonedas para convertir el valor a dólares. El resultado de la función devuelve un *Dataframe*.

```
# Diccionario de altcoins formado por un dataframe por cada criptomoneda.
# Cada dataframe contiene el ratio medio de intercambio entre altcoins y
BTC.

altcoins = ['ETH','LTC','XRP','ETC','STR','DASH','SC','XMR','XEM']

altcoin_data = {}
for altcoin in altcoins:
    coinpair = 'BTC_{}'.format(altcoin)
    crypto_price_df = get_crypto_data(coinpair)
    altcoin_data[altcoin] = crypto_price_df
```

Código 6. Creación de estructura de datos para el *dataframe* de criptomonedas [36].

A continuación, se muestran las últimas filas del *Dataframe* con los datos de *Ethereum* mediante el **Código *7***.

```
# Se observa las últimas filas de dataframe de ETH
altcoin_data['ETH'].tail()
```

Código 7. Muestra de los datos obtenidos mediante la *API Poloniex* [36].

Como resultado de ejecutar el código se obtiene el *Dataframe* de la **Tabla *4*** que corresponde con el ratio de cambio de *Bitcoin* a *Etherum (BTC/ETH)*. Se obtiene la fecha, la apertura y el cierre del mercado ese día, el máximo y mínimo, el volumen, el volumen de circulación y el precio medio para *Ethereum*.

date	close	high	low	open	quoteVolume	volume	weightedAverage
2018-08-06	0.058400	0.058650	0.057697	0.058215	6516.218631	379.809745	0.058287
2018-08-07	0.056320	0.058554	0.055300	0.058347	13308.258100	756.189357	0.056821
2018-08-08	0.056620	0.057501	0.055405	0.056320	16336.349294	925.227759	0.056636
2018-08-09	0.055635	0.057200	0.055453	0.056682	6084.771671	341.937158	0.056196
2018-08-10	0.055800	0.056412	0.055416	0.055669	3173.087717	177.246184	0.055859

Tabla 4. *Dataframe* de los datos históricos del precio de *Ethereum* [36].

5.3 Análisis previo, tratamiento y visualización de los datos

Una vez se conoce las fuentes de datos y la forma de obtener la información relevante de las criptomonedas, el siguiente paso es realizar un análisis previo y una visualización de los datos. A continuación, se enumeran los análisis que se realizan.

- **5.3.1 Análisis del precio medio de Bitcoin.**
- **5.3.2 Análisis del precio medio de las principales criptomonedas.**
- **5.3.3 Correlación del precio medio de Bitcoin y las principales criptomonedas.**
- **5.3.4 Visualización de los indicadores de Bitcoin.**

5.3.1 Análisis del precio medio de Bitcoin

A partir de los datos obtenidos de la *API Quandl* y con la librería *plotly* de *Python* se definen funciones para crear gráficos interactivos del precio medio del *Bitcoin*. A partir **Código *8*** se define un gráfico con los datos del *Dataframe* que se ha mostrado anteriormente en **Tabla *4*** utilizando el precio medio del *Bitcoin*.

```
# Gráfico del precio de BTC
btc_trace = go.Scatter( x=btc_usd_price_kraken.index,
                        y=btc_usd_price_kraken['Weighted Price'])
py.iplot([btc_trace])
```

Código 8. Definición del gráfico con la librería *plotly* [36].

Al ejecutar el código, se obtiene como salida un gráfico interactivo. El gráfico tiene diferentes opciones, como la posibilidad de que al pasar el ratón por la encima, se muestre la etiqueta del día y del precio de ese día. En **Gráfico *1*** se observan fechas en las que el valor del precio de *Bitcoin* es cero. Esto es debido a que para esas fechas por algún motivo en el *Exchange Kraken* no tienen datos, en el *Dataframe* el valor para esos días es *Not a Number (NaN)* .

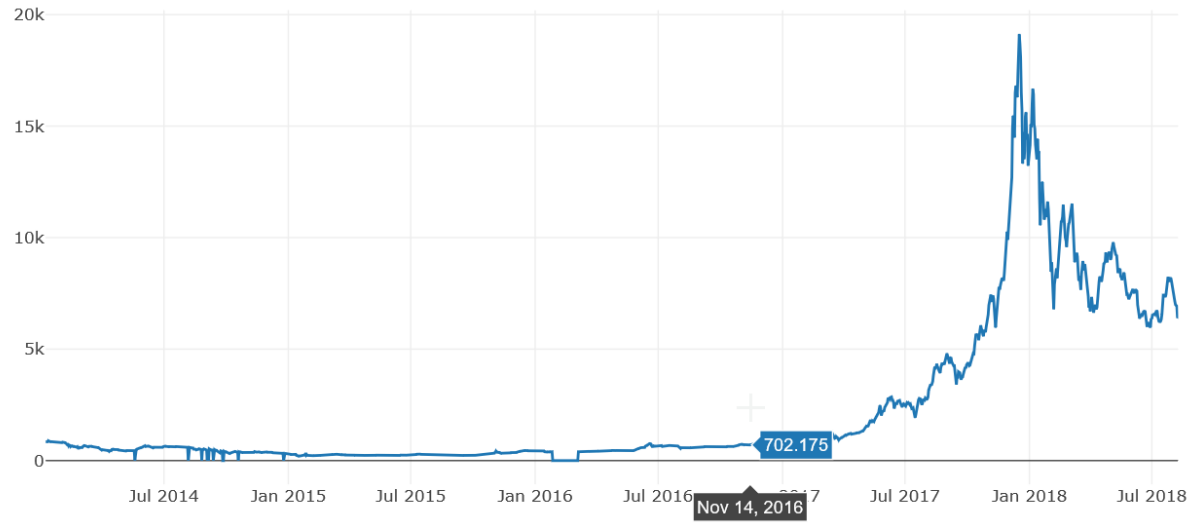

Gráfico 1. Precio medio de *Bitcoin* en dólares [36].

Para solucionar este hecho, se descargan datos de otros *exchanges* como *Coinbase*, *Bitstamp* y *Itbit* con la *API* de *Poloniex* mediante llamadas a la función que se ha descrito anteriormente.

El siguiente paso es limpiar en el *Dataframe* los valores que son *Not a Number (NaN).* Después, en el **Código *9*** se define una función más compleja para graficar el *Dataframe* combinado. El resultado se pude ver en **Gráfico *2***.

```
# Se define la función para visualizar los datos
def df_scatter(df, title,seperate_y_axis=False,
y_axis_label='',scale='linear',initial_hide=False):
    # Se definen la lista de los nombres de cada dataframe como una lista
label_arr = ['BITSTAMP', 'COINBASE', 'ITBIT', 'KRAKEN']
    label_arr = list(df)
    # Aplicamos una función lambda para mapear cada columna y asignar la
etiqueta correspondiente
    # Se guarda como otra lista series_arr
    series_arr = list(map(lambda col:df[col],label_arr))

    # Se definen los parametros de la salida gráfica
    layout = go.Layout(
        title = title,
        legend = dict(orientation='h'),
        xaxis = dict(type='date'),
        yaxis = dict(
            title = y_axis_label,
            showticklabels = not seperate_y_axis,
            type = scale
        )
    )

    # Se define la configuración del eje y
    y_axis_config = dict(
        overlaying = 'y',
        showticklabels = False,
        type = scale
    )

    # Se define la visibilidad
    visibility = 'visible'
    if initial_hide:
        visibility = 'legendonly'

    # Se define la forma para cada serie de datos
    trace_arr = []
    for index, series in enumerate(series_arr):
        trace = go.Scatter(
        x = series.index,
        y = series,
        name = label_arr[index],
        visible = visibility
        )

        #Añadir un eje separado para cada serie
        if seperate_y_axis:
            trace['yaxis'] = 'y{format}'.format(index + 1)
            layout['yaxis{}'.format(index + 1)] = y_axis_config
        trace_arr.append(trace)

    fig = go.Figure(data = trace_arr, layout = layout)
    py.iplot(fig)
```

Código 9. Definición de la función para crear una gráfica de series temporales [36].

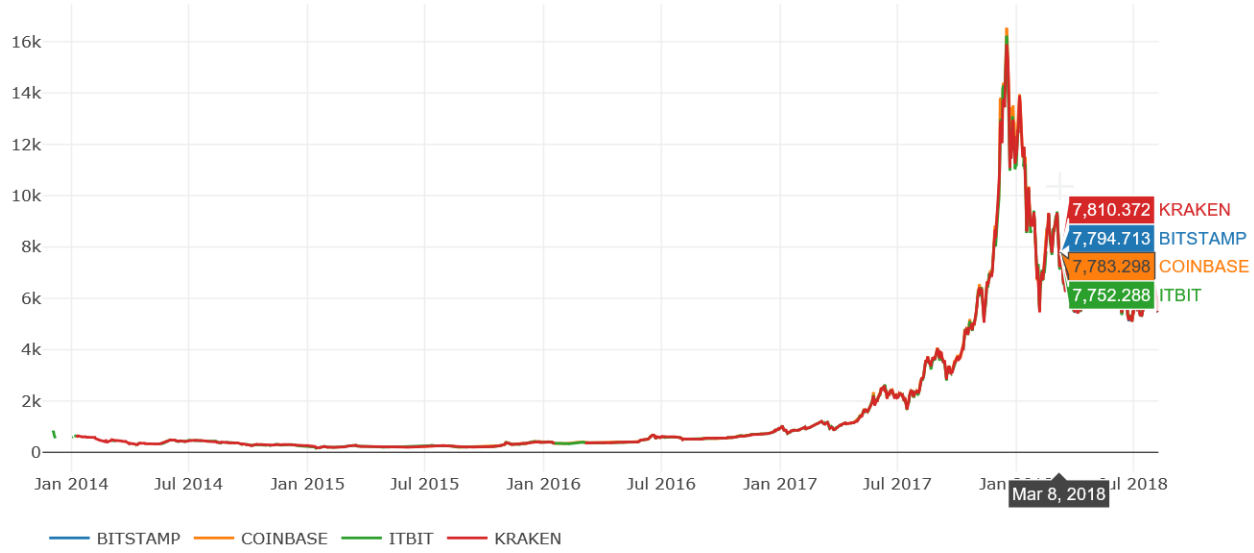

Gráfico 2. Precio medio de *Bitcoin* por cada *exchange* en dólares [36].

Finalmente, se calcula una nueva columna para obtener el promedio del precio medio de *Bitcoin* para cada *exchange*. Después se llama a la función que genera un gráfico con la nueva columna y obtenemos en el **Gráfico *3*** que muestra el promedio del precio medio de *Bitcoin* de cada *exchange*.

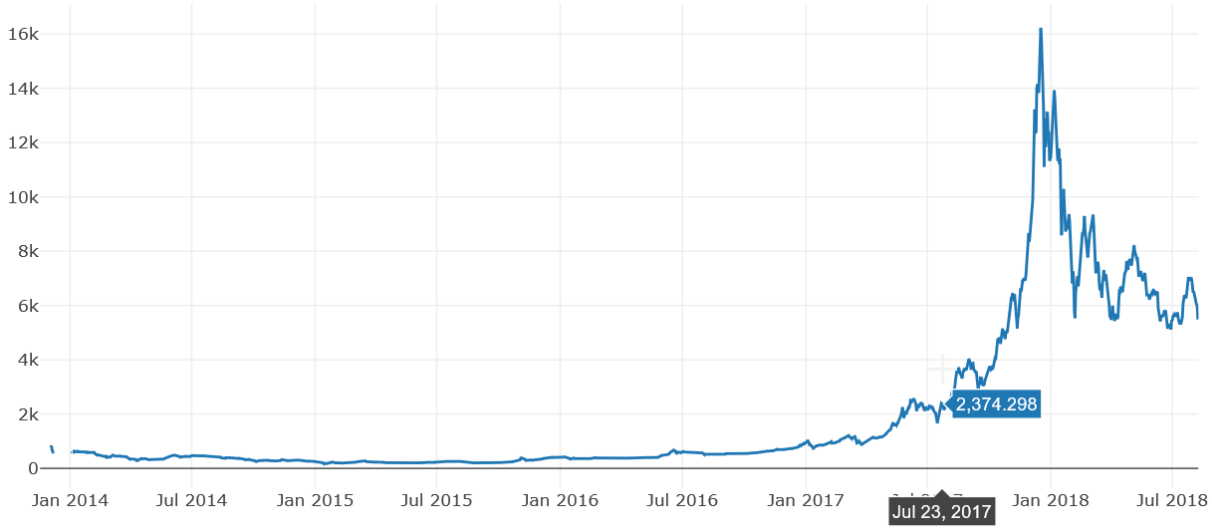

Gráfico 3. Promedio del precio medio de *Bitcoin* de cada *exchange* en dólares [36].

5.3.2 Análisis del precio medio de las principales criptomonedas

A continuación, a partir de los datos obtenidos de la *API Poloniex* y con la librería *plotly* de *Python* se definen un gráfico interactivo del precio medio de las principales criptomonedas. Para realizar la comparación entre criptomonedas se han elegido las criptomonedas con mayor capitalización del mercado. Se pueden ver en la siguiente

Tabla 5.

Abreviatura	Nombre criptomoneda	URL criptomoneda
DASH	Dash	https://www.dash.org
ETC	Ethereum Classic	https://ethereumclassic.github.io
ETH	Ethereum	https://www.ethereum.org
LTC	LiteCoin	https://developers.coinbase.com/
SC	Siacoin	https://sia.tech
STR o XML	Starcoin o Stellar	https://www.stellar.org
XEM	NEM	https://nem.io/es/
XRM	Monero	https://getmonero.org
XRP	Ripple	https://ripple.com/xrp/
BTC	Bitcoin	https://bitcoin.org/

Tabla 5. Tabla de las principales criptomonedas del mercado [36].

En el **Código *10*** se muestra la conversión a dólares de las distintas criptomonedas para comparar en las mismas unidades el resto de criptomonedas. Se crea un *Dataframe* combinado que se compone de la fecha y del precio de cada criptomoneda en dólares. Se añade el precio de *Bitcoin* y se obtiene como resultado el *Dataframe* de la **Tabla *6***.

```
# Conversión del precio de las criptomonedas a USD.
for altcoin in altcoin_data.keys():
    altcoin_data[altcoin]['price_usd'] =
altcoin_data[altcoin]['weightedAverage']*btc_usd_datasets['avg_btc_price_
usd']

# Crear un único dataframe combined_df con el precio de cada criptomoneda
combined_df = merge_dfs_on_column(list(altcoin_data.values()),
list(altcoin_data.keys()), 'price_usd')

# Añadir el precio de BTC en el dataframe
combined_df['BTC'] = btc_usd_datasets['avg_btc_price_usd']

# Se realiza el gráfico con el dataframe combined_df para ver las
criptomonedas
df_scatter(combined_df, 'Precio de las criptomonedas (USD)',
seperate_y_axis = False , y_axis_label='Valor (USD)', scale = 'log')
```

Código 10. Conversión, creación del *Dataframe* y visualización [36].

date	DASH	ETC	ETH	LTC	SC	STR	XEM	XMR	XRP	BTC
2018-08-06	176.629864	15.697879	351.682590	64.116267	0.006577	0.204662	0.123388	100.075218	0.365217	6033.654433
2018-08-07	169.905672	16.463070	341.113459	61.702503	0.006424	0.204472	0.121627	97.744292	0.344829	6003.292771
2018-08-08	157.853116	13.666822	313.358373	55.298585	0.005533	0.177106	0.106452	84.641921	0.294789	5532.835196
2018-08-09	157.598694	13.002755	311.164008	54.195487	0.005482	0.182893	0.104597	85.879746	0.299339	5537.163572

Tabla 6. *Dataframe* del precio de las principales criptomonedas [36].

Por último, en el **Código *10*** se llama a la función para crear un gráfico interactivo de las principales criptomonedas con una escala logarítmica como se puede ver en la ***Gráfico 4.*** En él se puede observar que a partir del 2017 aproximadamente, los precios de las diferentes criptomonedas parecen seguir la misma tendencia.

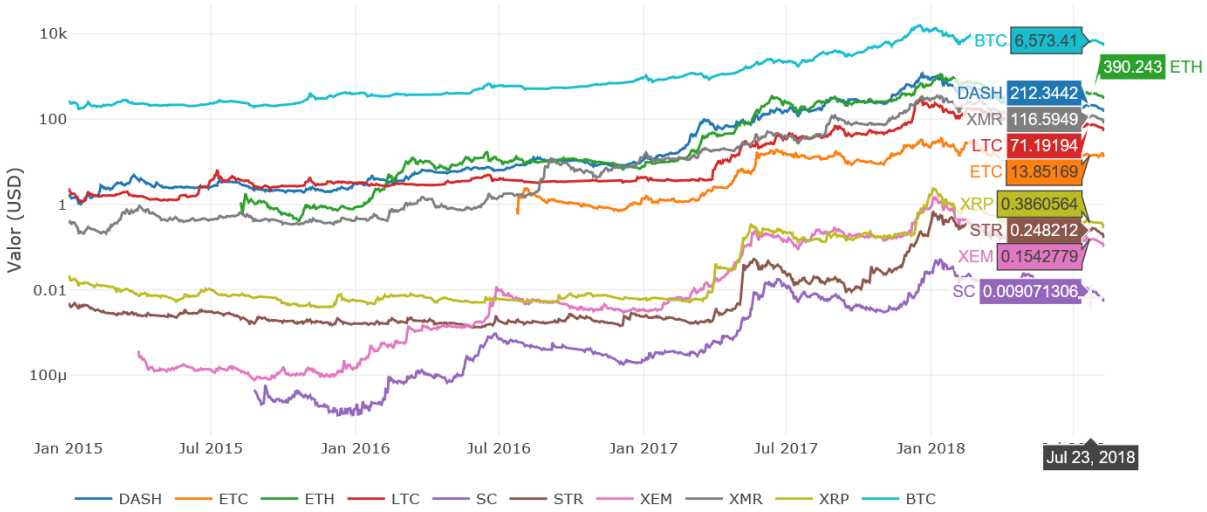

Gráfico 4. Precio medio de las principales criptomonedas [36].

5.3.3 Correlación del precio medio de Bitcoin y las principales criptomonedas

A continuación, se realiza una correlación entre las diferentes criptomonedas para ver la relación que existe entre ellas en los periodos de 2016, 2017 y de enero a agosto del 2018.

En el **Código *11*** se calcula el coeficiente de correlación de Pearson para el año 2016 y se trabaja con los incrementos y no con los valores absolutos. Después se aplica la correlación con el método de Pearson.

```
# Seleccionar los datos que sean del año 2016
combined_df_2016 = combined_df[combined_df.index.year == 2016]

# Aplicamos el coeficiente de correlación de Pearson
combined_df_2016.pct_change().corr(method='pearson')
```

Código 11. Selección del *Dataframe* del año 2016 y la correlación de Pearson [36].

Ejecutando el código se obtienen los datos de criptomonedas del año 2016 y se aplica la correlación de Pearson. El resultado es un *Dataframe* que se puede ver en la **Tabla *7***.

	DASH	ETC	ETH	LTC	SC	STR	XEM	XMR	XRP	BTC
DASH	1.000000	0.009257	0.130902	0.004937	0.032504	0.071789	0.019884	0.127224	0.107930	0.007432
ETC	0.009257	1.000000	-0.185372	-0.126139	-0.008077	-0.098499	-0.078601	-0.102095	-0.050868	-0.170523
ETH	0.130902	-0.185372	1.000000	-0.053582	0.172812	0.043997	0.047846	0.091030	0.097538	0.006477
LTC	0.004937	-0.126139	-0.053582	1.000000	0.017845	0.130447	0.168238	0.135977	0.075755	0.755778
SC	0.032504	-0.008077	0.172812	0.017845	1.000000	0.147741	0.108698	0.050074	0.027737	0.041985
STR	0.071789	-0.098499	0.043997	0.130447	0.147741	1.000000	0.230956	0.035057	0.330421	0.099985
XEM	0.019884	-0.078601	0.047846	0.168238	0.108698	0.230956	1.000000	0.019510	0.108758	0.235212
XMR	0.127224	-0.102095	0.091030	0.135977	0.050074	0.035057	0.019510	1.000000	0.036389	0.135429
XRP	0.107930	-0.050868	0.097538	0.075755	0.027737	0.330421	0.108758	0.036389	1.000000	0.071934
BTC	0.007432	-0.170523	0.006477	0.755778	0.041985	0.099985	0.235212	0.135429	0.071934	1.000000

Tabla 7. *Dataframe* filtrado el año 2016 aplicando la correlación de Pearson [36].

El siguiente paso es crear una función como se ve en **Código *12*** ,que permita crear un mapa de calor con la finalidad de observar si existe correlación y si es positiva o negativa de una forma muy rápida y visual.

```
# Función de visualización
def correlation_heatmap (df, title, absolute_bounds = True):
    '''Gráfico heatmap del dataframe de correlación'''
    heatmap = go.Heatmap(
    z = df.corr(method='pearson').as_matrix(),
    x = df.columns,
    y = df.columns,
    colorbar  = dict(title = 'Coeficiente de Pearson')
    )
    layout = go.Layout(title = title)

    if absolute_bounds:
        heatmap['zmax'] = 1.0
        heatmap['zmin'] = -1.0

    fig = go.Figure(data=[heatmap],layout = layout)
    py.iplot(fig)
```

Código 12. Función para crear un gráfico de tipo mapa de calor [36].

Una vez definida la función mediante ***Código 13*** se llama a la función y se le pasa como parámetros el *Dataframe.* Es importante destacar que se aplica la función *pct_change()* porque se trata de series temporales que probablemente no son estacionarias. Esta función permite transformar el *Dataframe* y trabajar con los incrementos o decrementos y no con los valores absolutos.

```
# Función de visualización
correlation_heatmap(combined_df_2016.pct_change()," Correlación de
criptomonedas en 2016")
```

Código 13. Llamada a la función que crea el gráfico del mapa de calor [36].

Se obtiene el siguiente **Gráfico *5*** donde se observa una correlación positiva entre *Bitcoin* y *Litecoin*. Y una correlación negativa entre *Ethereum Classic* y las demás criptomonedas.

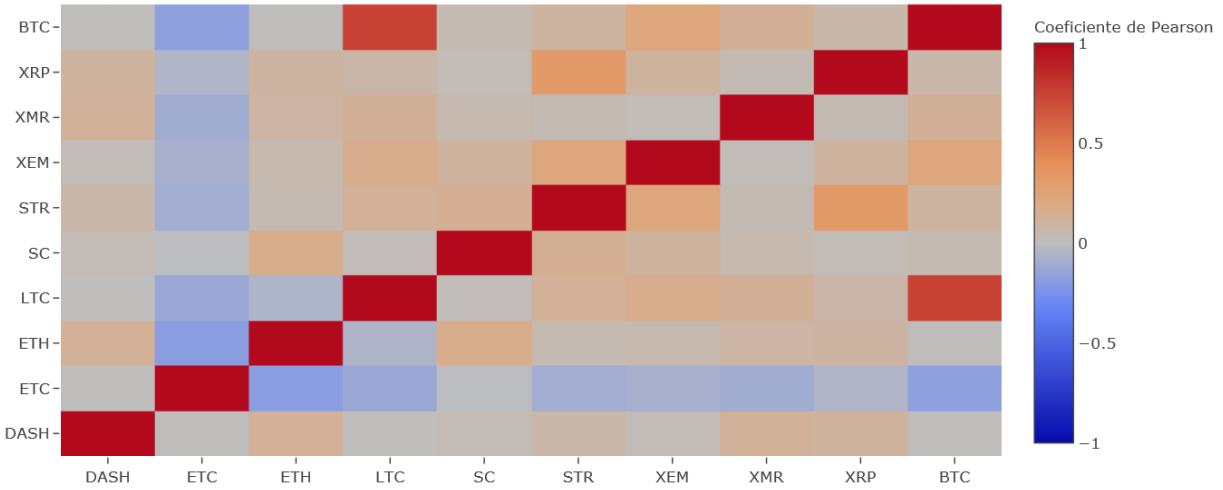

Gráfico 5. Correlación de Pearson de las principales criptomonedas en el 2016 [36].

Se realiza el mismo procedimiento y se obtienen los siguientes gráficos de mapa de calor. Se obtiene una correlación de Pearson de las principales criptomonedas empleando los datos del año 2017. En **Gráfico *6*** se obtiene que en el año 2017 todas las criptomonedas tienen una correlación positiva, excepto *Ripple* con *Dash* y *Ripple* con *Etherum Classic*.

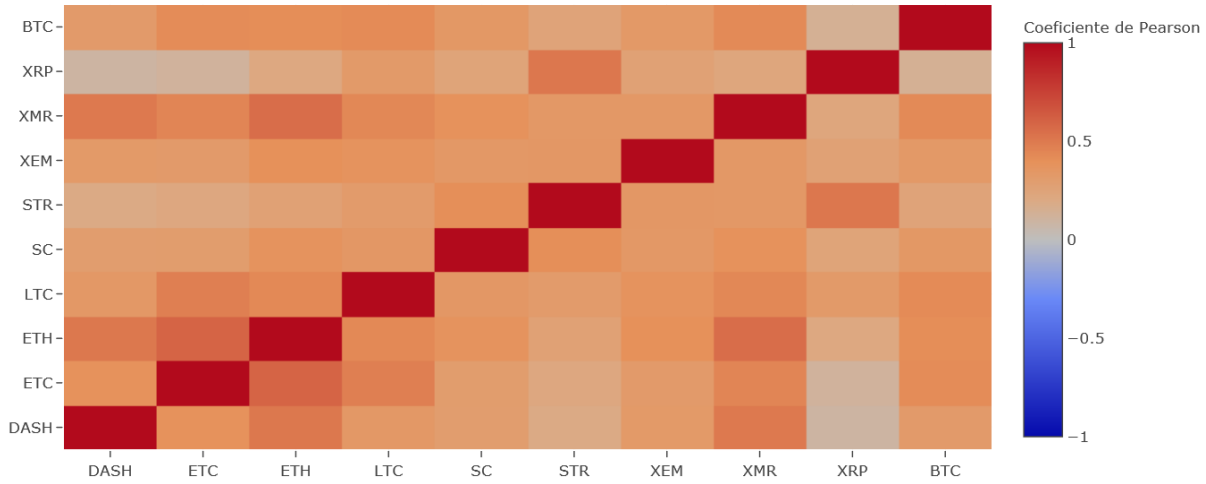

Gráfico 6. Correlación de Pearson de las principales criptomonedas en el 2017 [36].

Se realiza el mismo procedimiento y se realiza la correlación de Pearson de las principales criptomonedas empleado los datos de enero hasta agosto del 2018. En la **Gráfico *7*** se demuestra que todas las criptomonedas tienen una fuerte correlación positiva.

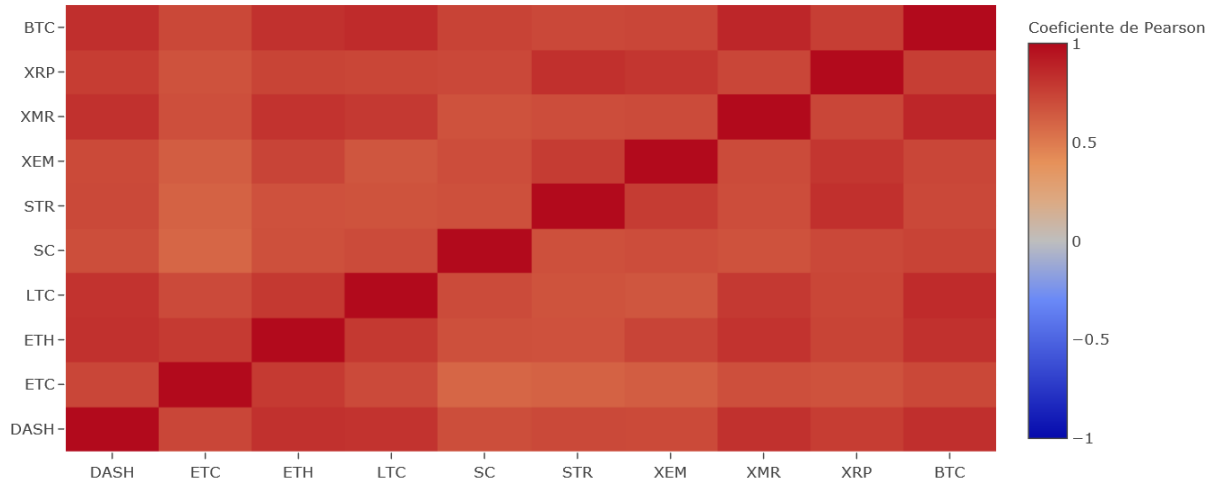

Gráfico 7. Correlación de Pearson de las principales criptomonedas de enero a agosto del 2018 [36].

Como resultado de la correlación, se puede afirmar que la evolución del precio de todas las criptomonedas que se han analizado tienen una gran fluctuación y una tendencia a la correlación. Esta tendencia entre *Bitcoin* y las principales criptomonedas se vuelve más fuerte a medida que pasa el tiempo. Por lo tanto, podemos afirmar que existe una relación entre las diferentes criptomonedas, sin embargo, la correlación no implica causalidad.

5.3.4 Visualización de los indicadores de Bitcoin

Seguidamente, se obtienen los datos de los indicadores de *Bitcoin* de la *API Quandl* y se visualizan mediante la función definida anteriormente en el **Código *9***.

Precio medio de *Bitcoin*

En **Gráfico *8*** se observa el periodo desde 2016, con el precio aproximado de 400 dólares, hasta finales de Agosto donde el precio está alrededor de los 6.500 dólares. Cabe destacar el precio del 17 de diciembre del 2017 llegando prácticamente a los 19.500 dólares.

Gráfico 8. Precio medio del Bitcoin en dólares [36].

Número de *bitcoins*

En el **Gráfico *9*** se puede ver el número de *bitcoins*. Actualmente la recompensa por añadir un bloque a la cadena de bloques es de 12,5 bitcoins aproximadamente cada 10 minutos. Se observa que cada 210,000 bloques o lo que es lo mismo cada cuatro años la tasa de creación de *bitcoins* se reduce a la mitad. Se puede ver en el gráfico un cambio de pendiente por este motivo a finales de 2013 y a finales de 2017.

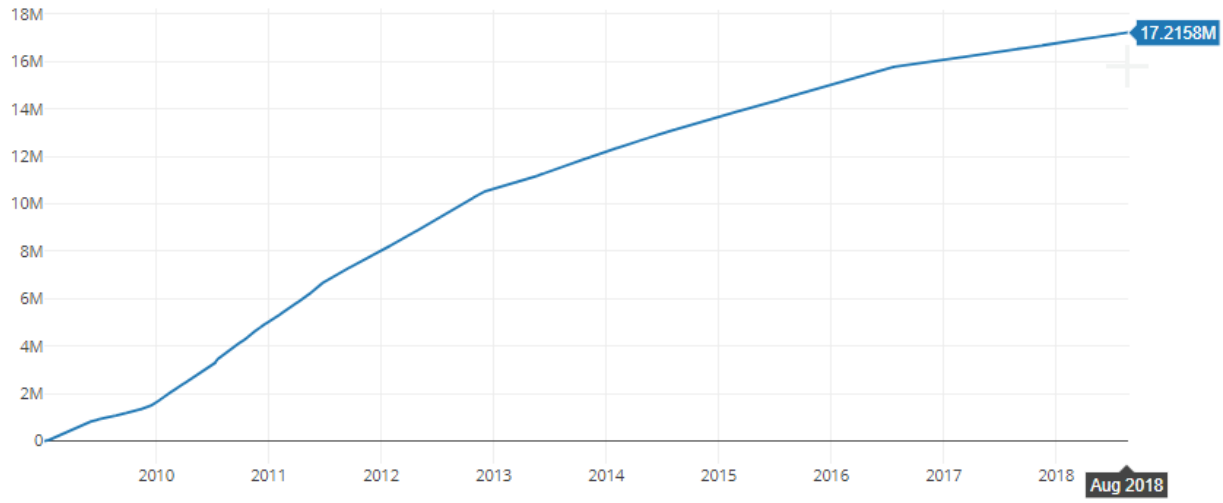

Gráfico 9. Número total de *bitcoins* [36].

Capitalización de mercado *Bitcoin*

En **Gráfico *10*** se puede ver que la capitalización es prácticamente igual al precio de *Bitcoin.* Llegando a un máximo de capitalización de más de 325.000 millones de dólares el 17 de diciembre del 2017.

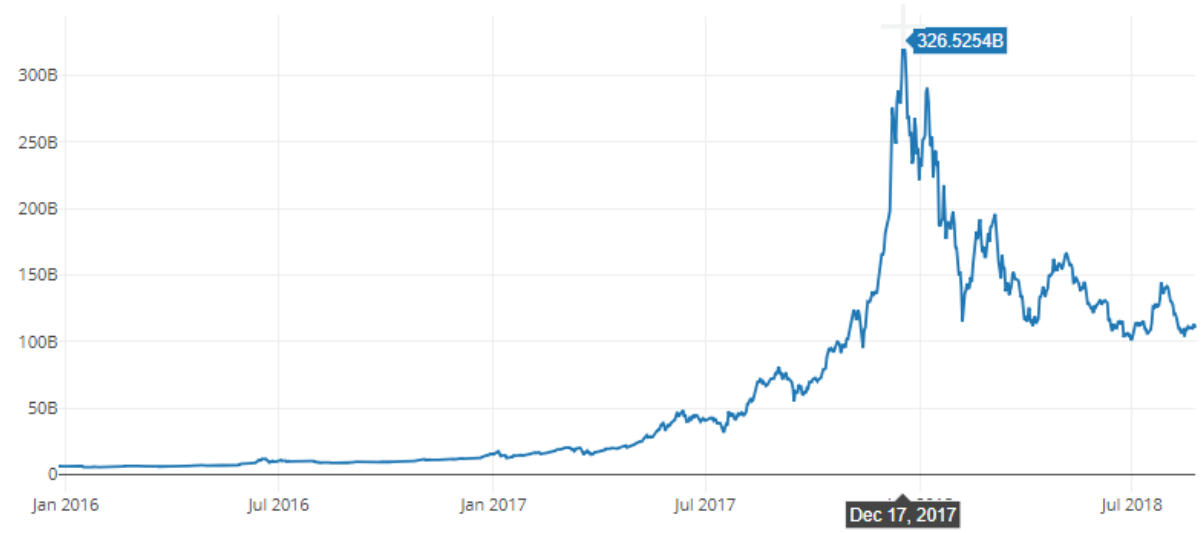

Gráfico 10. Capitalización del mercado de *Bitcoin* en dólares [36].

Direcciones *bitcoin*

En el **Gráfico *11*** se observa el número creciente de direcciones *bitcoin* desde que se creó en el 2009. Además de que hay grandes fluctuaciones en poco tiempo. El máximo corresponde al día 15 de diciembre con más de 1 millón de direcciones *bitcoin*.

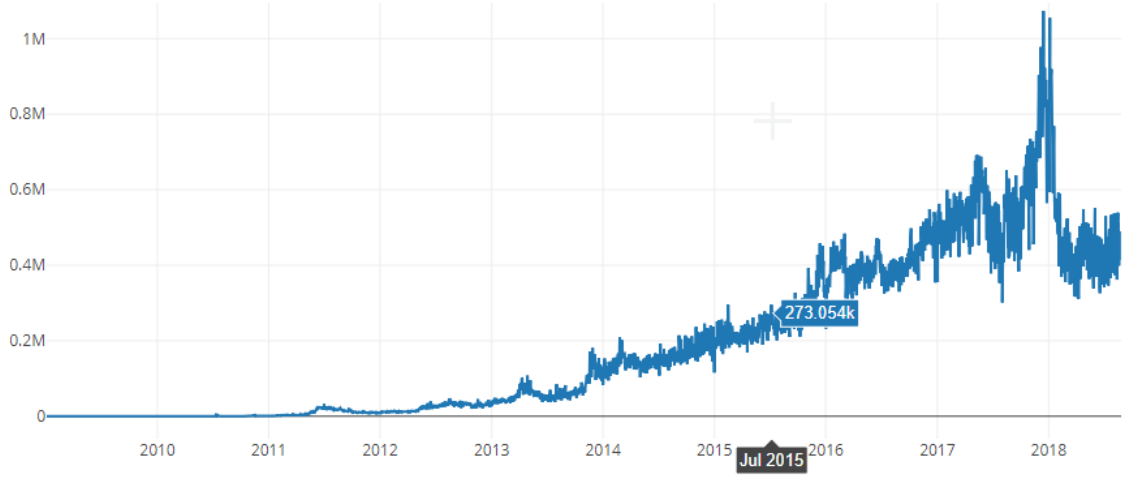

Gráfico 11. Número de direcciones *Bitcoin* [36].

Volumen de cambio de *bitcoins* a dólares

En el **Gráfico *12*** se puede ver el volumen de cambio de dólares a *bitcoins* de enero del 2017 hasta mediados de agosto del 2018. El máximo volumen de cambio en un día corresponde al 23 de diciembre del 2017 superando los 5.350 millones de dólares de volumen de cambio en un día.

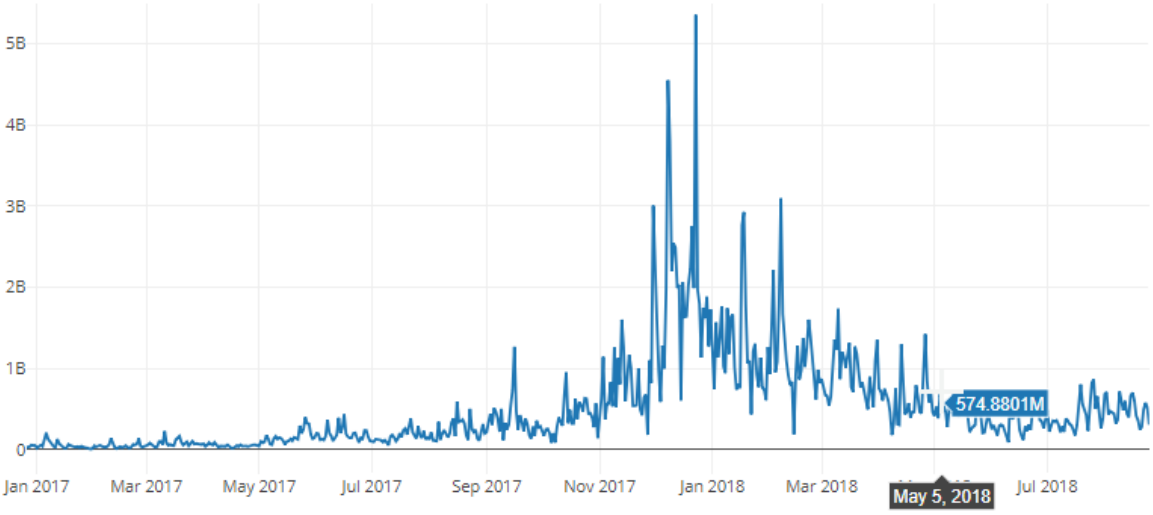

Gráfico 12. Ratio de volumen de cambio de *Bitcoin* en dólares [36].

Número de transacciones de *bitcoins*

En el **Gráfico *13*** se puede ver la fluctuación en el número de transacciones de bitcoins. El máximo de transacciones corresponde al día 15 de diciembre con más de 490.000 transacciones.

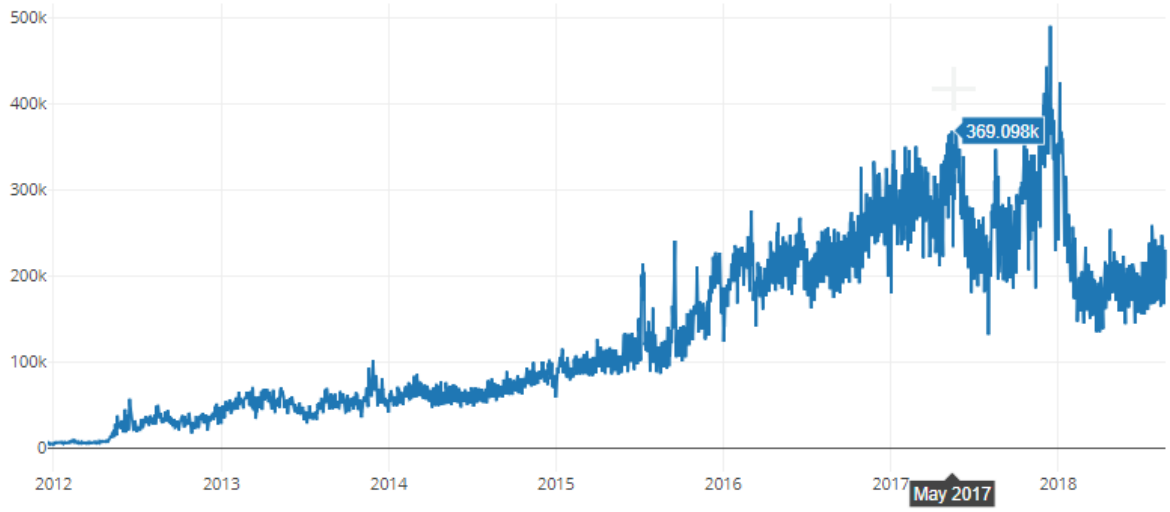

Gráfico 13. Número de transacciones *Bitcoin* [36].

Hash rate de *Bitcoin*

En el **Gráfico *14*** se puede ver la evolución y la fluctuación del hash rate. La unidad de medida es Tera Hashes por segundo (TH/s). 1TH/s equivale a 1 billón de hash por segundo. La tasa de hash rate del 12 de agosto es de 54 millones de TH/s.

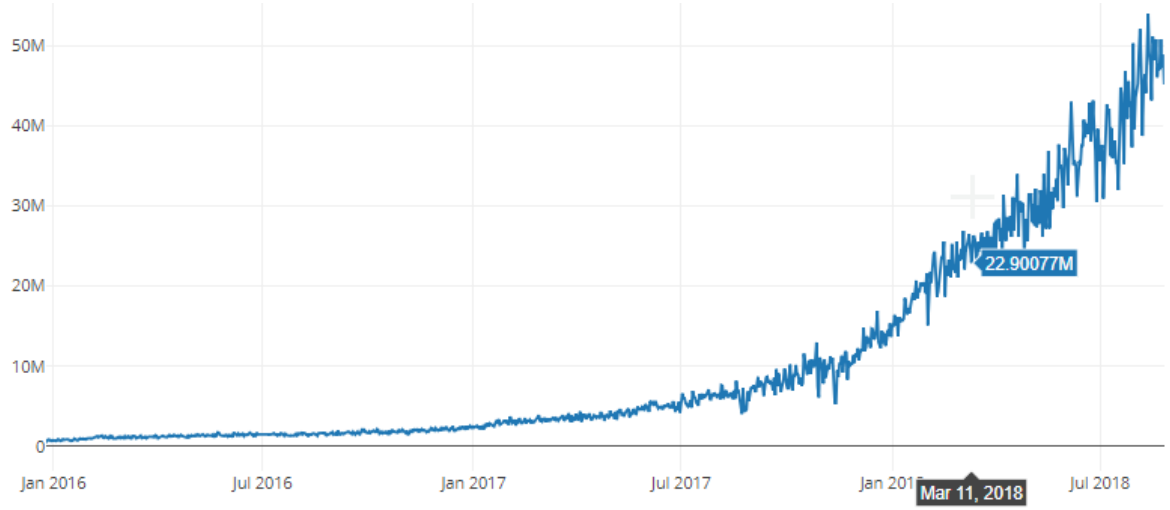

Gráfico 14. Ratio de Hash rate de *Bitcoin* en Terahashes por segundo [36].

Dificultad *Bitcoin*

En **Gráfico *15*** se observa la evolución de la dificultad de *Bitcoin.* Se observan los escalones que corresponden a los ajustes de la dificultad de la red de *Bitcoin* que se producen aproximadamente cada 2 semanas para continuar añadiendo bloques a la cadena de bloques con un promedio de 10 minutos.

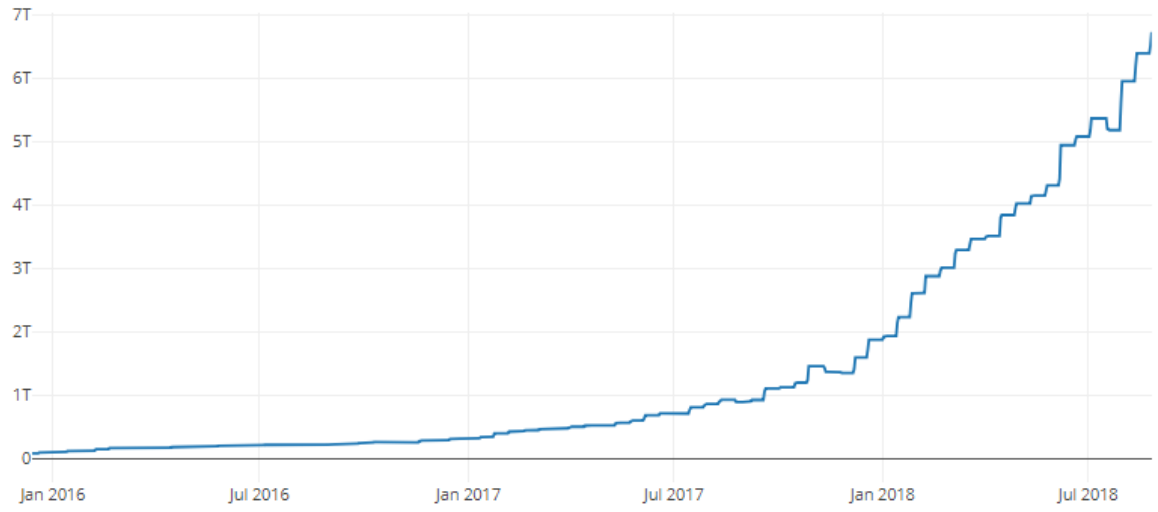

Gráfico 15. Dificultad *Bitcoin* [36].

Recompensa de los mineros de *Bitcoin*

En la **Gráfico *16*** se pude ver la recompensa obtenida por los mineros. La recompensa máxima corresponde al día 18 de diciembre del 2017 con 53 millones de dólares.

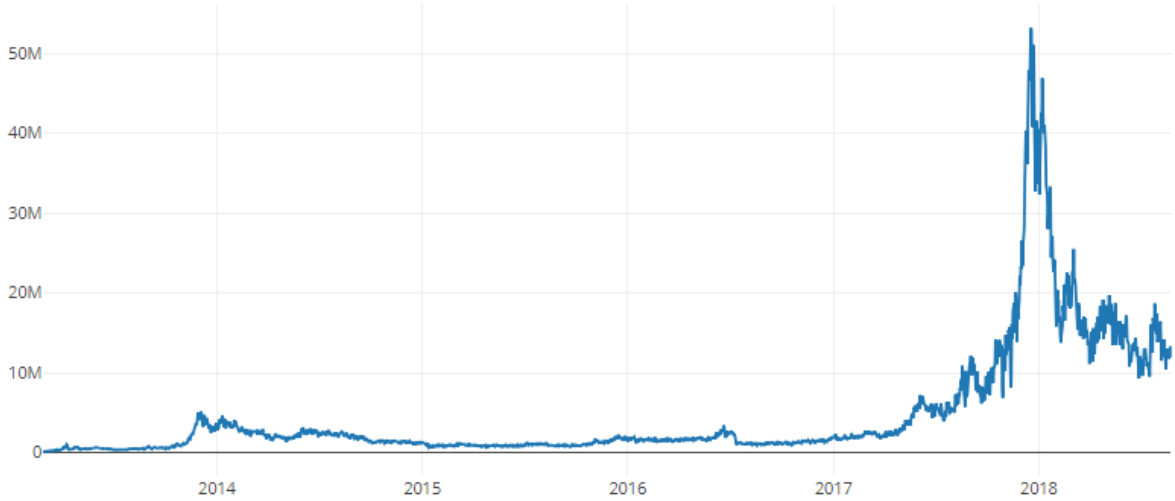

Gráfico 16. Recompensa de los mineros de *Bitcoin* en dólares [36].

Dificultad vs Recompensa de Bitcoin

En la **Gráfico *17*** destaca los datos cruzados entre la dificultad de la red *Bitcoin* y la recompensa de los mineros de Bitcoin. Se observa el periodo de mediados del 2010 a mediados de 2011 en el que era rentable para los usuarios minar *bitcoins*.

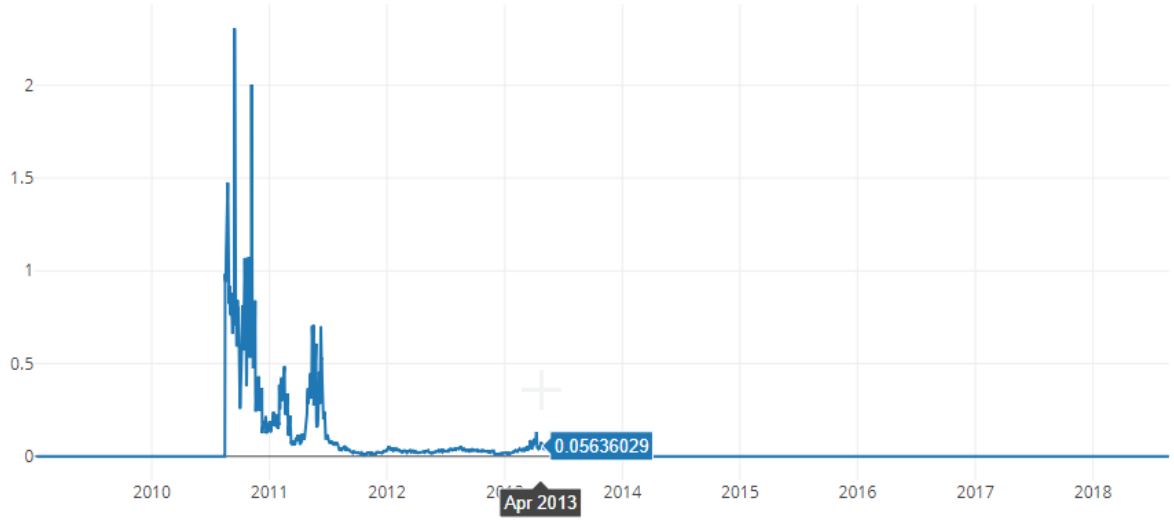

Gráfico 17. Dificultad vs Recompensa de los mineros de *Bitcoin* [36].

5.4 Desarrollo de la comparativa

Se enlaza el contexto, el estado del arte y el análisis previo para alcanzar el objetivo del proyecto, el planteamiento es establecer la comparativa implementando un modelo de regresión lineal múltiple y un modelo de red neuronal para estimar el precio de *Bitcoin.* Después comparar los resultados obtenidos mediante el error *Mean Absolute Error (MAE).*

Se escoge predecir el precio de *Bitcoin* porque es la primera criptomonedas que apareció y posee la mayor capitalización del mercado, más del 50% a mediados de agosto del 2018. Para realizar la comparativa se utilizan los datos de los indicadores vistos en el capítulo del análisis previo. A continuación, se verán el apartado de regresión lineal múltiple y la sección de la red neuronal artificial.

5.4.1 Regresión lineal múltiple

Se aplica un modelo de regresión lineal múltiple utilizando el estimador *Ordinary Lest Square (OLS).* Este permite estimar la relación entre una variable dependiente y un conjunto de variables explicativas. Este método de mínimos cuadrados permite conocer la ecuación de regresión y sus componentes de manera que minimiza la suma de las diferencias cuadráticas de los valores observados y de los valores calculados o predichos.

A continuación, se muestra la ecuación que define el modelo de regresión múltiple:

$$y = a + b_1 + b_1 x_1 + \ldots + b_n x_n + e$$

Donde y es la variable dependiente, α es la intersección del modelo, $b_1, b_2 \ldots b_n$ son los coeficientes de regresión de cada variable explicativa, x_1, $x_2 \ldots x_n$ representan cada observación , u indica el error. En la **Tabla *8*** Las variables explicativas que se toman como entrada del modelo son los indicadores de Bitcoin en la excepto el precio medio de *Bitcoin*, *price_btc*, que es la variable explicada.

ID	Nombre de la variables	Descripción de la variable
y	*price_btc*	El precio medio de *bitcoin.*
x_1	*total_number_btc*	El número total de *bitcoins.*
x_2	*market_capitalization_btc*	La capitalización del mercado en dólares.
x_3	*address_btc*	El número de direcciones *Bitcoin*.
x_4	*exchange_trade_btc*	El ratio de cambio de Bitcoins en dólares.
x_5	*transactions_btc*	Número de transacciones de *Bitcoin.*
x_6	*hash_rate_btc*	Ratio de *Hash Rate* en Terahashes por segundo.
x_7	*difficulty_btc*	Dificultad de *Bitcoin.*
x_8	*miners_revenue_btc*	Recompensa de los mineros de Bitcoin en dólares.

Tabla 8. Indicadores de Bitcoin para aplicar la regresión.

Incluyendo las variables, se obtiene la siguiente ecuación de regresión múltiple:

$$y = a + b_1 + b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x_4 + b_5 x_5 + b_6 x_6 + b_7 x_7 + b_8 x_8 + e$$

Se aplica la regresión lineal con el módulo *sklearn.* Primero se preparan los datos para después con el **Código *14*** crear el modelo y realizar los cálculos obteniendo como resultado el informe de la ***Figura 15***.

```
# Se añade la constante
X = sm.add_constant(X)
# Se crea el modelo de regresión
model = sm.OLS(y, X).fit()
# Realizar los cálculos
predictions = model.predict(X)
```

Código 14. Creación del modelo de regresión [36].

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              price_btc   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                 1.568e+06
Date:                Sat, 01 Sep 2018   Prob (F-statistic):               0.00
Time:                        14:29:47   Log-Likelihood:                -4844.0
No. Observations:                 968   AIC:                             9706.
Df Residuals:                     959   BIC:                             9750.
Df Model:                           8
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
const                         -481.0375     77.626     -6.197      0.000    -633.374    -328.702
total_number_btc              3.331e-05   5.18e-06      6.425      0.000    2.31e-05    4.35e-05
market_capitalization_btc     5.918e-08    1.5e-10    393.808      0.000    5.89e-08    5.95e-08
address_btc                      0.0002    3.9e-05      4.451      0.000    9.72e-05       0.000
exchange_trade_btc            5.218e-09   4.09e-09      1.277      0.202    -2.8e-09    1.32e-08
transactions_btc                -0.0003   7.02e-05     -4.030      0.000      -0.000      -0.000
hash_rate_btc                -2.392e-06   8.87e-07     -2.696      0.007   -4.13e-06   -6.51e-07
difficulty_btc               -2.692e-11   7.04e-12     -3.826      0.000   -4.07e-11   -1.31e-11
miners_revenue_btc             1.96e-06   9.93e-07      1.974      0.049    1.12e-08    3.91e-06
==============================================================================
Omnibus:                      534.067   Durbin-Watson:                   1.547
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           139417.433
Skew:                           1.319   Prob(JB):                         0.00
Kurtosis:                      61.734   Cond. No.                     1.44e+14
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.44e+14. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Figura 15. Resultados del modelo de regresión con el estimador OLS [36].

Se puede ver que el valor de R^2 es muy alto, lo que quiere decir que el modelo se ajusta muy bien. Además, se crea un Dataframe para poder visualizar los valores observados y los calculados en la regresión. En el **Gráfico *18*** se puede ver que las diferencias entre los valores observados y los valores calculados o predichos. Las diferencias son mínimas.

Además, si se realiza la diferencia del valor observado menos el valor obtenido para todo el conjunto y se realiza la suma de todas las diferencias, teóricamente el valor obtenido debería ser cero. Se ha realizado el cálculo y el valor obtenido es muy cercano al cero. Con lo que se

confirma la hipótesis de que el modelo se ajusta correctamente a los datos de entrada. El error *MAE* que se obtiene del modelo con los datos de entrenamiento es de 11,01.

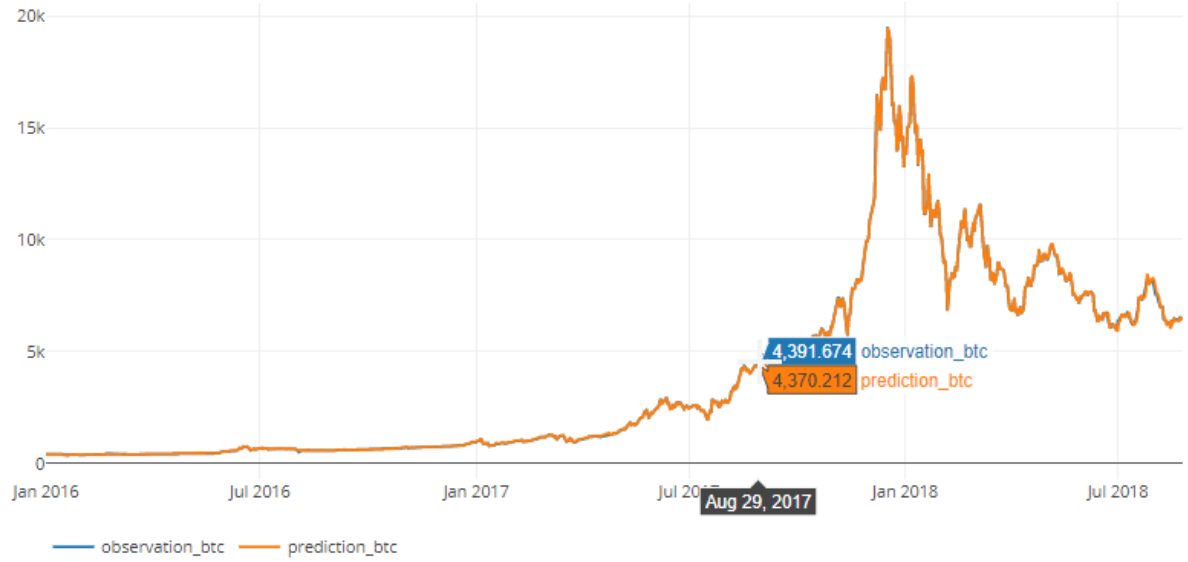

Gráfico 18. Observaciones y predicciones del modelo de regresión lineal de los datos de entreno [36].

El siguiente paso es introducir en el modelo predictivo los datos de. Se obtiene como resultado el error *MAE* de 91,79. En la **Gráfico *19*** se puede ver que la predicción parece funcionar bien con el precio de *bitcoin*.

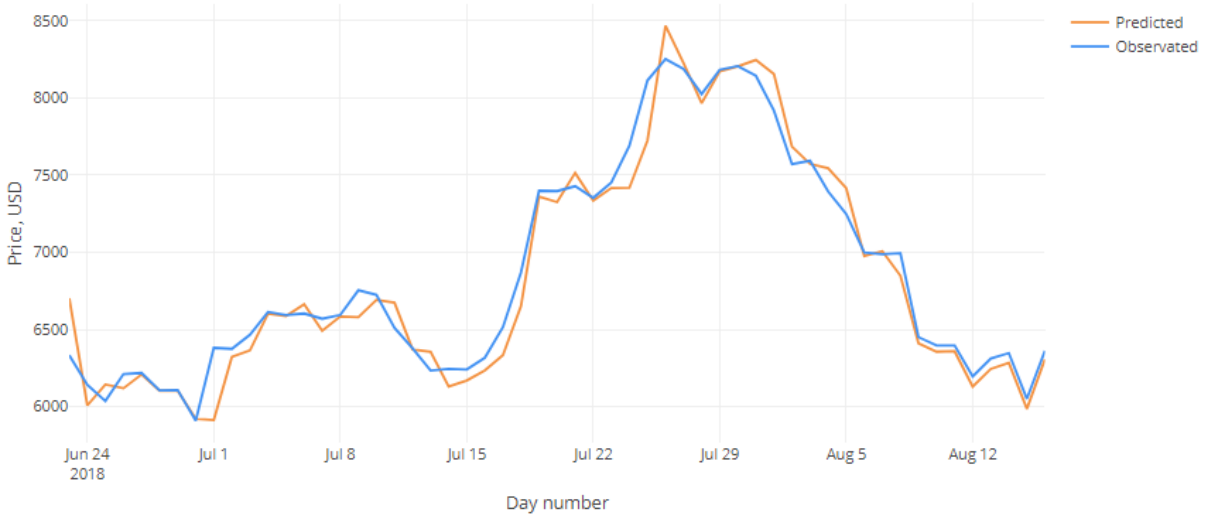

Gráfico 19. Muestra de las observaciones y predicciones de los datos de test [36].

5.4.2 Red neuronal artificial

Se implementa una red neuronal, es decir, un modelo computacional que trata de imitar el funcionamiento del cerebro humano aprendiendo a partir de la experiencia. Es una técnica de aprendizaje supervisado dentro del campo de la inteligencia artificial. En el libro [40] se puede consultar desde una perspectiva de ingeniería el libro más completo de redes neuronales.

Se implementa utilizando los módulos de *tensorflow*, *keras* y *sklearn.* Primero de todo se crea un conjunto de datos de entrenamiento y un conjunto de datos de evaluación. El modelo de la red neuronal aprende con los datos de entrenamiento y después evalúa en el conjunto de datos de evaluación. Normalmente se entrena con los datos de un periodo de tiempo y después se evalúa con datos de un periodo distinto.

El modelo más simple es establecer el precio de mañana igual al de hoy. Sin embargo, los modelos que realizan las predicciones en un punto a futuro son demasiado precisos debido a que los errores no se trasladan a las predicciones posteriores, sino que en cada punto del tiempo se restablece el error.

Por este motivo se va realizar predicciones multipunto con ventanas temporales. De esta manera, los errores de las predicciones anteriores no se restablecen, sino que se combinan con predicciones posteriores. Se utilizan los mismos datos que se han utilizado para realizar la regresión lineal.

Se utiliza un Dataframe donde el conjunto de datos esta ordenado por fecha, sin embargo, para la entrada de datos se elimina el campo de la fecha ya que no será necesario como entrada para el modelo. En la **Tabla *9*** se muestra los datos de entrada.

Date	price_btc	total_number_btc	market_capitalization_btc	address_btc	exchange_trade_btc	transactions_btc	hash_rate_btc	difficulty_btc	miners_revenu
2016-01-01	429.340000	15028600.0	6.452379e+09	310331.0	2.854093e+07	164132.0	7.436044e+05	1.038803e+11	1.55721
2016-01-02	432.330000	15031975.0	6.498774e+09	232199.0	2.040696e+07	123623.0	6.971292e+05	1.038803e+11	1.46732
2016-01-03	433.940000	15035400.0	6.524461e+09	334703.0	1.506949e+07	142904.0	7.074570e+05	1.038803e+11	1.49882
2016-01-04	428.130000	15039125.0	6.438701e+09	295177.0	2.332215e+07	141064.0	7.694240e+05	1.038803e+11	1.60463

Tabla 9. Muestra de los datos de entrenamiento [36].

El modelo utilizará los datos previos de los indicadores de *Bitcoin* para predecir el precio medio del día siguiente. El modelo se ha definido una ventana temporal de diez días, es decir, que para predecir un día concreto utilizará datos previos de los diez días anteriores.

Con lo cual, se define una estructura de datos como entrada para el modelo. Se debe tener el tamaño de la ventana ya que si se elige una ventana temporal pequeña es posible que el modelo no pueda tener suficiente información para detectar comportamientos complejos a largo plazo. Además, la entrada de datos del modelo se debe de normalizar puesto que las redes neuronales trabajan con valores normalizados. En el **Código *15*** se crea la estructura de datos de ventanas temporales y se normaliza los datos para la entrada de la red neuronal.

```
#Variables explicativas
LSTM_training_inputs = []

for i in range(len(df_train_x)-window_len):
    temp_set = df_train_x[i:(i+window_len)].copy()
    for j in mesures_name:
        temp_set.loc[:, j] = temp_set[j]/temp_set[j].iloc[0] - 1
    LSTM_training_inputs.append(temp_set)

#Variable explicada
LSTM_training_outputs = (df_train_y[window_len:].values/df_train_y[:-
window_len].values)-1
```

Código 15. Creación y normalización de la estructura de datos de entrada [36].

Esta función ofrece como resultado un vector de longitud k, donde cada uno de sus elementos es una matriz $n \cdot m$ elementos. En este caso n es el número de variables y m el número de días de la ventana temporal empleada. Esta estructura se transforma a una estructura matricial para obtener un vector de 3 dimensiones k, n, m.

En el **Código *16*** se implementa el modelo de tipo red neuronal recurrente *LSTM.*

```
def build_model(inputs, output_size, neurons, activ_func="linear",
                dropout=0.25, loss="mean_squared_error",
      optimizer="adam"):
    model = Sequential()

    model.add(LSTM(neurons, input_shape=(inputs.shape[1],
inputs.shape[2])))
    model.add(Dropout(dropout))
    model.add(Dense(units=output_size))
    model.add(Activation(activ_func))

    model.compile(loss=loss, optimizer=optimizer)
    return model
```

Código 16. Función de construcción del modelo LSTM [36].

La definición de la red neuronal tiene varios parámetros. Se pueden consultar los detalles en la documentación del módulo *keras* [41]. Se trata de una *API* de redes neuronales de alto nivel desarrollada para permitir realizar implementaciones de forma ágil. Admite redes recurrentes, convolucionales y además se puede ejecutar tanto con CPU como con GPU. En el **Código *17*** se muestra como se inicializa y se entrena el modelo.

```
# Se inicializa el modelo
model_btc = build_model(LSTM_training_inputs, output_size=1,neurons = 64)

# Comprobar el tiempo
start_time = time()

#Se entrena al modelo
model_btc_history = model_btc.fit(LSTM_training_inputs,
    LSTM_training_outputs, epochs=100,
    batch_size=2, verbose=2, shuffle=False,
    callbacks = [EarlyStopping(monitor='val_loss',
        min_delta=5e-5, patience=20, verbose=1)])

# Comprobar el tiempo
final_time = time() - start_time
```

Código 17. Se inicializa y se entrena el modelo [36].

En el **Gráfico *20*** se grafica el error de la red en cada iteración. Se observa que el error disminuye con cada iteración, lo que indica que la red está aprendiendo. El *MAE* obtenido en la gráfica es sobre los datos normalizados que se utilizan para entrenar la red. Para realizar la comparativa se calcula el *MAE* con los datos re-escalados.

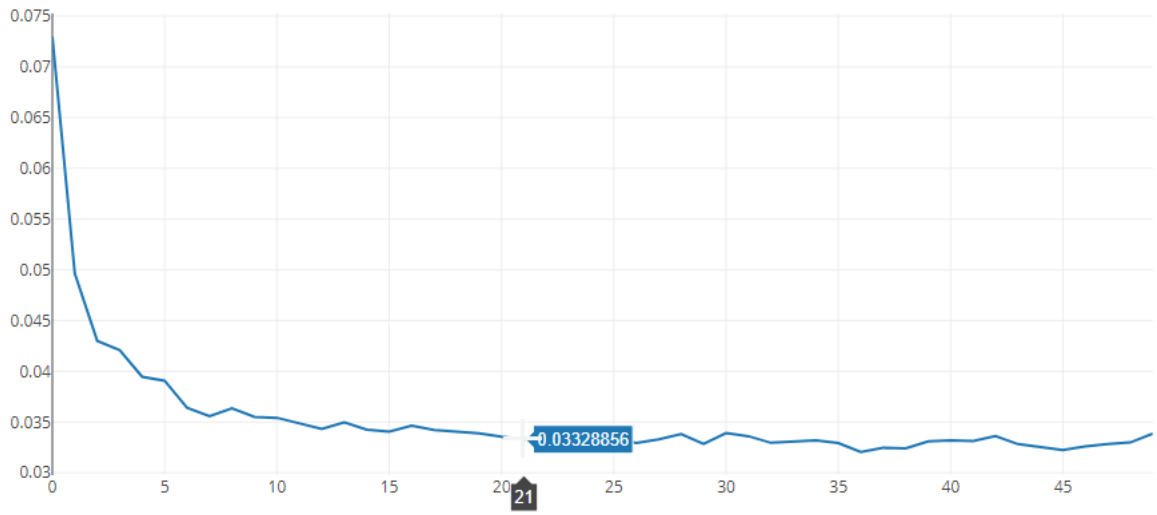

Gráfico 20. *Mean Absolute Error (MAE)* de cada interacción de la red neuronal [36].

En el **Gráfico *21*** se puede ver el resultado del modelo. Se podría llegar a ajustar más el modelo minimizando el error introduciendo más neuronas y más interacciones. Sin embargo, se podría llegar a obtener un problema de sobreajuste. El error *MAE* de los datos de entrenamiento es de 134.48. Se puede observar un desfase entre las observaciones y las predicciones.

Gráfico 21. Muestra de las observaciones y predicciones de los datos de entrenamiento [36].

En la **Gráfico *22*** muestra el resultado de la predicción de los datos de evaluación. El modelo se ha entrenado con grandes fluctuaciones repentinas del precio de *Bitcoin* es por eso que en él se puede observar esta tendencia en la predicción de los datos de evaluación. El error *MAE* obtenido es de 259,87.

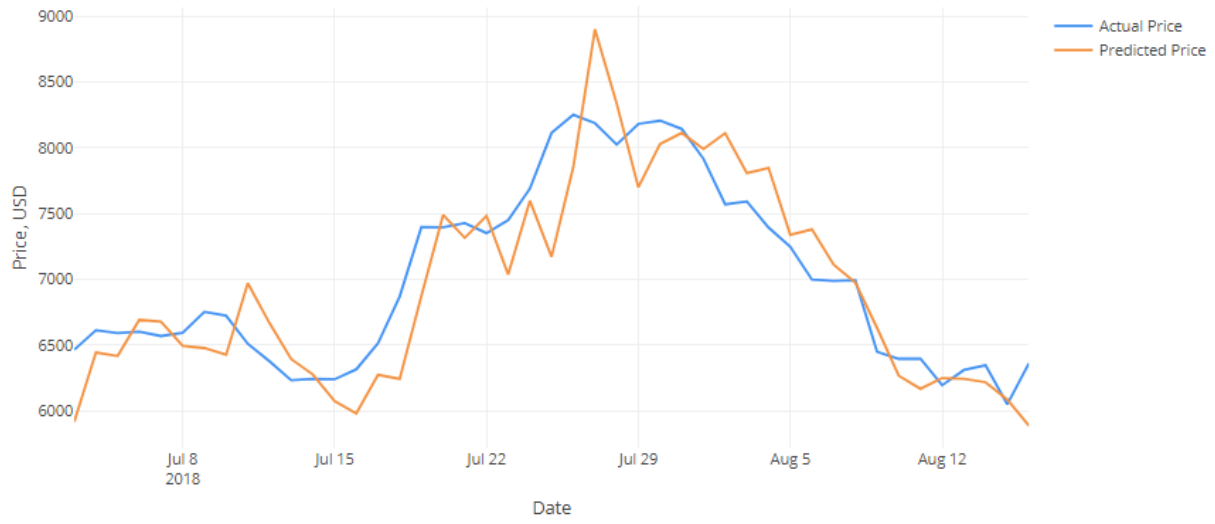

Gráfico 22. Muestra de las observaciones y predicciones de los datos de evaluación [36].

Si se observa con detalle los gráficos anteriores, el **Gráfico *21*** y el **Gráfico *22*** se puede observar un cierto patrón o tendencia. Parece ser que los valores predichos reflejan los valores reales previos ligeramente modificados. Los modelos de red neuronal parece ser que han seguido un estrategia de aproximación a un modelo auto regresivo donde los valores futuros son la suma ponderada de los n valores anteriores. Para confirmar esta hipótesis se realiza el siguiente análisis.

En el **Gráfico *23*** se visualiza el porcentaje de cambio de los valores observados y los valores predichos. Es decir, se realiza el porcentaje de cambio entre un valor actual y un anterior para todos los valores de las observaciones y de los valores predichos.

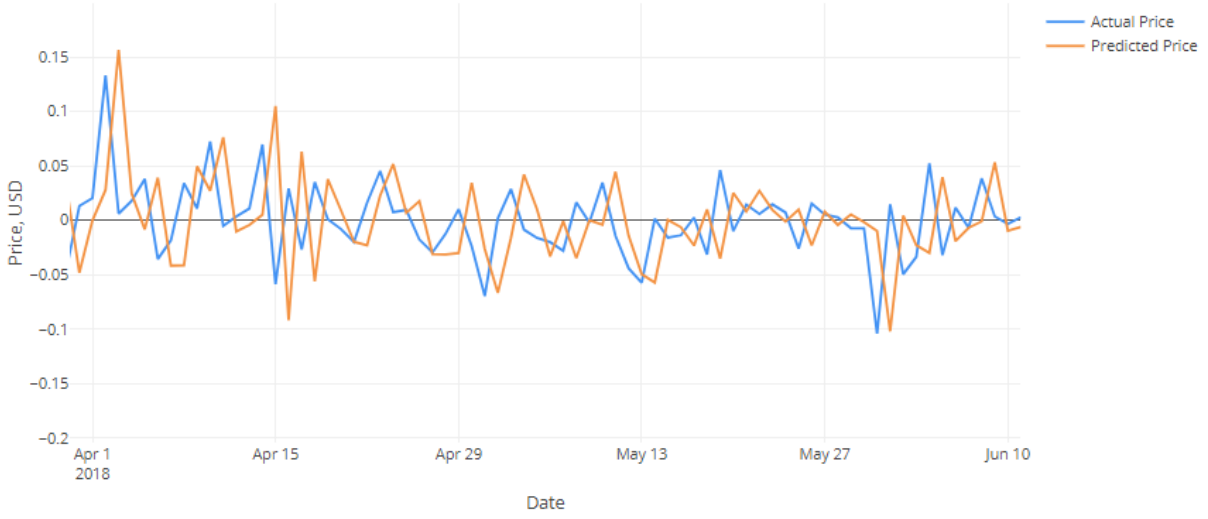

Gráfico 23. Muestra del porcentaje de cambio de las observaciones y de los valores calculados [36].

En el **Gráfico *24*** se observa que si se trasladan el conjunto de valores predichos un día hacía atrás prácticamente se solapan. Se observa la salida del precio previsto es equivalente al precio real del día anterior más una ligera modificación.

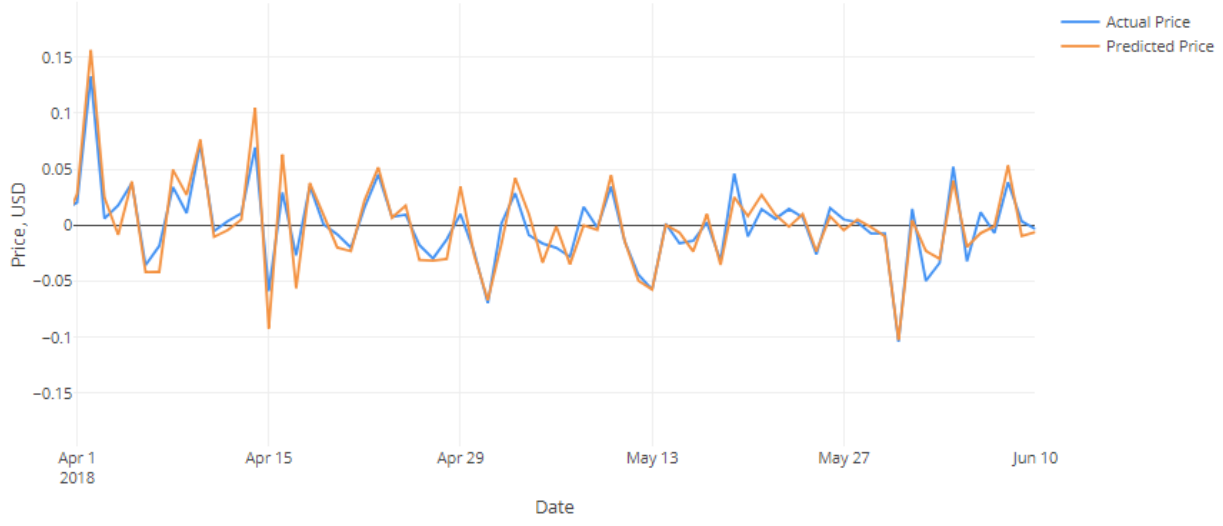

Gráfico 24. Muestra del porcentaje de cambio de las observaciones y de los valores calculados trasladados un día hacia atrás [36].

En **Gráfico *25*** se observa una correlación entre el porcentaje de cambio de las observaciones y las predicciones sin desplazamiento es de 0,83. Mientras que si se realiza la correlación del porcentaje de cambio entre los datos observados y los datos predichos trasladados en

conjunto un día anterior se obtiene como resultado una correlación de 0,14. Los datos son significativos y se confirma la hipótesis de que la salida predicha de la red utiliza el valor del día anterior más una modificación.

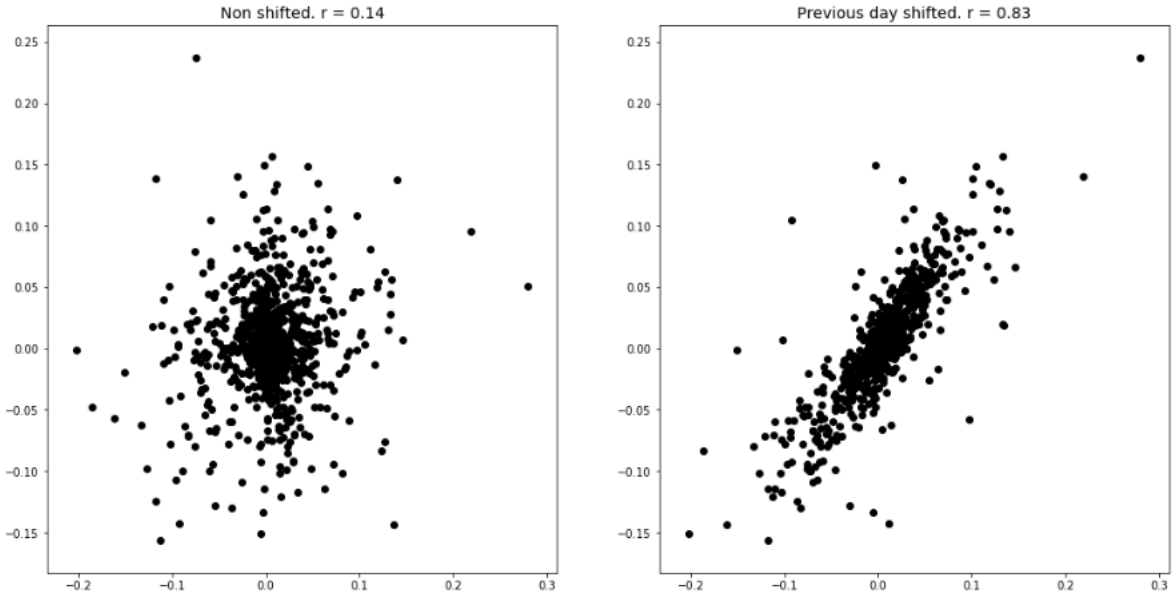

Gráfico 25. Comparativa de la dispersión entre porcentaje de cambio sin y con translación [36].

5.5 Resultados

A continuación, se comparan los resultados obtenidos de los diferentes modelos implementados. Los datos utilizados para cada modelo han sido las mismos. El periodo de tiempo ha sido del 1 de enero del 2016 hasta el 6 de junio del 2018 para los datos de entrenamiento y del 23 de junio del 2018 hasta el 16 de agosto del 2018 para los datos de evaluación.

Se ha aplicado un modelo de regresión múltiple *Ordinary Least Squares (OLS)* obteniendo como resultado un error *MAE* de 11,01 en los datos de entrenamiento y un error en los datos de prueba 91,79. Por otra parte, se ha aplicado un modelo de red neuronal *Long Short Term Memory (LSTM)* obteniendo como resultado un *MAE* de 133,31 en los datos de entrenamiento y un error de 259,87 en los datos de prueba.

Con el objetivo de implementar modelos de redes neuronales más precisos se han realizado otras parametrizaciones, modificando el tipo de red neuronal, aumentando el número de neuronas, aumentando el número de iteraciones, modificando la función de activación y modificando la ventana temporal. Sin embargo, en todas las demás parametrizaciones y los

nuevos modelos creados ninguno consigue mejorar el resultado obtenido de la regresión. La parametrización utilizada para generar los resultados se puede observar en la ***Tabla 10***. Para cada modelo cambia el parámetro de tipo de red neuronal. La primera de tipo *Long Short Term Memory* (*LSTM*) de una capa, la segunda *LSTM2* con dos capas *LSTM* conectadas de forma secuencial y la tercera de tipo *Gate Recurrent Unit (GRU)*.

Nombre del parámetro	Parameters	Valor del parámetro
Tipo de red neuronal	type of neuronal network	LSTM
Número de neuronas	neurons	64
Función de activación	activation_function	linear
Optimizador	optimizer	adam
Parámetro de abandono	dropout	0.25
Función de optimización	loss	mean_squared_error
datos de entrada del modelo	inputs	input_data
tamaño de la salida	output_size	1

Tabla 10. Parametrización de la red neuronal [36].

En la **Tabla *11*** se pueden ver el resultados obtenidos para cada modelo y se observa que el mejor resultado ha sido obtenido por la regresión lineal múltiple tanto para los datos de entrenamiento como para los datos de evaluación.

	Técnica	*MAE* de los datos de entrenamiento	*MAE* de los datos de evaluación
OLS	Regresión m	11,01	91,79
LSTM	Red neuronal	133,31	259,87
LSTM2	Red neuronal	119,44	295,11
GRU	Red neuronal	107,17	290,89

Tabla 11. Comparativa de los resultados obtenidos [36].

En comparación, el método de regresión múltiple crea el mejor modelo ya que el error ha sido menor. Sin embargo, este modelo se adapta muy bien para los datos de entrenamiento, pero para los datos de test el error aumenta considerablemente. Hay que tener en cuenta el valor de R^2 es muy alto y podría ser un indicador de sobreajuste del modelo ya que se ajusta correctamente para los datos de entrenamiento. Este hecho podría ser debido a que los datos de entrenamiento tienen variables altamente correlacionadas como por ejemplo la capitalización de *Bitcoin* y el precio medio de *Bitcoin*.

Los resultados de las redes neuronales han obtenido un error más alto, sin embargo, es posible que cambiando las parametrizaciones de los modelos de redes neuronales se pueda obtener un mejor resultado reduciendo el error, pero posiblemente aumentando el sobreajuste. Se observa que como los datos de entrenamiento tiene grandes fluctuaciones la predicción en los datos de evaluación heredan esta tendencia. Al comparar los errores obtenidos de los tres modelos neuronales se puede ver un posible problema de sobreajuste. La red con mayor error en los datos de entrenamiento es *LSTM*, sin embargo, con los datos de evaluación obtiene el menor error que *GRU* y *LSTM2*.

Se ha demostrado que, en la salida del modelo de la red neuronal, la línea de predicción va por detrás de la observación. Este comportamiento es debido para la predicción un día particular, el modelo utiliza el valor del día anterior o de los días anteriores de la ventana temporal. Es decir, que, para la predicción del día de mañana, la red neuronal extrae como salida una versión modificada del valor del precio del día anterior o de los días anteriores. O dicho de otro modo el modelo aprende en base al precio del día anterior o en base a la ventana anterior.

6 Conclusiones y trabajo futuro

La problemática consiste en la estimación del precio de las criptomonedas con el objetivo de obtener una mayor rentabilidad. La solución adoptada es implementar modelos predictivos y comparar los resultados obtenidos. En el primer modelo se utiliza la técnica de regresión múltiple *Ordinary Least Squares (OLS).* En el segundo modelo se emplea la técnica de aprendizaje automático, concretamente, la red neuronal. Se implementan tres redes neuronales con distintas configuraciones y con diferentes capas. La primera de tipo *Long Short Term Memory* (*LSTM*) de una capa, la segunda *LSTM2* con dos capas *LSTM* conectadas de forma secuencial y la tercera de tipo *Gate Recurrent Unit (GRU)*.

6.1 Conclusiones

A continuación, se enumeran las conclusiones respecto a los objetivos planteados.

1. Se ha realizado una **búsqueda y se ha elegido una tecnología idónea** que es *Python* sobre la platadorma *Anaconda* para para implementar las funciones, métodos y algoritmos y permitir realizar el análisis, transformación y visualización de los datos.
2. Se ha logrado **analizar los datos de diferentes fuentes e identificar los datos útiles**. Se han identificado las diferentes fuentes de datos de criptomonedas, se han seleccionado las APIs ha entendido la forma de obtener los datos e implementado funciones para la correcta extracción, se han seleccionado la información relevante y se han almacenado los datos.
3. Se han **analizado y visualizado los indicadores *Bitcoin*** mediante métodos estadísticos. Se han definido funciones para realizar visualización de los indicadores de *Bitcoin* y se ha realizado una correlación entre *Bitcoin* y las criptomonedas con mayor capitalización del mercado para confirmar la hipótesis de que existe relación entre el precio de las criptomonedas y se relacionan cada vez más unas con otras a medida que transcurre el tiempo.
4. Se ha logrado **implementar dos modelos predictivos empleando técnicas de inteligencia artificial.** Concretamente una regresión lineal multiple de *Ordinary Least Squares (OLS)* y tres redes neuronales**,** la primera de tipo *Long Short Term Memory* (*LSTM*) de una capa, la segunda LSTM2 con dos capas *LSTM* conectadas de forma secuencial y la tercera de tipo *Gate Recurrent Unit (GRU)*.

5. Se han **evaluado y comparado los resultados obtenidos** de los modelos predictivos mediante el *Mean Absolute Error (MAE)*. Los resultados demuestran que el modelo de la regresión múltiple *Ordinary Least Squares (OLS)* obtiene un mejor resultado que los modelos predictivos implementados con técnicas de aprendizaje automático de redes neuronales.

Se ha **establecido una comparativa entre dos técnicas de predicción de la evolución del precio de las criptomonedas: la regresión lineal múltiple y la red neuronal**. Por último, destacar que el objetivo general y los objetivos específicos han sido alcanzados.

6.2 Futuras líneas de trabajo

A continuación, se proponen distintas ideas para continuar por la línea de investigación generada con este proyecto. Una propuesta de ampliación sería el análisis y la incorporación de más indicadores relevantes que puedan afectar al precio de las criptomonedas. Así como realizar correlaciones con otros indicadores como, por ejemplo, las búsquedas en google sobre una criptomoneda en particular, el precio de las fuentes de energía, el precio de diferentes productos de los mercados financieros, entre otros.

Otra idea sería incluir estos indicadores en como datos de entrada de los modelos predictivos. La comparación entre diferentes modelos utilizando diferentes técnicas y la comparación entre diversas configuraciones de dichos modelos para identificar parametrizaciones óptimas.

Otra idea sería analizar los componentes del precio como la estacionalidad, las tendencias o las componentes estacionarias para después implementar los modelos predictivos. Un ejemplo aplicable sería aplicar algoritmos y librerías como es el caso de *prophet* una librería de *facebook* para la predicción de series temporales y análisis de estacionalidad múltiple con crecimiento lineal o no lineal.

Por último, la predicción indirecta del comportamiento humano de las personas dispuestas a invertir. Es decir, es conocido que la demanda afecta directamente al precio, por lo que si se conoce la demanda, se puede predecir el precio. Para conocer la demanda, se propone realizar un análisis de sentimiento de las noticias publicadas y su impacto en el precio de las criptomonedas. Es decir, capturar las noticias de diferentes fuentes periodísticas e informativas relevantes y realizar un análisis del texto para intentar averiguar qué fuentes son más influyentes. Además de dotar a cada fuente con un peso particular en función de su influencia global o local.

7 Referencias y enlaces

[1] S. Nakamoto, «Bitcoin: A Peer-to-Peer Electronic Cash System», 2009. [En línea]. Disponible en: www.bitcoin.org. [Accedido: 28-abr-2018].

[2] A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, y J. Clark, «Bitcoin and Cryptocurrency Technologies», 2016. [En línea]. Disponible en: https://www.uio.no/studier/emner/matnat/ifi/IN5420/v18/timeplan/resources/bitcoin-and-cryptocurrency-techniques.pdf. [Accedido: 28-abr-2018].

[3] C. Dwork y M. Naor, «Princing via Processing or Combatting Junk Mail», *CRYPTO 92*, 1992. [En línea]. Disponible en: http://www.hashcash.org/papers/pvp.pdf. [Accedido: 30-jun-2018].

[4] A. Back, «Hashcash -A Denial of Service Counter-Measure», 2002. [En línea]. Disponible en: http://www.hashcash.org/papers/hashcash.pdf. [Accedido: 30-jun-2018].

[5] S. Haber y W. S. Stornetta, «How to Time-Stamp a Digital Document», 1991. [En línea]. Disponible en: https://www.anf.es/pdf/Haber_Stornetta.pdf. [Accedido: 30-jun-2018].

[6] R. C. Merkle, «A digital signature based on a conventional encryption function», 2000. [En línea]. Disponible en: https://link.springer.com/content/pdf/10.1007%2F3-540-48184-2_32.pdf. [Accedido: 01-jul-2018].

[7] D. Johnson y A. Menezes, «The Elliptic Curve Digital Signature Algorithm (ECDSA) 1 2», 1999. [En línea]. Disponible en: http://www.cacr.math.uwaterloo.ca. [Accedido: 30-jun-2018].

[8] Z. Trifa y M. Khemakhem, «Sybil nodes as a mitigation strategy against sybil attack», *Procedia Computer Science*, 2014. [En línea]. Disponible en: http://dx.doi.org/10.1016/j.procs.2014.05.544.

[9] «Página web de servicio de intercambio de bitcoins. Coinbase.com». [En línea]. Disponible en: https://www.coinbase.com/.

[10] «Página web de servicio de intercambio de bitcoins. Blockchain.com». [En línea]. Disponible en: https://www.blockchain.com/. [Accedido: 08-jul-2018].

[11] «Página web de servicio de intercambio de bitcoins. Bitcoincharts.com». [En línea]. Disponible en: https://bitcoincharts.com/. [Accedido: 09-jul-2018].

[12] «Página web de servicio de intercambio de bitcoins. LocalBitcoins.com». [En línea].

Disponible en: https://localbitcoins.com/. [Accedido: 09-jul-2018].

[13] «Página web de servicio de intercambio de bitcoins. Bitcoin.org». [En línea]. Disponible en: https://bitcoin.org/en/exchanges#international. [Accedido: 09-jul-2018].

[14] «Cryptocurrency Market Capitalizations | CoinMarketCap». [En línea]. Disponible en: https://coinmarketcap.com/. [Accedido: 21-may-2018].

[15] V. Buterin, «A Next-Generation Smart Contract and Decentralized Application Platform», 2013. [En línea]. Disponible en: http://blockchainlab.com/pdf/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf. [Accedido: 11-jul-2018].

[16] F. Flamur Bunjaku, Lamur, O. Gjorgieva-Trajkovska2, y E. Miteva-Kacarski, «Cryptocurrencies - advantages and disadvantages», *Journal of Economics*, dic-2017. [En línea]. Disponible en: http://js.ugd.edu.mk/index.php/JE/article/view/1933. [Accedido: 24-jun-2018].

[17] Ivashchenko A. I., «Using Cryptocurrency in the Activities of Ukrainian Small and Medium Enterprises in order to Improve their Investment Attractiveness», *Problems of economy*, 2016. [En línea]. Disponible en: http://oaji.net/articles/2016/728-1479730699.pdf.

[18] S. Meiklejohn *et al.*, «A Fistful of Bitcoins: Characterizing Payments Among Men with No Names», 2013. [En línea]. Disponible en: http://dx.doi.org/10.1145/2504730.2504747. [Accedido: 15-jul-2018].

[19] G. Vora, «Cryptocurrencies: Are Disruptive Financial Innovations Here?», 2009. [En línea]. Disponible en: http://www.scirp.org/journal/mehttp://dx.doi.org/10.4236/me.2015.67077http://dx.doi.org/10.4236/me.2015.67077http://creativecommons.org/licenses/by/4.0/. [Accedido: 16-jul-2018].

[20] J. Pratt, P. Bednar, C. Kwon, «What drives Bitcoin price?», *What drives Bitcoin price?*, 2016. [En línea]. Disponible en: https://schoolnutrition.org/5--News-and-Publications/4--The-Journal-of-Child-Nutrition-and-Management/Fall-2012/Volume-36,-Issue-2,-Fall-2012---Pratt,-Bednar,-Kwon/.

[21] L. Kristoufek, «What Are the Main Drivers of the Bitcoin Price? Evidence from Wavelet Coherence Analysis», *PLOS ONE*, 15-abr-2015. [En línea]. Disponible en: http://dx.plos.org/10.1371/journal.pone.0123923. [Accedido: 25-jun-2018].

[22] P. Ciaian, M. Rajcaniova, y A. Kancs, «The economics of BitCoin price formation», 2015. [En línea]. Disponible en: http://www.tandfonline.com/action/journalInformation?journalCode=raec20. [Accedido: 25-jun-2018].

[23] O. Snihovyi, O. Ivanov, y V. Kobets, «Cryptocurrencies Prices Forecasting With Anaconda Tool Using Machine Learning Techniques», 2018. [En línea]. Disponible en: https://www.researchgate.net/profile/Vitaliy_Kobets/publication/325454476_Cryptocurrencies_Prices_Forecasting_With_Anaconda_Tool_Using_Machine_Learning_Techniques/links/5b0edf93a6fdcc80995bab21/Cryptocurrencies-Prices-Forecasting-With-Anaconda-Tool-Using-.

[24] M. Polasik, A. Piotrowska, y T. P. Wisniewski, «Price Fluctuations and the Use of Bitcoin : An Empirical Inquiry», 2015. [En línea]. Disponible en: https://pdfs.semanticscholar.org/f711/087cf35409a51f1485dcecad043579c78831.pdf.

[25] S. Colianni, S. Rosales, y M. Signorotti, «Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis». [En línea]. Disponible en: http://cs229.stanford.edu/proj2015/029_report.pdf. [Accedido: 04-ago-2018].

[26] L. Catania, S. Grassi, y F. Ravazzolo, «Forecasting Cryptocurrencies Financial Time Series», 2018. [En línea]. Disponible en: https://brage.bibsys.no/xmlui/bitstream/handle/11250/2489408/WP_CAMP_5_2018.pdf?sequence=1&isAllowed=y. [Accedido: 05-ago-2018].

[27] E. Bouri, G. Azzi, y A. H. Dyhrberg, «On the return-volatility relationship in the Bitcoin market around the price crash of 2013». [En línea]. Disponible en: https://ssrn.com/abstract=2869855. [Accedido: 25-jun-2018].

[28] A. Hayes, «What Factors Give Cryptocurrencies Their Value: An Empirical Analysis», *SSRN Electronic Journal*, 16-mar-2015. [En línea]. Disponible en: https://www.ssrn.com/abstract=2579445. [Accedido: 25-jun-2018].

[29] J. Kurkä, «Do Cryptocurrencies and Traditional Asset Classes Influence Each Other?», 2017. [En línea]. Disponible en: https://www.econstor.eu/bitstream/10419/174222/1/wp_2017_29_kurka.pdf. [Accedido: 03-ago-2018].

[30] I. D. Raheem, K. O. Isah, y A. A. Adedeji, «The Hidden Predictive Power of Cryptocurrencies: Evidence from US Stock Market», *Economic Change and Restructuring*, 2018. [En línea]. Disponible en: https://www.researchgate.net/profile/Kazeem_Isah2/publication/325346540_The_Hidd

en_Predictive_Power_of_Cryptocurrencies_Evidence_from_US_Stock_Market_Kazeem_O_Isah_and_Ibrahim_Raheem/links/5b06f076aca2725783dabb36/The-Hidden-Predictive-Power-of-Cryptocurr.

[31] A. ElBahrawy, L. Alessandretti, A. Kandler, R. Pastor-Satorras, y A. Baronchelli, «Evolutionary dynamics of the cryptocurrency market», *Royal Society Open Science*, 01-nov-2017. [En línea]. Disponible en: http://rsos.royalsocietypublishing.org/lookup/doi/10.1098/rsos.170623. [Accedido: 15-may-2018].

[32] «Página web de Python. Python.org». [En línea]. Disponible en: https://www.python.org/. [Accedido: 09-ago-2018].

[33] «Página web de Anaconda. Anaconda.org». [En línea]. Disponible en: https://anaconda.org/. [Accedido: 09-ago-2018].

[34] G. van Rossum, «Página web personal». [En línea]. Disponible en: https://gvanrossum.github.io/. [Accedido: 09-ago-2018].

[35] G. van Rossum, «King's Day Speech - Python: a programming language created by a community». [En línea]. Disponible en: http://neopythonic.blogspot.com/2016/04/kings-day-speech.html. [Accedido: 09-ago-2018].

[36] C. G. Díaz, «Github de Cristian García Díaz». [En línea]. Disponible en: https://github.com/cgarciadiaz/cryptocurrency_analysis. [Accedido: 14-abr-2018].

[37] «Página web de R. r-project.org/». [En línea]. Disponible en: https://www.r-project.org/. [Accedido: 09-ago-2018].

[38] P. Triest, «Analyzing Cryptocurrency Markets Using Python». [En línea]. Disponible en: https://blog.patricktriest.com/analyzing-cryptocurrencies-python/. [Accedido: 21-may-2018].

[39] Patrick Triest, «Github de Patrick Triest». [En línea]. Disponible en: https://github.com/triestpa/Cryptocurrency-Analysis-Python. [Accedido: 14-abr-2018].

[40] S. Haykin *et al.*, *Neural Networks and Learning Machines Third Edition*. Pearson, 2009.

[41] «Keras documentation». [En línea]. Disponible en: https://keras.io/layers/recurrent/.