

WEBIR 2019

Automatización de cálculo de triángulos en gráficos de
vela de criptomonedas

Guillermo Aguirre (4.817.028-5), Julieta Cavalli (4.670.993-9),
Elizabeth Lofredo (4.677.501-9)

*Facultad de Ingeniería. Universidad de la República
Montevideo, Uruguay*

Índice

1. Introducción	3
2. Marco Teórico	4
3. Problema	8
4. Diseño e implementación	9
4.1. Etapas de implementación	9
4.2. Herramientas	11
4.3. Arquitectura	12
5. Trabajo a futuro	13
6. Conclusiones	14
Referencias	15

1. Introducción

El objetivo principal de este trabajo es alcanzar una solución para la problemática del análisis de criptomonedas aplicando las técnicas aprendidas durante el curso. En este informe se verá reflejado el proceso para el alcance de dicha solución así como también la metodología utilizada y los resultados obtenidos.

El presente informe se organiza de la siguiente manera: En la sección 2 se encuentran definidos conceptos necesarios para el entendimiento del trabajo. Dicha sección contiene las definiciones de criptomoneda, gráfico de velas, entre otras. En la sección 3 se describe la problemática a abordar así como la motivación por resolverla. En la sección 4 se encuentran detallados los procesos de diseño e implementación. Dicha sección incluye las cuatro etapas diferenciadas en la implementación, las herramientas utilizadas y la arquitectura del sistema. En la sección 5 se detalla el trabajo posible a realizar en el futuro para seguir expandiendo y mejorando la solución alcanzada. Finalmente en la sección 6 se encuentran las conclusiones generales del trabajo.

2. Marco Teórico

Para entender el problema a abordar se deberán definir en primer lugar algunos conceptos.

La **criptomoneda** no tiene una definición oficial. Sin embargo se pueden encontrar diversas definiciones como las presentadas a continuación:

”La criptomoneda es una moneda digital producida por una red pública en lugar de cualquier gobierno, que utiliza la criptografía para asegurar que los pagos se envían y reciben de forma segura.”

”La criptomoneda es una forma de moneda basada únicamente en las matemáticas. En lugar de la moneda fiduciaria, que se imprime, una criptomoneda se produce mediante la resolución de problemas matemáticos basados en criptografía.”

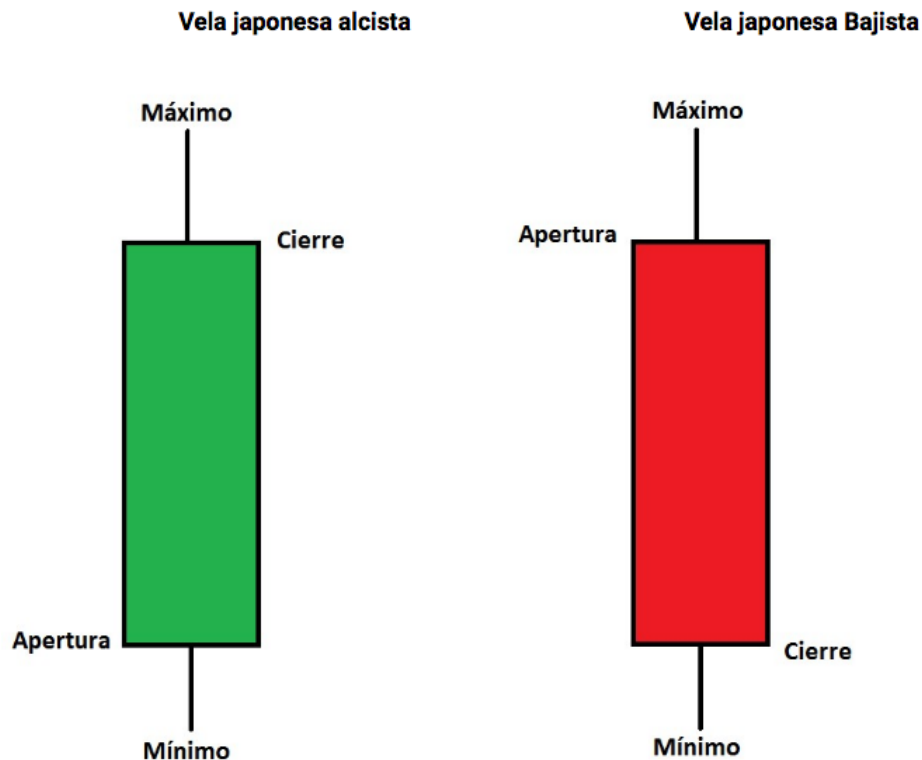
Una de las bases que presenta esta tecnología, es que las criptomonedas están respaldadas por las matemáticas, lo que hace que no pueda ajustarse por un solo grupo o persona. Las criptomonedas no están vinculadas a la disponibilidad de los bienes físicos como, como por ejemplo el oro.

En su origen las criptomonedas utilizan una red distribuida para permitir el P2P – Peer-to-Peer, un sistema de verificación de transacciones sin necesidad de terceros. Con el fin de mantener esta seguridad, las criptomonedas utilizan algoritmos matemáticos y un registro de contabilidad público (llamado blockchain) para así, asegurar que cada transacción que se realiza sea legítima y evitar el fraude.

Para visualizar la variación del valor de una criptomoneda en el tiempo existen muchos tipos de gráficas. Este trabajo se enfocará en las **velas japonesas** o **Candlesticks**, estas son una forma de representación gráfica de la información esencial de la cotización de un activo financiero en un determinado período de tiempo. Son una alternativa a la representación gráfica habitual en forma de línea. Las velas japonesas, a diferencia de los gráficos en forma de línea, nos informan, en general, de cuatro parámetros (precio de apertura, de cierre, el máximo y el mínimo). Además, las velas japonesas pueden ser alcistas, bajistas o neutrales.

- **Alcistas:** si el precio de cierre es superior al precio de apertura.

- **Bajistas:** si el precio de cierre es inferior al precio de apertura.
- **Neutrales:** si el precio de apertura es igual al precio de cierre.



¿Cómo interpretar las velas japonesas? Podemos encontrar velas japonesas con cuerpos en blanco y negro, pero también podemos encontrar gráficos de color verde y rojo, ya que son colores que definen la dirección del mercado, verde positivo (alcista) y rojo negativo (bajista). Para esta tarea utilizaremos verde y rojo.

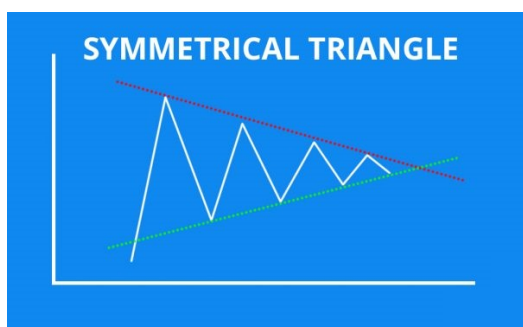
Por otro lado, las líneas que sobresalen del cuerpo de las velas se denominan **sombras**. El punto superior de la sombra, es el precio más alto de este período, mientras que el punto inferior de la sombra, es el precio más bajo alcanzado en este período.



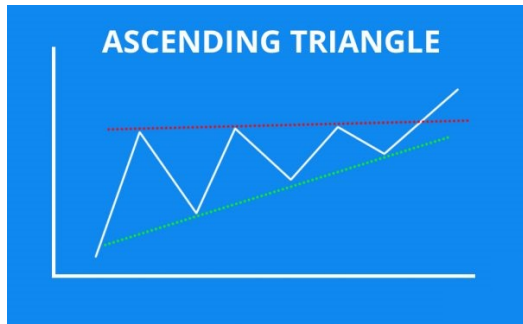
En el caso de que no haya sombra encima del cuerpo de la vela, significa que el precio de cierre (en período verde) o el precio de apertura (en período rojo) es igual al precio más alto. Si no hay sombra en el extremo inferior del cuerpo de la vela, supone que el precio de apertura (período verde) o el precio de cierre (período rojo) es igual al precio más bajo del periodo.

Para realizar un estudio del modelo de continuidad sobre las gráficas es necesario introducir el concepto de **patrones de triángulos**. Estos son conocidos como un patrón bilateral, lo que significa que después de un punto de quiebre la tendencia puede tanto continuar como revertirse. Existen básicamente tres tipos de triángulos:

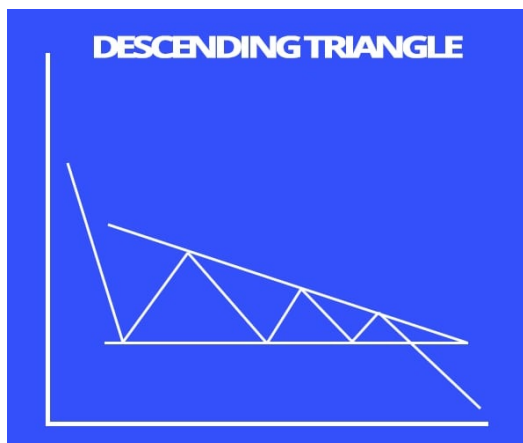
- **Simétrico** (el precio está contenido por dos líneas de tendencia convergentes con una pendiente similar)



- **Ascendiente** (el precio está contenido por una línea de tendencia horizontal que actúa de resistencia y una línea de tendencia ascendente que actúa de soporte)



- **Descendiente** (el precio está contenido por una línea de tendencia horizontal que actúa de soporte y una línea de tendencia descendente que actúa de resistencia)



La línea de resistencia es la línea que actúa de piso o techo, en los triángulos ascendentes y descendentes respectivamente. La línea de soporte es la línea que actúa de techo o piso, en los triángulos ascendentes y descendentes respectivamente.

3. Problema

La problemática a resolver es la toma de decisión al momento de la compra o venta de algún tipo de criptomoneda. La idea es tratar de facilitarle a un usuario inexperto la toma de la decisión mediante el uso de herramientas de estimación de tendencias sobre gráficos de criptomonedas.

Existen muchos cambios (exchanges) que trabajan con la compra y venta de las criptomonedas, el usuario puede ver el valor de las mismas y en algunos casos también proveen gráficas de su variación en el tiempo. Para visualizar la variación en el tiempo existen muchos tipos de gráficas, este trabajo se centrará en las gráficas conocidas como “Las velas japonesas” o “candlestick”, dicho nombre viene dado por la forma de la gráfica. Las gráficas provistas por los cambios son interactivas y dinámicas, dejándole al usuario dibujar líneas, figuras y cálculos sobre ellas. El usuario utilizando estas herramientas puede hacer un estudio del modelo de continuidad de la moneda, esto es la generación de triángulos que nos indican si es probable que el precio de esta criptomoneda suba, baje o se mantenga. Esto, para el usuario, es un componente importante a la hora de comprar o vender una moneda. El problema está en que todo esto se hace de forma manual, apostando a la habilidad del usuario en detectar y conseguir valor a través de esas herramientas. Un usuario inexperto frente a esto queda en una posición de desventaja.

Lo que proponemos como solución, es proveer al usuario el cálculo y sugerencia automáticamente, pudiendo éste decidir si le conviene comprar o vender, basándonos en la predicción de la tendencia del precio obtenida a partir del análisis con los triángulos.

4. Diseño e implementación

4.1. Etapas de implementación

Se puede dividir la implementación del proyecto en cuatro etapas:

1. Obtención de los datos de precios del cambio.
2. Representación de los datos mediante gráficas de velas
3. Calcular y graficar los triángulos
4. Renderizar vista HTML de las gráficas.

1 - Obtención de los datos de precios del cambio

Primero, se decidió reducir el alcance a un solo cambio para apuntar a una primera versión de la solución. Luego se buscó un cambio que previera los datos necesarios a partir de una API pública, sin la necesidad de scrapear explícitamente la información.

Se encontró la API pública de [Bitfinex](#) que provee al usuario de una API rest con los datos ya formateados para los gráficos de velas. Los datos poseen el siguiente formato: *open, close, high, low, mts*. Siendo éstos los precios a los que abrió, cerró, el máximo, el mínimo que logró y el tiempo en epoch ms¹ en que esto ocurrió. Se puede filtrar por moneda y rango de tiempo entre un dato y otro y la fecha de comienzo y fin de los datos.

Para conseguir los datos de la API se creó un script en Python, que mediante la librería `http` hace requests a la API y luego los datos que vienen en la respuesta se guardan en memoria para su procesamiento. Se consideró como una posible complicación el hecho de guardar todo en memoria, pero dado el alcance de este proyecto no resultó problemático. Este paso no representó mayores problemas de implementación dado que la API cuenta con una extensa documentación online que facilitó todo el proceso.

2 - Generación de gráfico de velas

Luego de la obtención de los datos con el formato acorde, el esfuerzo se concentró en graficarlos. En primer lugar se necesitó manipular los valores de

¹ <https://www.symantec.com/connect/articles/what-epoch-time-and-how-convert-human-understandable-format>

tiempo en epoch ms a valores legibles de fechas. Para nuestra suerte, la práctica de guardar datos de tiempo en epoch es una práctica ampliamente aplicada en la industria por lo tanto las funciones de manejo de fechas nativas de python ya manejan por defecto estas unidades.

Para la generación del gráfico se utilizó la librería de Python plotly² que presenta una gran capacidad de manipulación de datos y variedad de gráficas para generar. Entre las gráficas que abarca se encuentra el formato para gráficas de tipo candlestick charts³ que fue la utilizada.

3 - Calculando y graficando los triángulos

Una vez que se obtuvieron los datos en el formato correcto y se logró generar la gráfica de velas, se debió realizar el cálculo y la obtención de los triángulos, siendo esta etapa la etapa con mayor dificultad del proyecto.

Primero se tuvo que investigar y definir la forma de calcularlos. Para esto se utilizó fuertemente la información brindada por la página StockCharts en referencias. Se decidió tomar como puntos los máximos y mínimos locales de la gráfica y a partir de ellos, generar mediante una regresión lineal las líneas de tendencia correspondientes.

En la implementación se revisó para cada punto, el punto inmediatamente anterior y posterior. Para el caso de los mínimos, el punto tomado debía ser menor al punto anterior y al punto siguiente. De forma contraria para el caso de los máximos. La cantidad de puntos a considerar quedó determinada empíricamente, ya que en ningún lugar se explicitaba cuantos puntos máximos o mínimos se necesitaban. Se comenzó tomando dos puntos notando que a la hora de hacer la regresión lineal, la línea quedaba "atravesada" con el resto de los puntos. Luego de probar distintas variantes, se comprobó que lo que daba mejores resultados era tomar, para el caso de los máximos, los tres máximos locales más altos y el último máximo local (el más cercano al final de la gráfica). El razonamiento detrás de esto es que el propósito de la línea superior es conseguir la tendencia de los máximos, entonces se supuso que tomando los más representativos (los más grandes) se lograría mostrar correctamente esta tendencia en la gráfica.

El enfoque anterior mejoró mucho la regresión pero notamos que en algunos casos se tienen líneas en lugares que no tenían sentido. Por ejemplo, casos en que se

² <https://plot.ly/python/>

³ <https://plot.ly/python/candlestick-charts/>

tenían varias líneas de tendencia en un rango muy acotado. Para contrarrestar esta situación se introdujo el último máximo en el cálculo, esto eliminó el problema de varias líneas, logrando que se graficara solo una y que quedará correctamente posicionada en la gráfica.

Una vez decidido el cómo conseguir los puntos que formarán las líneas, se definió como tomar el comienzo y fin de las mismas. Se definió una "ventana" que define el largo del triángulo, siendo el largo la cantidad de puntos considerados para su cálculo. La ventana cubre desde el último punto dado por la fecha de fin de la gráfica hasta N puntos hacia atrás. Una vez que se tienen los puntos de comienzo y fin, se generan las líneas de tendencia. Es raro que estas líneas se intersecten dentro de la ventana, entonces por razones de apariencia y claridad para el usuario lo que se realizó fue extender las líneas hasta su intersección y de esta manera generar el triángulo.

Con el triángulo ya graficado se puede analizar que tipo de triángulo es, calculando el ángulo que forman las líneas entre sí. Para esto se aplicaron conceptos básicos de trigonometría, utilizando una línea paralela al eje x de la gráfica y generando una proyección de esta línea junto con cada una de las líneas del triángulo. Si el ángulo entre la línea de resistencia y la línea paralela está entre 0 y 10 grados es un triángulo ascendente, si el ángulo entre la línea de soporte y la línea paralela está entre 0 y 10 grados es un triángulo descendente, si no es un triángulo simétrico.

Es importante aclarar que no siempre se es posible graficar un triángulo, hay casos donde simplemente no se puede marcar ninguna tendencia con esta técnica.

4 - Renderizar vista HTML de las gráficas.

Se utilizó Flask ⁴ para renderizar la vista de las gráficas. Esta vista posee dos inputs de entradas: fecha inicial y fecha final, con los cuales se generará la gráfica correspondiente.

4.2. Herramientas

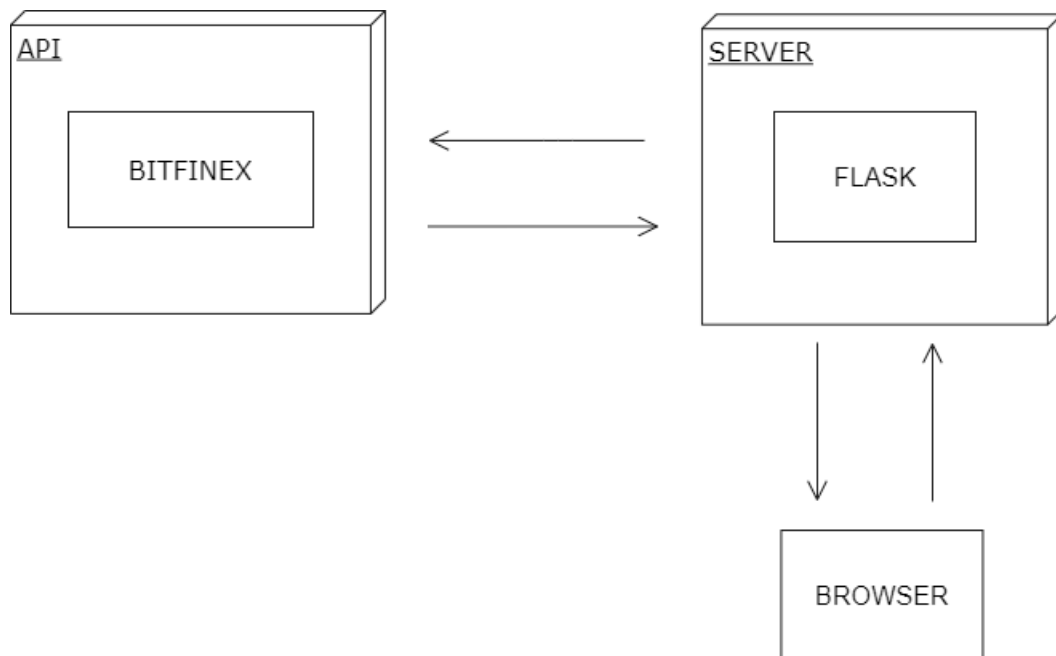
A continuación se listan las herramientas utilizadas y las versiones de las mismas:

⁴ <https://www.fullstackpython.com/flask.html>

- Python version 3.7
- requests version 2.19.1
- plotly version 3.0.0
- numpy version 1.14.5
- scipy version 1.1.0
- ipython version 6.4.0
- matplotlib version 2.2.2
- pandas version 0.23.3
- flask version 1.1.1

4.3. *Arquitectura*

A continuación se muestra el diagrama de la arquitectura del proyecto, el mismo contiene un servidor que realiza los request hacia la API de Bitfinex y genera los gráficos que serán consumidos por un cliente que utilice el browser.



5. Trabajo a futuro

La solución alcanzada en este proyecto abarca los datos solamente para un cambio: BitFinex. Por lo tanto como trabajo a futuro se podrían agregar nuevos cambios para generar gráficos con los datos de los mismo, dado que la lógica para generar estos gráficos es re-utilizable.

Se puede trabajar también en la obtención de una interfaz de front-end más compleja y que sea más configurable. Esto es, por ejemplo, tener la posibilidad de seleccionar el local de cambio que el usuario desea consultar o el tipo de moneda.

6. Conclusions

La manipulación de criptomonedas es cada vez más popular y cada vez hay más personas interesadas en las mismas, por lo tanto es necesario contar con herramientas que logren ayudar en el momento de la toma de decisiones para la compra o la venta de las mismas.

Durante este proyecto se logró llegar a una solución que representa de manera gráfica los datos de un cambio determinado y brinda al usuario a través de la técnica de patrones de triángulo una primera idea de cómo se comportará en el futuro la criptomoneda que está consultando. De esta forma un usuario que no es experto tendrá la posibilidad de acercarse un poco más a el análisis de estas criptomonedas.

A su vez, a lo largo de este proyecto se incorporó mucho conocimiento sobre las criptomonedas y su comportamiento general.

Referencias

- [1] Página web StockCharts.
URL: https://school.stockcharts.com/doku.php?id=chart_analysis:chart_patterns:ascending_triangle_continuation
- [2] Velas Japonesas definición.
URL: <https://economipedia.com/definiciones/velas-japonesas.html>
- [3] Velas Japonesas interpretación.
URL: <https://www.rankia.cl/blog/analisis-ipsa/3320503-velas-japonesas-tipos-interpretacion-patrones-cambio-tendencia>
- [4] Patrones de triángulo.
URL: <https://www.investorsunderground.com/stock-chart-patterns/triangle-patterns/>
- [5] Patrones de triángulo.
URL: <https://www.tradingview.com/ideas/triangle/>
- [6] Generación de triángulos
URL: https://school.stockcharts.com/doku.php?id=chart_analysis:chart_patterns:symmetrical_triangle_continuation