



Universidad del Salvador

Facultad de Ciencias Económicas y Empresariales

Licenciatura en Mercados de Capitales

Trabajo Final de Grado

TRADING ALGORÍTMICO

DESARROLLO DE UN SISTEMA AUTOMÁTICO BASADO EN UNA ESTRATEGIA DE CRUCE DE MEDIAS MÓVILES

Septiembre 2016
Buenos Aires, Argentina.

Autor: Carlos Galceran

Tutor de Tesina: Hector Cosentino

Email: cargalceran@gmail.com

Agradecimientos

A mi madre.

A mi familia por ayudarme a concretar mis sueños.

A mi mujer Loana Martinez por apoyarme incondicionalmente.

A Wenceslao Maislin por presentarme el mundo que cambiaría mi vida.

A Damián Sastre por haberme enseñado y sido mi mentor compartiendo sus conocimientos conmigo.

I.	Introducción	4
A.	Antecedentes	4
B.	Evolución del Modelo Electrónico de Mercado	6
C.	Trading Algorítmico	8
D.	Diferencias entre Sistema de Trading Algorítmico y HFT	8
E.	Ventajas y desventajas del trading automático	9
F.	Objetivo de la Tesina	12
II.	Estrategias	13
A.	Introducción	13
B.	Relative Strength Index (RSI)	13
C.	Moving Average Convergence-Divergence (MACD)	14
D.	Medias Móviles	15
E.	Selección de la estrategia a programar	17
III.	Arquitectura del Sistema	18
A.	Introducción	18
B.	Plataforma de Trading - Quantopian®	23
C.	Lenguaje Python	24
D.	Las principales librerías de Python para finanzas	26
IV.	Desarrollo del Sistema de Trading	28
A.	Introducción	28
B.	Desarrollo de nuestro sistema	28
V.	Backtesting	33
A.	Introducción	33
B.	Resultados	33
VI.	Conclusión	40
	Referencias	41

Tabla de Ilustraciones

Ilustración 1. Crecimiento del Trading Algorítmico durante los últimos años	4
Ilustración 2. Mercado Norteamericano: Porcentaje de órdenes generadas por algoritmos.	6
Ilustración 3. Posiciones de un trader de alta frecuencia estudiado por Jovanovic y Menkveld en el 2010. Este es el 30 de enero del 2008, un HFT operando en el EuroNext y ChiX.	8
Ilustración 4. Ejemplo de Estrategia de Cruce de Medias Simples	16
Ilustración 5. Ejemplo de árbol de decisión de una estrategia	19
Ilustración 6 Ejemplo de requerimiento de Confiabilidad	21
Ilustración 7. Rentabilidad del Backtesting	32
Ilustración 8. Gráfico del Precio y las Medias Móviles	33
Ilustración 9. Transacciones operadas por el Algoritmo	33
Ilustración 10. Detalle de la Rentabilidad alcanzada por periodos	34
Ilustración 11. Pérdida máxima desde el máximo alcanzado	35

I. Introducción

A. Antecedentes

Siempre existió una tendencia de largo plazo hacia los mercados financieros cada vez más electrónicos. Durante la década de 1970, el NYSE¹ introdujo el sistema de manejo de órdenes designadas (“DOT” designated order turnaround, por sus siglas en inglés) que lo que hacía era dirigir las órdenes de compra y venta al sector físico en el mercado donde se compraban y vendían esas acciones para que esas órdenes pudieran ser ejecutadas manualmente. Luego durante la década de los 80s, los mercados avanzaron progresivamente a un modelo donde las órdenes eran enviadas hacia un libro electrónico el cual las publicaba y automáticamente eran ejecutadas cuando encontraban contrapartes que satisfacían el criterio. En la década de 1990, surgieron las redes de comunicación electrónica (“ECN”, por sus siglas en inglés) ofrecían un nuevo lugar donde operar que irrumpió con el estatus monopólico establecido por los mercados financieros centrales. Es decir, ya se podía elegir otros “pseudo mercados” donde ejecutar la misma operación, no solo tenía que ser en el NYSE.

A medida que los mercados se movieron hacia un modelo más y más electrónico, también lo hicieron los inversores.²

El uso de *algoritmos* que monitorean el mercado y manejan el proceso de compra y venta de activos en los mercados financieros comenzó a finales de los años 90s y se ha incrementado en los últimos años.

El rápido desarrollo tecnológico de los mercados financieros mundiales trajo aparejado la creación de nuevos instrumentos financieros y estrategias de compraventa más complejas, que se usan para lidiar con las condiciones del mercado que cambian constantemente.

En este nuevo mercado, muchas de las oportunidades de generar una ganancia se presentan en menos de un segundo, haciendo imposible que un ser humano pueda reaccionar para tomar ventaja. Es por eso que desarrolladores y expertos en finanzas cuantitativas se han juntado para desarrollar sistemas automáticos de trading que capturan los rápidos movimientos de precios en el mercado y a través de una estrategia definida pueden explotar esas oportunidades.

Los tipos más comunes de participantes en el mercado financiero que se pueden encontrar hoy en día son: los inversores minoristas, traders propietarios, hacedores de mercados, instituciones que se dedican solamente al lado de la compra (buy-side) e instituciones que solo operan el lado de la venta (sell-side).

¹ La Bolsa de Nueva York (New York Stock Exchange, NYSE, en inglés) es el mayor mercado de valores del mundo en volumen monetario y el primero en número de empresas adscritas.

² ver Biais and Woolley, “High Frequency Trading,” 03.

Según el reporte de la consultora The Tabb Group publicado en el año 2005, y basado en una encuesta del sector, indica que 61% de las firmas norteamericanas del buy-side emplean mayormente modelos de ejecución automáticos y pronosticaban un crecimiento del 144% en el uso de algoritmos de trading a finales del 2006³. Esta dinámica está cambiando con la implementación del concepto del trading algorítmico como servicio o *ATaaS* (*Algorithmic Trading as a Service* por sus siglas en inglés). Ya que los sistemas se están haciendo cada vez más accesibles para los inversores minoristas. Las ventajas del trading algorítmico se han popularizado tanto que la mayoría de los brokers y proveedores de tecnología mundiales ofrecen el servicio de alguna manera.

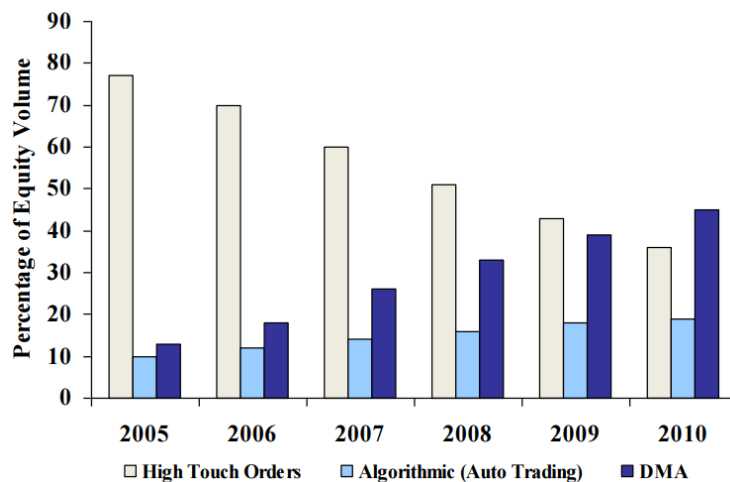
Ilustración 1. Crecimiento del Trading Algorítmico durante los últimos años



Fuente: Aite Group

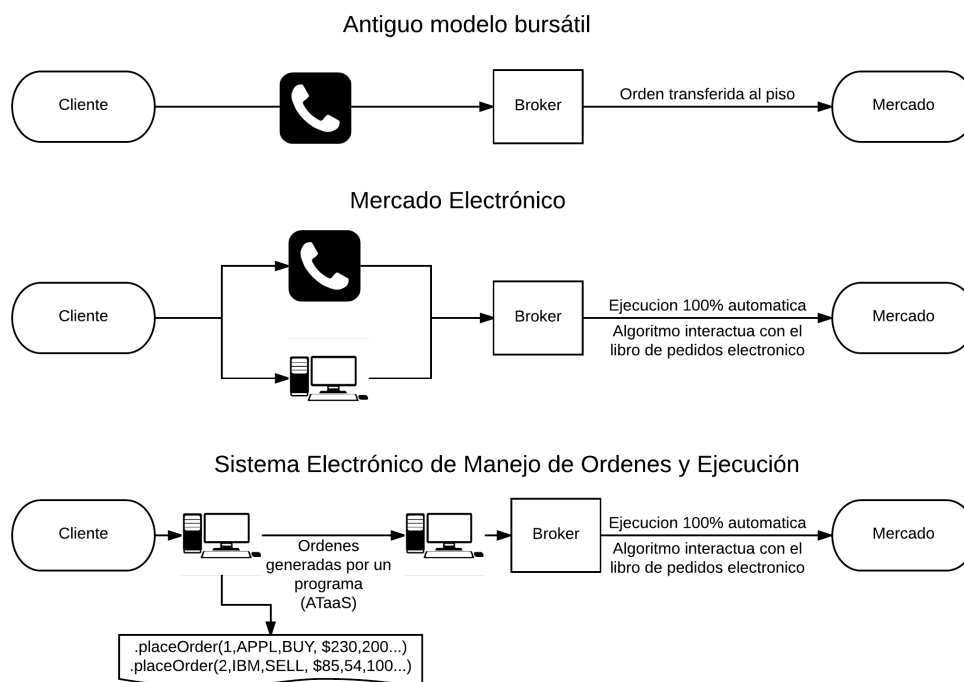
³ Tabb, Larry, "Institutional Equity Trading in America: A Buy-Side Perspective," consulting report, The Tabb Group, April 2004.

Ilustración 2. Mercado Norteamericano: Porcentaje de órdenes generadas por algoritmos.⁴

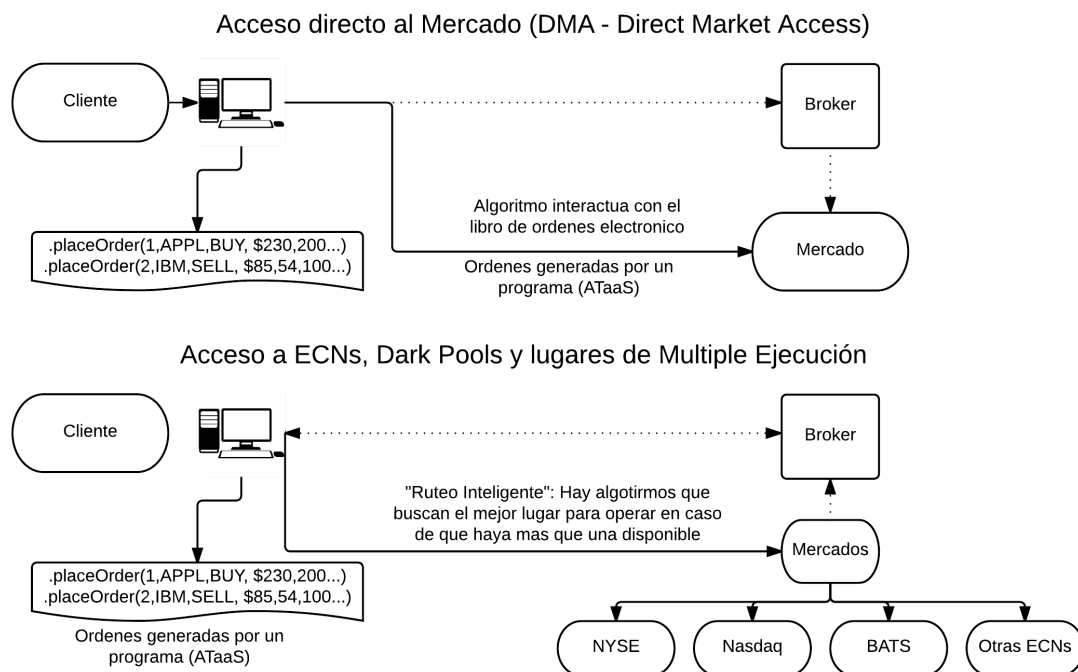


Fuente: Algorithmic and High-frequency trading: an overview por Marco Avellaneda (2011)

B. Evolución del Modelo Electrónico de Mercado



⁴ ver Avellaneda, "Algorithmic and High-Frequency Trading: An Overview," 2.



C. Trading Algorítmico

Básicamente, un sistema de trading algorítmico es un sistema conectado a un mercado financiero que puede enviar electrónicamente operaciones al mercado. Los tipos de operaciones que estos sistemas producen incluyen, la oferta (el precio al cual se desea comprar), la venta (el precio al cual se desea vender) y el volumen (cantidad de activos que se pretenden operar). La decisión de cuándo enviar al mercado esas operaciones está dictada por una estrategia de trading predefinida.

Los sistemas algorítmicos de trading son creados a través de la conversión de una estrategia de compra y venta de activos a un conjunto de líneas de código o programa para que las computadoras puedan leer y ejecutar. Lo que cada sistema hace es correr esas líneas de código una y otra vez en búsqueda de operaciones que adhieren a la estrategia predefinida.

Por lo tanto, un sistema de trading robusto y una estrategia completamente probada juegan un rol de vital importancia para minimizar los riesgos y maximizar el beneficio. Una de las *ventajas* de estos sistemas es que producen operaciones automáticamente y no conllevan emoción, lo que hace que los resultados sean consistentes para los traders⁵. ⁶ En este sentido, habiendo expuesto uno de los beneficios de esta modalidad, en el trading automático también existen riesgos que voy a detallar más en profundidad en un apartado exclusivo.

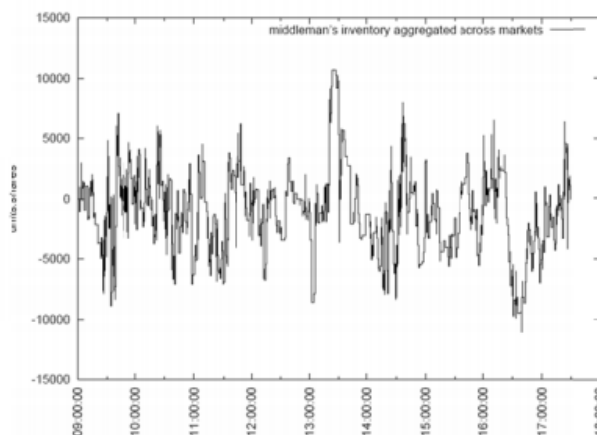
⁵ *trader* es un agente que puede funcionar como intermediario, operador de cobertura y especulador.

⁶ Le, "Automated Trading System," 9.

D. Diferencias entre Sistema de Trading Algorítmico y HFT

La modalidad del High Frequency Trading⁷, HFT por sus siglas en inglés y el más tradicional “Sistemas de Trading Algorítmico o Inversiones Cuantitativas”, tienen un número de características comunes, ambas en el uso que se le da a los algoritmos y en el impacto que tiene en el mercado. Ambos buscan de identificar y explotar patrones sistemáticos de precios, como por ejemplo, el momento o la reversión de mercado. Sin embargo, los HFTs se concentran más en la información que los movimientos de precio transmiten sobre las estrategias de trading o señales de otros participantes del mercado. La diferencia clave está en el periodo de tenencia de los activos o en el horizonte de inversión. Los HFTs tienen rangos de milisegundos hasta horas. Las posiciones enteras son cerradas al final de cada día.⁸ Para ilustrar el corto horizonte de las tenencias de los traders que negocian con alta frecuencia, el siguiente estudio realizado por Jovanovic y Menkveld en el año 2010, lo puede hacer.⁹

Ilustración 3. Posiciones de un trader de alta frecuencia estudiado por Jovanovic y Menkveld en el 2010. Este es el 30 de enero del 2008, un HFT operando en el EuroNext y ChiX.



E. Ventajas y desventajas del trading automático

⁷ La negociación de alta frecuencia, también conocida en el ámbito financiero por su nombre en inglés high-frequency trading o por sus siglas HFT, es un tipo de negociación que se lleva a cabo en los mercados financieros utilizando intensamente herramientas tecnológicas sofisticadas para obtener información del mercado y en función de la misma intercambiar valores financieros tales como activos u opciones.

⁸ ver Biais and Woolley, “High Frequency Trading,” 3.

⁹ Jovanovic and Menkveld, “Middlemen in Limit Order Markets 1.”

Una de las mayores atracciones del trading automático es la capacidad de remover las emociones a través de la ejecución de operaciones automáticamente cuando ciertos criterios son alcanzados. Pero existen otras y en este apartado pretendo explicar cuáles son las ventajas y desventajas, como tan bien las realidades, de esta modalidad.¹⁰

Existen una gran cantidad de ventajas de tener a una computadora que monitoree el mercado en busca de oportunidades y ejecute las operaciones, entre ellas se incluye:

1. Minimizar Emociones:

Los sistemas algorítmicos de trading minimizan emociones a través del proceso automático de trading. Al reducir significativamente la emoción, los traders pueden llevar a cabo su estrategia sin ningún problema.

2. Posibilidad de hacer *BackTesting*¹¹:

Se puede aplicar la estrategia a datos históricos de los activos seleccionados y determinar la viabilidad de la idea. Cuando se diseña un sistema de trading, todas las reglas tienen que ser absolutas, no puede haber interpretación (la computadora no puede adivinar, se le tiene que decir exactamente lo que tiene que hacer). Lo bueno de este tipo de pruebas es que no hay que arriesgar capital para determinar si la estrategia sirve.

3. Preservar la disciplina:

Debido a que las reglas son establecidas de antemano dentro de la estrategia y programada dentro del sistema, los individuos que ejecuten esta estrategia podrán mantener la disciplina incluso en momentos donde el mercado se mantenga volátil.

4. Conseguir lograr consistencia:

Uno de los mayores desafíos en trading es planear la operaciones y luego operar de acuerdo al plan, aunque el plan tenga el potencial de ser beneficioso, los operadores que ignoran las reglas alteran cualquier expectativa de beneficio que el sistema les podría dar. No hay ningún plan que garantice ganar el 100% de las veces, las pérdidas son parte del juego. Pero las pérdidas, pueden ser psicológicamente traumatizantes y un trader que ha experimentado dos o tres pérdidas consecutivas, puede llegar a dejar pasar la próxima operación. Si la próxima operación hubiera sido

¹⁰ investopedia - The Pros And Cons Of Automated Trading Systems

¹¹ Backtesting: es el proceso de probar una estrategia de operación en períodos anteriores. En lugar de aplicar una estrategia para el período de tiempo hacia delante, lo que podría llevar años, un operador puede hacer una simulación de la estrategia de operación con los datos del pasado con el fin de medir la eficacia.

ganadora, el trader destruiría la expectativa del sistema. El trading automático permite a los traders conseguir lograr consistencia a través de operar el plan determinado.

5. Mejorar la velocidad de ingresar en el mercado:

Las computadoras responden inmediatamente a los cambios en las condiciones del mercado, tan pronto como los criterios son alcanzados los sistemas automáticos son capaces de generar órdenes, incluyendo Stop Losses¹² y Profit Target¹³.

6. Diversificación

Los sistemas automáticos de trading permiten al usuario operar en múltiples cuentas o varias estrategias a la misma vez. Esta modalidad permite crear posiciones de cobertura para mitigar algunos riesgos.

Entre las *desventajas* que encontramos en el trading automático podemos incluir:

1. Fallas Mecánicas:

La teoría detrás del trading automático es simple: uno configura el software, programa las reglas y se sienta a mirar cómo opera solo. Sin embargo, en realidad, es un método sofisticado de trading pero no es infalible. Dependiendo de la plataforma en que se opere, la operación puede residir dentro de una computadora y no de un servidor, por lo tanto la conexión a internet se puede perder y la orden no ser enviada al mercado.

2. Monitoreo:

Aunque estaría bueno prender la computadora y luego tomarse el día, los sistemas automáticos de trading requieren monitoreo. Esto es debido a que pueden haber potenciales fallas mecánicas, desde problemas de conectividad, como nombramos anteriormente hasta cortes de luz, y fallas en la computadora o en el sistema. Es posible que un sistema sufra de anomalías que puedan resultar en órdenes de compra errantes, perdidas o tal vez duplicadas. Si el sistema es monitoreado estos problemas se pueden corregir y solucionados rápidamente.

3. Riesgo Sistémico

¹² *stop loss* es una orden que das a tu broker para “detener pérdidas” (“stop loss”). La regla de oro de la especulación en Bolsa es mantener el riesgo siempre bajo control.

¹³ Similar al Stop Loss pero este caso cierra la operación si se llega al precio objetivo.

Unos de los riesgos más peligrosos dentro de estos sistemas automáticos es el riesgo sistémico¹⁴, si el mercado en cierto momento comienza a comportarse de una manera errática y el sistema no cuenta con un mecanismo de Stop Loss, o no cuenta con una serie de mecanismos que se disparen para protegerse ante algún tipo de anomalía, probablemente afecte a los resultado de la estrategia.

4. Discrepancia entre operaciones teóricas:

Puede existir una discrepancia entre la las “operaciones teóricas” generadas por la estrategia y el componente de entrada en la plataforma que hace las operaciones reales. Muchos de los traders tienen que esperar que la curva de aprendizaje sea larga cuando se trata de utilizar estos sistemas automáticos de trading. Generalmente es una buena idea comenzar con montos pequeños a medida que el proceso se va refinando.

F. Objetivo de la Tesina

El objetivo de la tesina es el desarrollo de un sistema de trading algorítmico que se base en una estrategia específica de compraventa de acciones. Se hará mención de las estrategias de trading más comunes en el área bursátil y describiremos sus características, aunque solo elegiremos una y de tal forma que nos apoyaremos en sus reglas para la elaboración del programa. Por otro lado, evaluaré el por qué se ha optado por construir el sistema dentro de la plataforma de Quantopian® y por lo tanto, utilizado el lenguaje de programación Python.

Abordaré el marco teórico en el cual nos moveremos definiendo algunos conceptos básicos, y también mostrare las pruebas de fiabilidad a las que someteré a la estrategia antes de colocarla en tiempo real y que operen en el mercado en base a los resultados obtenidos por el BackTesting de nuestro sistema.

Por lo tanto, el objetivo radica en la elaboración de un algoritmo que sea capaz de operar de forma automática en los mercados financieros. Limitaremos el experimento dentro de un escenario *simple* y se irá dando las explicaciones pertinentes a la complejidad del mercado a medida que avancemos en las especificaciones técnicas del proyecto.

¹⁴ En finanzas, riesgo sistémico es el riesgo común para todo el mercado entero. Puede ser interpretado como "inestabilidad del sistema financiero, potencialmente catastrófico, causado por eventos idiosincráticos o condiciones en los intermediarios financieros".

II. Estrategias

A. Introducción

Se puede definir una estrategia de compra y ventas de activos como el conjunto de reglas, comportamientos y procedimientos diseñados para guiar a un inversor en la selección de una cartera de valores.

Las bases de todos los sistemas de trading es el análisis técnico. El principio básico del análisis técnico es que la historia de alguna manera se repite.¹⁵ Es por ello que según J.J. Murphy, *las estrategias están basadas en indicadores que se refieren al análisis técnico de estos mercados.*¹⁶

Existen multitud de indicadores que dan lugar a diversas estrategias que pueden complementarse entre sí hasta alcanzar aquella que mejor se adapte a las expectativas del inversor. Es por ello que en esta tesina se hace referencia a tres de los indicadores más utilizados para el desarrollo de estrategias: RSI, MACD y las medias móviles.

B. Relative Strength Index (RSI)

El RSI es un indicador muy utilizado en el análisis técnico debido a la facilidad de su interpretación porque muestra la fuerza del precio mediante la comparación de los movimientos individuales al alza o a la baja.¹⁷ Este indicador nació de un estudio en el año 1978 realizado por Welles Wilder, tiene en cuenta los precios de cierre del activo de los últimos 14 días e indicaba techos o pisos en los precios de los activos si es que el oscilador RSI supera los niveles establecidos de 70% o 30%, respectivamente. Hoy en día todas estas variables son manipulables y se pueden configurar de la mejor manera que le parezca al operador. En el siguiente ejemplo solo se tomaron los últimos 10 días de los cierre de precio de un activo.

¹⁵ ver John R. Hill, *The Ultimate Trading Guide*, 153.

¹⁶ ver Murphy, "Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications."

¹⁷ Welles Wilder, *New Concepts in Technical Trading Systems*, 63.



C. Moving Average Convergence-Divergence (MACD)

El MACD entendido como el indicador de análisis técnico significa literalmente

Convergencia/Divergencia de la media móvil y está compuesto por tres componentes: el MACD en sí, la señal y el histograma. El componente MACD es la diferencia entre dos medias móviles exponenciales de diferente longitud: un promedio rápido establecido por convenio en 12 periodos de tal forma que sea sensible a los movimientos del precio a corto plazo y un promedio más lento, establecido por un promedio de 26 periodos.

El segundo componente se trata de la Señal, que será el encargado de realizar un promedio móvil exponencial del MACD explicado anteriormente, siendo el parámetro más común para establecer esta nueva variable dentro del conjunto es de 9 periodos. El histograma es el último de los componente que interactúan dentro del indicador MACD, será el encargado de indicar si finalmente se inicia o se cierra una posición basándose en la diferencia entre el MACD y la Señal.

La interpretación en el mercado radica en la posición del MACD con respecto a la Señal, de tal forma que si el primero está por encima del segundo se entiende que la decisión deberá ser de compra y si se produce el caso contrario en el que el MACD esté por debajo de la señal, se vende. No es recomendable esperar a confirmar la señal cuando el indicador cruce la línea cero ya que esto puede derivar en una operación tardía y sin criterio dentro del mercado.

Es posible que el precio alcance máximos más altos mientras el indicador disminuye sus máximos, esto se conoce como una divergencia negativa y se relativiza dentro del contexto como que el movimiento del precio está acabando y por tanto se genera una señal de venta. Análogamente en sentido contrario, la divergencia positiva se produce cuando el precio alcanza mínimos más bajos y el indicador aumenta sus mínimos, lo que produciría una señal de compra.

En el ejemplo a continuación se puede observar como el indicador prevé la caída cuando se produce una divergencia en el Histograma y en las medias.



D. Medias Móviles

Podemos definir a la media móvil como un indicador de seguimiento de tendencia cuyo objetivo es identificar o señalar el principio o final de una tendencia o su próximo cambio. Las medias móviles ayudan a aprovechar la estacionalidad y eliminar distorsiones coyunturales, ya que atenúan las fluctuaciones y ayudan a identificar las tendencias y posibles cambios de dirección¹⁸.

La media móvil es la más sencilla de calcular, pero no por ello la menos efectiva, de ahí que se trate de la media móvil más usada.

Su expresión matemática es la siguiente:

$$MMS_t = n^{-1} \sum_{i=0}^{n-1} P_{t-i}$$

Consiste en realizar un promedio proporcional de n observaciones móvil en el tiempo, es decir, tomar promedios sucesivos de un conjunto de valores cuyo número se mantiene constante durante el proceso de cálculo. La media móvil simple considera todas las sesiones de la misma manera, cosa que hace que se retrasen los datos del periodo que cubre, no entrando en su cómputo ningún dato o coeficiente ponderado que recoja datos pasados.

¹⁸ ver Marcelo and Quirós, "Estabilidad de Las Estrategias de Inversión Con Medias Móviles," 853.

La media móvil exponencial pretende un allanamiento de la media móvil simple, para ello emplea una ponderación de tipo exponencial que da mayor importancia a las cotizaciones más recientes, mientras que las más antiguas pierden progresivamente importancia. Su expresión matemática es la que voy a describir a continuación.

$$MME_t(n) = \alpha \cdot p_t + (1 - \alpha) \cdot MME_{t-1}(n)$$

donde α es el factor de ponderación acotado entre 0 y 1 y $MME_{t-1}(n)$ es el valor de la media móvil exponencial del periodo anterior.

Las medias móviles ponderadas por lo contrario, tienen la ventaja sobre las medias móviles simples que, al ponderar algunas sesiones sobre otras, nos permiten dar más importancia a los días más recientes sobre los más alejados en el tiempo, consiguiendo que las señales de cambio de tendencia no se retrasen. Su expresión matemática es la siguiente:

$$MMP_t(n) = \frac{\sum_{i=0}^{n-1} P_{t-1}(n-i)}{\sum_{i=0}^{n-1} (n-i)}$$

donde el factor de ponderación permite otorgar más importancia a la información contenida en las observaciones más cercanas a t y menos importancia conforme nos alejamos del momento t .

Una vez obtenida la media móvil, la aplicación de la misma como indicador puede realizarse por simple comparación entre la media móvil y la cotización del índice, o bien mediante la comparación de medias móviles de distinto número de periodos. *La generación de señales con la utilización de una sola media móvil se produce cuando los precios de cierre se mueven por encima de ella, generando una señal de compra, en cambio, si los precios se mueven por debajo de la propia media móvil, genera una señal de venta.*

E. Selección de la estrategia a programar

Tal se describió en la estrategia de medias móviles, en esta Tesina y por motivos de mantener los conceptos lo más simple posible, nos centraremos en la utilización de la estrategia de cruce de medias de corto y largo plazo simples. La estrategia del sistema de trading algorítmico en este caso consistirá en invertir con medias móviles, es decir en entrar en el mercado cuando la media a corto plazo supera la media de largo y salir cuando está por

debajo. Gráficamente la orden de entrada se produce cuando la media a corto *corta* la media a largo en sentido ascendente y la orden de salida cuando lo hace en sentido descendente. Siendo claramente $n_1 < n_2$, entonces si $MM_t(n_1) > MM_t(n_2)$ esto es visto como una señal de compra. De la misma forma si, $MM_t(n_1) < MM_t(n_2)$ se interpreta como una señal de venta.

La estrategia de inversión a seguir por el algoritmo en este caso será la siguiente, la cartera permanecerá invertida en el activo elegido si la media de corto supera a la media de largo plazo. Si la media a corto plazo está por debajo de la media a largo entonces desinvertimos, colocando la cartera en posición de liquidez.

Ilustración 4. Ejemplo de Estrategia de Cruce de Medias Simples



III. Arquitectura del Sistema

A. Introducción

Si uno analiza el documento publicado por Stuart Gordon Reid en el 2013, sobre los Sistemas de Trading Algorítmicos¹⁹, llega a la conclusión de que la arquitectura de los sistemas es un tema muy amplio, porque se requiere pensar fuertemente en qué tecnología se va a desarrollar, ya que la complejidad varía por la diversidad de requerimientos funcionales y regulatorios que estos deben cumplir. En otras palabras, la arquitectura provee una infraestructura para que el sistema satisfaga nuestros requerimientos funcionales, y no funcionales.

Según el autor, estos requerimientos funcionales que el sistema deberá satisfacer se dividen en tres categorías, que incluyen otros requerimientos dentro de ellas:

1. Requerimientos funcionales:

a. Tomar decisiones de Trading:

- i. *Obtener datos del mercado:* bajar, filtrar y guardar datos estructurados y no estructurados. Los datos estructurados incluyen datos del mercado en tiempo real de Reuters o Bloomberg, por ejemplo. Los Datos no estructurados son las noticias y los datos de las redes sociales.
- ii. *Definir una estrategia:* especificar reglas y una estrategia. Las reglas de trading están estructuradas en un árbol de decisión que define la estrategia. (ver gráfico más abajo).
- iii. *Analizar títulos contra la estrategia definida:* por cada título valor, el sistema debe obtener los datos y filtrar a través de la estrategia para determinar si comprarlo o no. De la misma manera que para las posiciones abiertas debe determinar qué título valor vender.

¹⁹ ver Reid, "Algorithmic Trading System," 9.

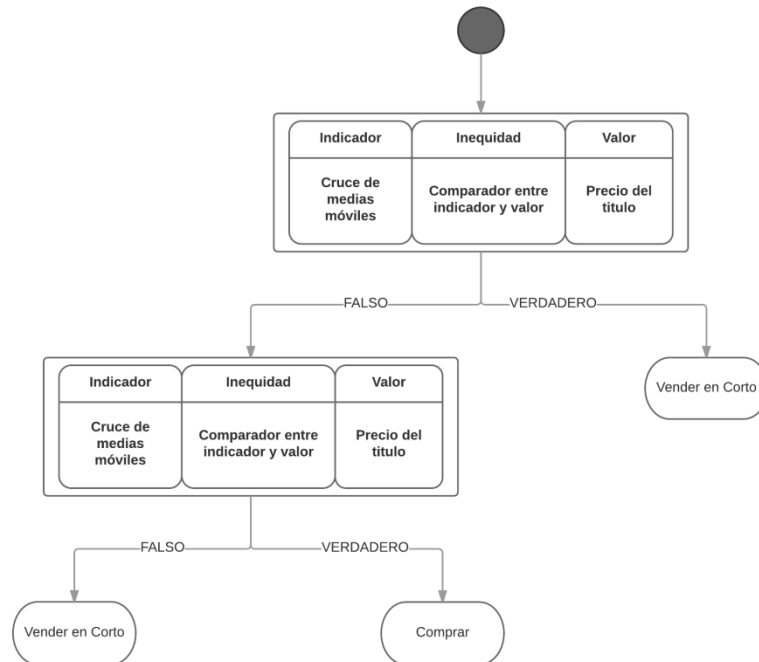
b. Crear una orden de trading

- i. *Obtener información sobre la operación:* para cada decisión, obtener el símbolo del título (MSFT, por ejemplo), precio y cantidad.
- ii. *Crear la orden de trading:* por cada decisión de trading, especificar el tipo de orden e información de la operación. Hay seis tipos de órdenes: Compra, Venta en Corto, Compra a precio de mercado, Compra o venta a precio límite, Stop Loss, y condicionales.

c. Manejar las órdenes:

- i. *Manejar órdenes pendientes:* por cada orden hay que validar y confirmar esa orden.
- ii. *Enviar órdenes:* enviar cada operación al mercado, ECN o agencia bursátil.
- iii. *Manejar órdenes enviadas al mercado:* rastrear el estado de cada orden enviada al mercado, si la orden fue ejecutada entonces el sistema deberá crear una posición abierta.

Ilustración 5. Ejemplo de árbol de decisión de una estrategia



2. Requerimientos no funcionales:


Hay muchos requerimientos funcionales los cuales se compensan entre unos y otros, por ejemplo, el incremento en la performance del sistema usualmente es directamente proporcional con el incremento en el costo de propiedad. Es decir, para que el sistema funcione mejor, hay que alojar el sistema en servidores de terceros que funcionan mejor.

1. Requerimientos no funcionales:

- a. Escalabilidad: Esta es la habilidad del sistema en adaptarse y funcionar ante un incremento en el trabajo.
- b. Performance: Es la cantidad de trabajo que produce un sistema comparado con el tiempo y los recursos que tomaron para funcionar.

- c. **Modificabilidad:** Es la tranquilidad con la que el programa puede ser cambiado. El sistema debe tener estrategias y procesos de datos modificables.
- d. **Confiabilidad:** Es la eficacia y la dependabilidad de un sistema con respecto a los resultados producidos. (ver artículo abajo sobre un ejemplo)
- e. **Auditable:** Es la tranquilidad con la que el sistema se puede auditar, tienen que ser auditables a ver si cumplen con la normativa financiera y regulatoria.
- f. **Seguridad:** Es la seguridad de una organización frente a actividades criminales o de terrorismo. Las estrategias de trading deben ser aseguradas porque representan una propiedad con valor intelectual.
- g. **Tolerante a las fallas:** Es la habilidad del sistema en operar ante una dificultad o una falla. Esto es similar a la Confianza, salvo que el sistema debe continuar siendo confiable hasta luego de una falla para evitar pérdidas financieras.
- h. **Interoperabilidad:** esta es la habilidad del sistema para operar y comunicarse con otros sistemas. Esto es importante porque un sistema puede requerir interactuar con una interfaz de otro sistema de gestión, de portfolio o contable.


Ilustración 6 Ejemplo de requerimiento de Confiabilidad²⁰



HOME U.S. ▾ NEWS MARKETS INVESTING TECH MAKE IT VID

Knight Capital Debacle a Serious Warning Signal: Pros

Lee Brodie | @LeeBrod
Wednesday, 1 Aug 2012 | 1:25 PM ET



Again, unusual and volatile trading has triggered a ripple that's cut a swatch across a range of stocks and it calls the reliability of technology into question.

In a statement, market maker Knight Capital said a "technology issue" affected the routing of about 150 stocks listed on the New York Stock Exchange.

- [Market Snafus: From Facebook to Flash Crash](#)

Although the events are complex, the pros say what you need to know is that there was a problem with Knight Capital *software* and the way it interacted with the NYSE.

"Hundreds of these systems interact all around the country – that's the modern market making system," explains trader Joe Terranova, , chief market strategist for Virtus. "And we're learning it doesn't always work."

That's scary.

Henry Blodgett, of Business Insider, says Wall Street needs to wake up. "Trading relies on technology. When you have a big technical error – it's a warning."

A serious warning.

²⁰ ver pagina de CNBC - <http://www.cnbc.com/id/48444555>

B. Plataforma de Trading - Quantopian®

El propósito del apartado anterior fue para servir de marco para introducir el uso de la plataforma de Quantopian® que vamos a utilizar para poder escribir el sistema de trading algorítmico. Esta plataforma cumple con todos los requerimientos descritos en el apartado anterior, funcionales y no funcionales haciendo el trabajo mucho más simple.



Este sitio provee todo lo que un trader necesita para poder escribir estrategias de sistemas de trading algorítmicos de alta calidad. En donde uno puede hacer investigaciones usando una variedad de fuentes de datos, probar la factibilidad de la estrategia sobre datos históricos, y después probar su uso hacia adelante con datos en tiempo real.

La plataforma de Quantopian® está basada en algunos componentes interrelacionados. Esta contiene un ambiente de desarrollo que es usado para la investigación y análisis de datos durante la creación de los algoritmos. Este es un concepto bastante abstracto pero en realidad lo que quiere decir es que cuando uno abre una cuenta en esta plataforma de Quantopian®, se le asigna un espacio de trabajo, ese espacio se maneja a través de un lenguaje de programación llamado ipython (que proviene de python y que detallaré más adelante), y está cargado con todos los datos que uno puede llegar a necesitar para llevar a cabo una investigación, de esa manera simplificando el proceso al máximo.²¹

Por el momento y dada la reciente creación de esta nueva tecnología la plataforma ofrece datos históricos minuto-a-minuto desde el 2002 de todas las acciones que cotizan en los Estados Unidos. Como también acceso libre de cargos a más de 600 métricas provenientes de la empresa Morningstar, donde se puede medir y analizar la performance financiera de esas empresas, basados en sus datos publicados ante la SEC.

En resumen, la plataforma contiene los siguientes componentes:

²¹ <https://www.quantopian.com/help>

1. Servidor donde se aloja el ambiente de desarrollo.
2. Fuentes de Datos.
3. Ambiente de desarrollo interactivo para escribir algoritmos y probarlos.
4. Capacidad para “Operar en Papel”²².
5. Capacidad para Operar realmente a través de un Broker, llamado Interactive Brokers.

C. Lenguaje Python

Este lenguaje está dando que hablar en el mundo de las finanzas y programadores. Es un tipo de lenguaje de programación (conjunto de símbolos y reglas sintácticas y semánticas que son necesarias para crear programas que controlen el comportamiento físico y lógico de una máquina) que apuesta por la simplicidad, versatilidad y rapidez de desarrollo.

Su expansión y popularidad es relativamente reciente ya que fue creado a finales de la década de los ochenta por Guido Van Rossum en el Centro para las Matemáticas y la Informática en los Países Bajos, como un sucesor del otro lenguaje de programación, el ABC. Aunque no se publicó hasta 1991. En la actualidad es administrado por la Python Software Foundation.²³

1. ¿Por qué Python?

En los últimos años Python ha alcanzado gran popularidad debido a varias razones:

- a. Contiene una gran cantidad de librerías (tipos de datos y funciones incorporadas en el propio lenguaje) que te ayudan a realizar tareas sin tener que programar desde cero.
- b. Permite crear programas de forma veloz y sencilla y con menos líneas de código que con otros lenguajes.
- c. Se puede desarrollar Python en diversas plataformas como Unix, GNU/Linux, macOS, Window.
- d. Es gratuito, incluso para propósitos empresariales.

2. ¿Cuáles son sus principales características?

²² Operar en papel significa que uno puede operar simuladamente sin riesgo alguno ya que no hay dinero envuelto en el proceso.

²³ resumen de una comunicacion del Banco BBVA:

<https://www.bbva.com/es/noticias/ciencia/tecnologia/las-claves-entender-lenguaje-moda-python/>

a. Multiparadigma

Python es un lenguaje de programación multiparadigma, es decir, que no fuerza a los programadores a adoptar un enfoque particular de programación, sino que permite utilizar varios estilos.

b. Multiplataforma

Hay versiones disponibles de Python en muchos sistemas operativos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.

c. Interpretado

Un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete. Es decir, el código fuente se traduce a un pseudo código máquina intermedio llamado bytecode la primera vez que se ejecuta. Los lenguajes interpretados son más flexibles y portable frente a los compilados que tienen una ejecución más rápida.

d. Interactivo

Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.

e. Funciones y librerías

Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de cadena de textos, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos.

f. Sintaxis clara y legible

Python tiene una sintaxis muy visual, gracias a una notación indentada (con sangrado de texto mediante tabulación o espacios) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave `begin` y `end`. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tenga un aspecto muy similar.

D. Las principales librerías de Python para finanzas

Las principales librerías que se utilizan para el análisis financiero, todas estas librerías están contenidas dentro del ambiente de desarrollo que obtenemos al abrir una cuenta en el sitio de Quantopian®.

1. Pandas: La cual fue diseñada desde un comienzo para facilitar el análisis de datos financieros, principalmente las series de tiempo propias del mercado cambiario de acciones. Con las estructuras de datos que nos brinda esta librería se vuelve sumamente fácil modelar y resolver problemas financieros.
2. Numpy: El principal módulo matemático que nos ofrece Python, en él no solo vamos a encontrar las siempre prácticas matrices que facilitan en sobremanera el manejo de información numérica; sino que también vamos a poder encontrar un gran número de funciones matemáticas.
3. Matplotlib: librería para realizar gráficos en Python.

4. statsmodels: librería para realizar cualquier tipo de análisis estadístico.
5. PuLP: La cual nos permite crear modelos de programación lineal en forma muy sencilla.
6. Quandl: Este módulo nos permite interactuar fácilmente con la API de quandl.com para obtener en forma muy sencilla todo tipo de información financiera.
7. Zipline: es una librería para el comercio algorítmico y fue desarrollada íntegramente por Quantopian®, está basada en eventos y trata de aproximarse lo más cerca posible a como operan los verdaderos sistemas de comercio electrónico de las principales bolsas del mundo.

IV. Desarrollo del Sistema de Trading

A. Introducción

En Quantopian®, un *sistema de trading algorítmico* es un programa hecho en python que está definido por dos funciones especiales: `initialize()` y `handle_data()`. La función `initialize()` es solamente llamada cuando el programa es iniciado, y `handle_data()` es llamada una vez por minuto durante una simulación o trading real. El trabajo de la función `initialize()` es aplicar la lógica de una vez cuando el programa es iniciado, y el trabajo de la función `handle_data()` es decidir qué órdenes, si hay alguna, deben ser ejecutadas cada minuto.

El siguiente ejemplo es un algoritmo que coloca 100% de su portfolio en YPF:

```
def initialize(context):
    # Referencia a YPF
    context.ypf = symbol('YPF')

def handle_data(context, data):
    # Posicionar 100% de nuestro portafolio en acciones de YPF
    order_target_percent(context.ypf, 1.00)
```

B. Desarrollo de nuestro sistema

1. Planeamiento de la estrategia

Para comenzar hay que elegir una estrategia, como ya lo habíamos descrito en el último apartado de la sección de estrategias, utilizaremos la del cruce de medias móviles simples.

- Hipótesis utilizada: Si la media móvil simple de 50 ruedas *corta* la media móvil simple de 200 ruedas en sentido ascendente los precios del activo deberían subir, y si la media móvil simple de 50 ruedas *corta* la media móvil simple de 200 ruedas en sentido descendente, los precios del activo deberían bajar.

2. Elección de activo a operar

Este tipo de estrategia es usualmente implementada con una selección dinámica de títulos, pero para mantener el programa simple vamos a elegir a mano a una acción que cotiza en el NYSE. En este caso elegimos las acciones de la empresa de comida rápida, Mcdonald's.

```
# Seleccionar el título que queremos operar
context.titulo = symbol('MCD')
```

3. Configuración de las Comisiones a pagar

Este es un supuesto para cuando probemos el sistema con datos históricos, estas son las comisiones que cobra “Interactive Brokers” uno de los brokers electrónicos más conocidos de Estados Unidos. Cabe aclarar que el programa solo va a ser probado con datos históricos entonces no tiene sentido crear código para conectar el sistema con el broker para operar en la realidad.

```
set_commission(commission.PerShare(cost=0.0075, min_trade_cost=1))
```

4. Configuración de las Medias Móviles Simples

En este caso, esta parte del código se escribe dentro de la función `handle_data()`, la cual es llamada cada minuto, ya que las medias móviles simples van a ser calculadas a cada minuto así el algoritmo puede tomar una decisión cuando se cumplan las reglas.

Se puede observar que los datos de las medias móviles los guardamos dentro de la variable MM1 y MM2.

```
# Obtención de la serie histórica de datos
historial = data.history(context.titulo, 'price', 200, '1d')
historial_50 = historial[-50:]
historial_200 = historial
# Construcción de la media móvil de corto plazo
MM1 = historial_50.mean()
# Construcción de la media móvil de largo plazo
MM2 = historial_200.mean()
```

5. Configuración de variables a utilizar en el Algoritmo

Estas son las variables que vamos a utilizar para hacer los cálculos de cuántas acciones debe comprar el sistema cada vez que hace una operación.

```
# Identificar el actual precio del título
precio_actual = data.current(context.titulo,'price')
# Identificar actualmente cuantas posiciones en el título tenemos
posiciones_actuales = context.portfolio.positions[context.titulo].amount
# Identificar el efectivo que tenemos en la cuenta
efectivo = context.portfolio.cash
```

6. Lógica del algoritmo y ejecución de órdenes

Esta es la parte más importante del programa y es aquí donde reside la lógica del sistema, si la media móvil simple de 50 ruedas corta la media móvil simple de 200 ruedas en sentido ascendente y no tenemos ninguna posición en cartera, entonces el sistema compra automáticamente la cantidad de acciones que se pueda. Por lo contrario, si la si la media móvil simple de 50 ruedas corta la media móvil simple de 200 ruedas en sentido descendente, vende toda la posición si es que tuviéramos una abierta.

```
# Esta es la logica del algoritmo
if (MM1 > MM2) and posiciones_actuales == 0:
    cantidad_de_acciones = int(efectivo/precio_actual)
    order(context.titulo,cantidad_de_acciones)
    log.info('Comprando Acciones a %s!' % (precio_actual))

elif (MM1 < MM2) and posiciones_actuales != 0:
    order_target(context.titulo,0)
    log.info('Vendiendo Acciones! a %s'% (precio_actual))
```

7. Creación de un gráfico con el Precio y las Medias Móviles

En este caso la función `record()` sigue a las variables que dispongamos en su argumento y con ellas se crea un gráfico para visualizar los movimientos en el tiempo.

```
#Funcion que ayuda a construir un grafico segun las variables que sigamos
record(MM1=MM1,MM2=MM2,Price=precio_actual)
```

8. Código del Programa completo

Este es el código final del sistema, este programa puede ser iniciado una vez e interpretado por la computadora una vez por minuto hasta ser apagado.

```
def initialize(context):
    """
    Esta funcion es llamada una sola vez cuando el programa es iniciado.
    """
    # Seleccionar el titulo que queremos operar
    context.titulo = symbol('MCD')

    #Estas son las comisiones que nos pretederian cobrar el Agente
    set_commission(commission.PerShare(cost=0.0075, min_trade_cost=1))

def handle_data(context,data):
    """
    Esta función es llamada a cada minuto mientras el programa esté encendido.
    """
    # Obtencion de la serie historica de datos
    historial = data.history(context.titulo, 'price', 200, '1d')
    historial_50 = historial[-50:]
    historial_200= historial
    # Construccion de la media movil de corto plazo
    MM1 = historial_50.mean()
    # Construccion de la media movil de largo plazo
    MM2 = historial_200.mean()

    # Identificar el actual precio del titulo
    precio_actual = data.current(context.titulo,'price')

    # Identificar actualmente cuantas posiciones en el título tenemos
    posiciones_actuales = context.portfolio.positions[context.titulo].amount

    # Identificar el efectivo que tenemos en la cuenta
    efectivo = context.portfolio.cash

    # Esta es la lógica del algoritmo
    if (MM1 > MM2) and posiciones_actuales == 0:
        cantidad_de_acciones = int(efectivo/precio_actual)
        order(context.titulo,cantidad_de_acciones)
        log.info('Comprando Acciones a %s!' % (precio_actual))

    elif (MM1 < MM2) and posiciones_actuales != 0:
        order_target(context.titulo,0)
        log.info('Vendiendo Acciones! a %s' % (precio_actual))

    # Funcion que ayuda a construir el grafico segun las variables que sigamos
    record(MM1=MM1,MM2=MM2,Price=precio_actual)
```

V. Backtesting

A. Introducción

El proceso de evaluación de las rentabilidades que se espera de una estrategia de trading basada en los rendimientos calculados a partir de datos históricos se denomina BackTesting.²⁴ Gran cantidad de traders utilizan esta herramienta para demostrar que la estrategia ha funcionado en el pasado y establecen parámetros de confianza para que genere buenos resultados en tiempo real, no obstante hay que tener en cuenta que los componentes predictivos nunca serán absolutamente ciertos y que existen una serie de cuestiones inherentes al análisis de los rendimientos históricos que no se consiguen evaluar.

Cualquier estrategia se puede *adaptar* a un histórico de tal forma que los resultados que arrojen sean excepcionalmente positivos y robustos. Ajustando los criterios estratégicos dentro de un periodo específico se pueden alcanzar rendimientos que probablemente nunca se vuelvan a repetir en tiempo real. Sin una compensación adecuada de la liquidez de los precios, el BackTesting podría reflejar resultados beneficiosos que no se puedan reproducir en el futuro.

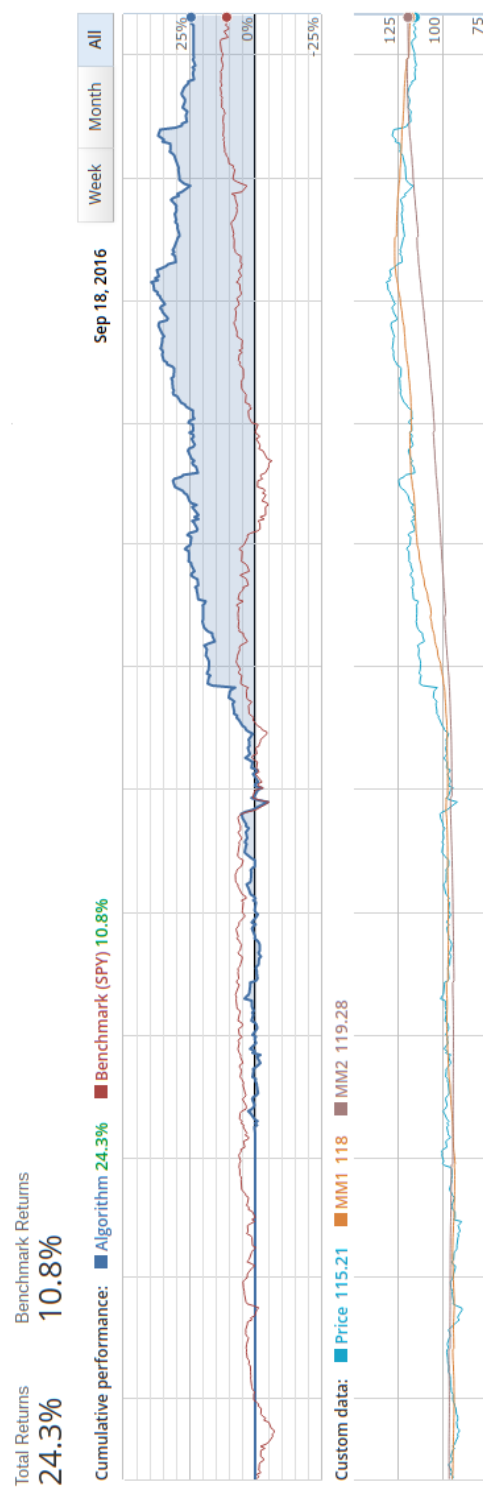
En definitiva, lo que se pretende es abordar la evaluación de la estrategia elegida para determinar si va a funcionar en el futuro, no obstante, el BackTesting señalará con total seguridad cómo habría funcionado en el pasado pero no brinda seguridad a la hora de establecer estrategias de cara al futuro.

B. Resultados

Para evaluar al sistema vamos a necesitar plantear varios supuestos, uno de ellos es el periodo mediante el cual se realizará el backtesting de la estrategia, este periodo estipulamos que será de dos años, y regirá desde el 20 de septiembre del 2014 hasta el 20 de septiembre del 2016, por otro lado, para que la simulación sea lo más real posible, utilizaremos un capital de 10.000 dólares, dado que en la realidad, esta es la cantidad de dinero que se necesita para abrir una cuenta en “Interactive Brokers”, broker online que Quantopian® ofrece en su plataforma para operar en tiempo real. Cabe aclarar, que en ningún lugar del código fuente del programa es declarado la cantidad de dinero que utilizaremos para el BackTesting, ya que en la plataforma del sitio existe un lugar para hacerlo antes de comenzar el proceso de BackTest, haciendo innecesario que tengamos que declararlo dentro del código del programa.

Ilustración 7. Rentabilidad del Backtesting

²⁴ ver Holland, “Issues Related to Back Testing.”

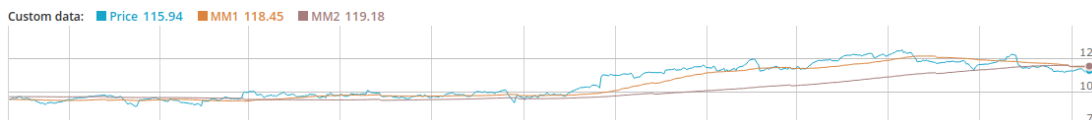


Fuente: Quantopian®

En el gráfico anterior se puede observar la rentabilidad que resulta del BackTesting comparado con los resultados para ese mismo periodo del índice de referencia o

Benchmark²⁵, Standard & Poor's 500, el cual se basa en la capitalización bursátil de 500 grandes empresas que poseen acciones que cotizan en las bolsas NYSE o NASDAQ. La rentabilidad resultante del sistema algorítmico asciende a un 24,3%, comparado con un 10,8% obtenido por mercado desde el inicio de la evaluación. Por otro lado, el gráfico ubicado abajo, monitorea el precio y las medias móviles que creamos con el sistema.

Ilustración 8. Gráfico del Precio y las Medias Móviles



Fuente: Quantopian®

El algoritmo produjo un total de diez operaciones de compra-venta siendo cinco de la compra de las acciones de MCD y cinco las ventas de las mismas. En el siguiente gráfico podremos ver el detalle de la cantidad de operaciones realizadas por el algoritmo durante los dos años.

Ilustración 9. Transacciones operadas por el Algoritmo

Date ▲ ▼	Security ▼	Transaction ▼	Quantity ▼	Last Sale Price ▼	\$ Amount ▼
2015-03-16 - 10:32 AM	MCD	BUY	103	\$96.83	\$9,973.59
2016-08-30 - 10:49 AM	MCD	SELL	-103	\$115.03	(\$11,847.58)
2016-08-30 - 10:51 AM	MCD	BUY	106	\$115.10	\$12,200.60
2016-08-30 - 10:53 AM	MCD	SELL	-106	\$114.88	(\$12,177.28)
2016-08-30 - 1:31 PM	MCD	BUY	106	\$115.04	\$12,194.66
2016-08-30 - 1:32 PM	MCD	SELL	-106	\$115.07	(\$12,197.42)
2016-08-30 - 1:33 PM	MCD	BUY	106	\$115.07	\$12,197.63
2016-08-30 - 1:35 PM	MCD	SELL	-106	\$115.02	(\$12,192.54)
2016-08-30 - 1:38 PM	MCD	BUY	106	\$115.09	\$12,199.75
2016-08-31 - 10:32 AM	MCD	SELL	-106	\$115.54	(\$12,247.24)

Fuente: Quantopian®

Si analizamos las transacciones podemos ver que la operación de compra iniciada el 16 de marzo del 2016 y la consecuente venta el 30 de agosto del mismo año, produjo un total de \$1873,99 dólares siendo esa la operación que reportó la mayor ganancia para el sistema.

²⁵ El *benchmark* es una técnica utilizada para medir el rendimiento de un sistema o componente del mismo, frecuentemente en comparación con el que se refiere específicamente a la acción de ejecutar un benchmark.

Ilustración 10. Detalle de la Rentabilidad alcanzada por periodos

RETURNS	1 MONTH	3 MONTH	6 MONTH	12 MONTH
September 2014	0	N/A	N/A	N/A
October 2014	0	N/A	N/A	N/A
November 2014	0	0	N/A	N/A
December 2014	0	0	N/A	N/A
January 2015	0	0	N/A	N/A
February 2015	0	0	0	N/A
March 2015	0.007	0.007	0.007	N/A
April 2015	-0.009	-0.003	-0.003	N/A
May 2015	-0.007	-0.009	-0.009	N/A
June 2015	0	-0.016	-0.01	N/A
July 2015	0.05	0.0429	0.04	N/A
August 2015	-0.048	-0.001	-0.01	-0.01
September 2015	0.045	0.045	0.028	0.035
October 2015	0.137	0.131	0.18	0.176
November 2015	0.017	0.209	0.208	0.197
December 2015	0.041	0.2039	0.258	0.246
January 2016	0.046	0.108	0.253	0.304
February 2016	-0.052	0.033	0.249	0.236
March 2016	0.08	0.0709	0.29	0.326
April 2016	0.005	0.029	0.1409	0.3459
May 2016	-0.034	0.049	0.083	0.309
June 2016	-0.007	-0.036	0.033	0.299
July 2016	-0.022	-0.062	-0.0351	0.21
August 2016	-0.02	-0.048	-0.002	0.246
September 2016	0.007	-0.034	-0.0691	0.201

Fuente: Quantopian®

Ilustración 11. Pérdida máxima desde el máximo alcanzado

MAX DRAWDOWN	1 MONTH	3 MONTH	6 MONTH	12 MONTH
September 2014	0%	N/A	N/A	N/A
October 2014	0%	N/A	N/A	N/A
November 2014	0%	0%	N/A	N/A
December 2014	0%	0%	N/A	N/A
January 2015	0%	0%	N/A	N/A
February 2015	0%	0%	0%	N/A
March 2015	-2.3%	-2.3%	-2.3%	N/A
April 2015	-3%	-4.5%	-4.5%	N/A
May 2015	-4.8%	-4.8%	-4.8%	N/A
June 2015	-2.3%	-6%	-6%	N/A
July 2015	-3%	-6%	-6%	N/A
August 2015	-9.7%	-9.7%	-9.7%	-9.7%
September 2015	-2.6%	-9.7%	-9.7%	-9.7%
October 2015	-2.1%	-9.7%	-9.7%	-9.7%
November 2015	-3.3%	-3.3%	-9.7%	-9.7%
December 2015	-1.5%	-3.3%	-9.7%	-9.7%
January 2016	-3.3%	-4%	-9.7%	-9.7%
February 2016	-7.2%	-7.2%	-7.2%	-9.7%
March 2016	-1.8%	-7.2%	-7.2%	-9.7%
April 2016	-2.5%	-7.2%	-7.2%	-9.7%
May 2016	-7.01%	-7.01%	-7.2%	-9.7%
June 2016	-5.6%	-10.7%	-10.7%	-10.7%
July 2016	-8%	-10.7%	-10.7%	-10.7%
August 2016	-4.1%	-10.4%	-12%	-12%
September 2016	0%	-10.4%	-12%	-12%

Fuente: Quantopian®

Con estos datos que pudimos rescatar del proceso de evaluación podemos comenzar a hacer una evaluación de la Fiabilidad, la Estabilidad y Volatilidad:

1. Fiabilidad:

Para medir la fiabilidad del sistema podemos hacerlo a través del Win Ratio²⁶:

$$Win\ Ratio = \left(\frac{Operaciones\ Ganadoras}{Operaciones\ Perdedoras} \right) \times 100 = \left(\frac{2}{5} \right) = 40\%$$

Es bastante común en el mercado de analistas cuánticos que se busque un nivel de eficacia de alrededor de un mínimo 60-65%, y la estrategia de cruce de medias simples solo obtuvo un resultado del 40%, lo que la hace no confiable.

2. Estabilidad del sistema:

Este indicador muestra que la mayor pérdida alcanzada desde el máximo alcanzado en nuestra cartera fue de 12%. En algún momento del periodo ejecutado nuestra posición cayó un 12%, lo cual no pone en demasiado riesgo la cartera.

3. Volatilidad del Activo frente al mercado:

Este indicador es medido a través del coeficiente Beta (β), cuanto más alto es el Beta, más volátil es el activo y si es igual a 1 equivale al mercado, el valor que arrojan los datos es:

$$\beta_{im} = \frac{Cov(r_i - r_m)}{Var(r_m)} = 0,54 \text{ Siendo } i \text{ el activo y } m \text{ el mercado de referencia.}$$

²⁶ El Win Ratio es el índice de ganancias sobre pérdidas

VI. Conclusión

En este apartado quisiera llegar a una conclusión sobre los principales aspectos en el desarrollo del sistema algorítmico de trading. Incluso aunque la estrategia no cumpla los niveles ideales de fiabilidad y estabilidad como es el caso de la estrategia que hemos elaborado, es posible que resulte ganadora. De igual manera, el hecho de cumplir con esos índices establecidos nunca asegura que finalmente se trate de un sistema ganador.

Este tipo de estrategia aquí desarrollada es comúnmente utilizada, por lo que usualmente no suele ser ganadora, una de las razones por lo cual podríamos criticarla es porque el cruce de medias móviles simples se basa en dos pilares fundamentales, uno es que se basa en un periodo determinado de cotizaciones independientemente de los valores que le hayan precedido y también al hecho de que se asigna la misma importancia a todas las cotizaciones del periodo. La evolución natural deriva en la utilización de medias móviles ponderadas para solventar el problema de dotar con mayor importancia relativa a las últimas cotizaciones aunque no resuelve el problema de la independencia de las cotizaciones en el periodo establecido.

Es por esto que considero que se debería utilizar la estrategia de medias móviles exponenciales, porque está basada en la ponderación de los datos que la compone y tiene en cuenta todos los datos disponibles del valor que intervienen en el cálculo, de esta manera se encuentra una medida para salvar los inconvenientes que tienen tanto la media simple como la ponderada.

Pero esto no es todo, considero que para que los algoritmos de trading sean efectivos tienen que combinar varias estrategias, una simple estrategia como la presentada ya no es suficiente como para garantizar un rendimiento. La tecnología está avanzando tanto que ya los inversores pueden concentrar todas sus energías en la construcción de una estrategia confiable, sin tener que preocuparse por la arquitectura en general del sistema. Como pudimos ver en esta Tesina, construimos nuestro desarrollo sobre un sistema hecho por terceros, que está lejos de ser el más eficiente pero quedó demostrado que permite enfocarse en lo más importante, que es la búsqueda de una estrategia eficiente.

Referencias

- Avellaneda, Marco. "Algorithmic and High-Frequency Trading: An Overview," n.d.
<https://www.math.nyu.edu/faculty/avellane/QuantCongressUSA2011AlgoTradingLAST.pdf>.
- Biais, B., and P. Woolley. "High Frequency Trading." *Manuscript, Toulouse University, IDEI*, 2011. <http://www.eifr.eu/files/file2220879.pdf>.
- Holland, M. "Issues Related to Back Testing." *Financial Trading*, 2013.
<http://www.financialtrading.com/issues-related-to-back-testing/>.
- John R. Hill, George Pruitt And Lundy Hill. *The Ultimate Trading Guide*, 2000.
- Jovanovic, Boyan, and Albert J. Menkveld. "Middlemen in Limit Order Markets 1," n.d.
- Le, D. "Automated Trading System," 2009.
<http://eprints.nottingham.ac.uk/22657/2/>.
- Marcelo, Jlm, and Mmm Quirós. "Estabilidad de Las Estrategias de Inversión Con Medias Móviles." *Spanish Journal of Finance and*, 2005.
<http://www.tandfonline.com/doi/abs/10.1080/02102412.2005.10779564>.
- Murphy, J. J. "Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications," 1999.
https://googledrive.com/host/oBorcwmcuCEWpQTYodjNCRjlqQVE/Murphy_I.pdf.
- Reid, Stuart Gordon. "Algorithmic Trading System." *Open Source Algorithmic Trading Architectures (OSATA)*, September 29, 2013.
- Welles Wilder, J., Jr. *New Concepts in Technical Trading Systems*. Dickinson School of law - Law Library, 1978.
- Tabb, Larry, "Institutional Equity Trading in America: A Buy-Side Perspective," consulting report, The Tabb Group, April 2004.