



Facultade de Informática

UNIVERSIDADE DA CORUÑA

TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

Evaluador de estrategias de mercado

Estudiante: Alejo Nahuel Cedeira Vidal

Dirección: Paula María Castro Castro
José Pablo González Coma

A Coruña, xuño de 2020.

A mis padres y a mi hermana

Agradecimientos

A mis directores de proyecto José Pablo González Coma y Paula María Castro Castro por su ayuda, atención y amabilidad durante todos estos meses.

Resumen

La bolsa de valores es un espacio donde las empresas venden o compran acciones. Desde sus inicios, se ha buscado sacar la máxima rentabilidad posible a las inversiones. Para analizar el mercado e intentar predecir los movimientos futuros del activo, se creó el Análisis Técnico. En este Trabajo Fin de Grado se presenta una aplicación web que permite al usuario generar estrategias de inversión mediante una interfaz gráfica intuitiva y sencilla. La base de la aplicación es la utilización de indicadores técnicos para analizar el activo e identificar el instante en el que realizar acciones de compra y venta. Finalmente, se le proporciona al usuario una interfaz donde puede simular las estrategias que ha programado y compararlas entre sí.

Abstract

The stock market is a space where companies sell or buy stocks. Since its inception, we have sought to get the maximum possible return on investments. To analyze the market and try to predict the future movements of the asset, the Technical Analysis was created. In this Final Degree Project, a web application that allows the user to generate investment strategies through an intuitive and simple graphical interface is presented. The basis of the application is the use of technical indicators to analyze the asset and identify the moment in which to carry out purchase and sale actions. Finally, the user is provided with an interface where he or she can simulate the strategies he or she have programmed and compare them with each other.

Palabras clave:

- Bolsa
- Activo
- Acción
- Índice
- Indicador

Keywords:

- Stock market
- Asset
- Stock
- Index
- Indicator

Índice general

1	Introducción	1
1.1	Introducción	1
1.2	Motivación	2
1.3	Objetivos	5
1.4	Estructura de la memoria	6
2	Fundamentos del Análisis Técnico	7
2.1	Conceptos previos	7
2.2	Análisis bursátil	8
2.3	Definición del Análisis Técnico	9
2.4	Análisis cuantitativos e indicadores técnicos	10
3	Fundamentos tecnológicos	23
3.1	Control de versiones	23
3.2	Diseño	23
3.3	Entorno de desarrollo	24
3.4	Lenguajes de desarrollo	24
4	Metodología	27
4.1	Metodología Scrum	28
4.1.1	Roles	28
4.1.2	Eventos	28
4.1.3	Artefactos	29
4.2	Scrum aplicado al proyecto	30
4.2.1	Roles	30
4.2.2	Eventos	30
4.2.3	Artefactos	31

5	Planificación y costes	35
5.1	Planificación del proyecto	35
5.1.1	Planificación inicial	35
5.1.2	Seguimiento	36
5.2	Recursos utilizados	38
5.3	Estimación de costes	38
5.3.1	Seguimiento de los costes	40
6	Análisis de requisitos	41
6.1	Requisitos funcionales	41
6.1.1	Generación del flujo principal	41
6.1.2	Visualización de los datos.	41
6.2	Requisitos no funcionales	42
6.3	Actores	42
6.4	Historias de usuario	42
6.4.1	Flujo	43
6.4.2	Interfaz	46
7	Diseño	49
7.1	Estructura	49
7.1.1	Fuente de datos	49
7.1.2	Back-end	50
7.1.3	Front-end	51
8	Implementación	53
8.1	Implementación	53
8.1.1	Índices y acciones	53
8.1.2	Indicadores	54
8.1.3	Estrategias	60
8.1.4	Interfaz	63
8.2	Interfaz Node-RED	64
8.3	Interfaz del usuario	67
8.4	Instalación	69
9	Pruebas	71
9.1	Pruebas funcionales	71
9.1.1	Inyectores	71
9.1.2	Indicadores	71
9.1.3	Estrategias	72

9.1.4 Visualización	72
10 Conclusiones	73
10.1 Líneas futuras de trabajo	74
Lista de acrónimos	75
Bibliografía	77

Índice de figuras

1.1	Interfaz Metatrader.	3
1.2	Interfaz Plus500.	4
1.3	Interfaz TradingView.	4
2.1	Indicador de Medias Móviles.	11
2.2	Indicador RSI.	12
2.3	Indicador MACD.	14
2.4	Indicador Estocástico.	15
2.5	Bandas de Bollinger.	17
2.6	Indicador CCI.	18
2.7	Variación de porcentaje.	19
2.8	Indicador Ichimoku.	21
2.9	Indicador OBV.	22
5.1	Diagramas de Gantt	37
6.1	Historias de usuario en el flujo,	43
6.2	Seleccionar índice.	44
6.3	Conexión.	44
6.4	Inversión total.	44
6.5	Seleccionar indicador/es.	44
6.6	Parámetros indicador/es.	45
6.7	Unión con Realizar estrategia.	45
6.8	Inversión inicial.	45
6.9	Flujo completo.	46
6.10	Inyectar datos.	46
6.11	Historias de usuario en la interfaz.	46

7.1	Nodo Indicador.	50
7.2	Nodo <i>Definir estrategia</i>	51
7.3	Nodo <i>Realizar estrategia</i>	51
7.4	Nodo <i>Gráfica</i>	52
8.1	Índices y acciones.	54
8.2	Valores requeridos en el inyector.	54
8.3	Interior del indicador.	55
8.4	Variables de un indicador.	56
8.5	Variables del indicador MACD.	57
8.6	Variables del indicador Ichimoku.	58
8.7	Variables del indicador Estocástico.	58
8.8	Variables del indicador RSI.	59
8.9	Variables del indicador Variación de porcentaje.	59
8.10	Variables del indicador Movimiento por fecha.	59
8.11	<i>Definir estrategia</i>	60
8.12	Inversión total.	61
8.13	Contenido <i>Definir estrategia</i>	61
8.14	<i>Realizar estrategia</i>	61
8.15	Inversión inicial.	61
8.16	Contenido <i>Realizar estrategia</i>	62
8.17	<i>Gráfica</i>	63
8.18	Contenido nodo <i>Gráfica</i>	63
8.19	Interfaz Node-RED.	64
8.20	Flujo antes de pulsar <i>Deploy</i>	65
8.21	Implementación de una estrategia.	66
8.22	Varias estrategias.	66
8.23	Interfaz del usuario.	67
8.24	Directo.	68
8.25	Día a día.	68
8.26	Comando Docker.	70

Índice de tablas

1.1	Comparación de aplicaciones.	2
5.1	Resumen de la duración del proyecto en horas.	39
5.2	Estimación del coste de los recursos humanos.	39
5.3	Coste total del proyecto.	40
5.4	Seguimiento de costes.	40
5.5	Seguimiento de costes humanos.	40
5.6	Seguimiento del coste total del proyecto.	40
6.1	Prioridades de los casos de uso.	43

Introducción

1.1 Introducción

La bolsa de valores, también conocida como **la bolsa**, es un mercado donde las empresas que requieren financiación se exponen a personas u organizaciones, las cuales buscan invertir dinero para obtener una rentabilidad. Estas inversiones se realizan sobre las acciones de las empresas del mercado.

Desde la creación de la bolsa de valores y de sus primeras inversiones, siempre se ha sentido la necesidad de predecir lo que va a ocurrir en un futuro, para así hacer nuestras inversiones lo más rentables posibles. Para invertir, debemos analizar previamente dónde lo vamos a hacer, ya que un buen análisis fundamental de la empresa o empresas va a ser crucial a la hora de confiar en su crecimiento. A partir de ahí, el inversor tiene que ser consciente de los riesgos que asume y de las consecuencias que esto puede conllevar. Este riesgo es otro factor importante a la hora de invertir, pues determina el grado de incertidumbre del comportamiento del activo en el futuro.

El análisis técnico tuvo sus orígenes en EE.UU. a finales del siglo XIX cuando **Charles Henry Dow** [1] creó la Teoría de Dow. Esta teoría se basa en predecir lo que va a pasar utilizando indicadores técnicos, los cuales permiten analizar la tendencia que va a seguir un activo y así estudiar las posibilidades de compra y venta que se pueden tener. En la actualidad, nos encontramos con un gran abanico de indicadores que permiten realizar un buen análisis. Se pueden clasificar en indicadores de tendencia (por ejemplo, el **MACD**), de momentum (por ejemplo, el **RSI**), de volatilidad (por ejemplo, **Bandas de Bollinger**) y de volumen (por ejemplo, el **OBV**). Esos son los más utilizados, aunque hay otros también muy utilizados como las medias móviles y el indicador **Ichimoku**.

Una mala estrategia puede llevar al inversor a arruinarse rápidamente, mientras que otros inversores pueden volverse ricos. Una estrategia de inversión nunca va a ser ideal ni va a asegurar una rentabilidad positiva. En cambio, tenemos la oportunidad de ayudarnos de diferentes herramientas que permiten analizar el mercado y ayudan a tomar decisiones que en un futuro pueden ser beneficiosas.

1.2 Motivación

Hoy en día hay muchas aplicaciones que permiten implementar estos indicadores y realizar operaciones automáticamente. Las más conocidas son:

	Metatrader	Plus500	TradingView	TFG
Crear estrategias	✓	✓	✓	✓
Configurar indicadores	✓	X	X	✓
Gráficos interactivos	✓	✓	✓	✓
Comparar estrategias	X	X	X	✓
Estrategia con múltiples indicadores	✓	X	X	✓
Avances diarios	✓	✓	✓	✓
Tablas resumen	X	X	X	✓
Diseño de la estrategia mediante interfaz gráfica	X	X	X	✓
Gratuita	✓	X	X	✓

Tabla 1.1: Comparación de aplicaciones.

Metatrader [2] es la aplicación para simulación de estrategias más conocida en la actualidad. Está presente en prácticamente todas las aplicaciones que realizan operaciones en bolsa. Nos ofrece diversas funcionalidades ya que permite programar una estrategia de manera sencilla y utilizar un gran número de indicadores con los que probar diferentes combinaciones. Es gratuita y fácil de instalar en diferentes entornos. La configuración de parámetros de los indicadores y de la estrategia se realiza mediante cuadros de texto y botones, pero no tenemos opción de crearlo gráficamente. No nos permite personalizar en los indicadores las acciones de compra y venta, sólo se puede variar el periodo que se necesita para crearlo.

Como podemos observar en la figura 1.1, Metatrader presenta una compleja interfaz con muchas ventanas que puede llegar a confundir al usuario que no esté acostumbrado a usar este tipo de aplicaciones. También destaca la configuración de la estrategia, que se limita a definición de parámetros.

CAPÍTULO 1. INTRODUCCIÓN

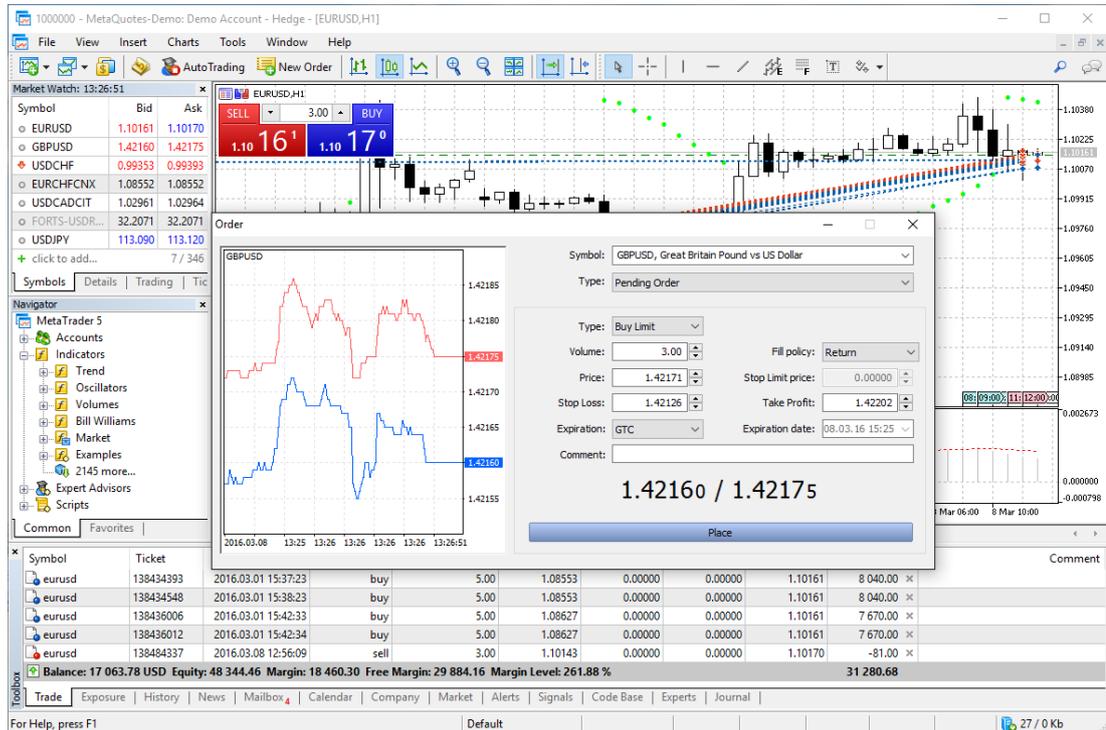


Figura 1.1: Interfaz Metatrader.

Otra aplicación de trading muy popular es **Plus 500** [3]. Esta aplicación está más orientada a ejecutar acciones de compra y venta y no a analizar el mercado. Con ella se pueden generar diferentes estrategias de inversión, pero presenta escasez de indicadores técnicos. Es de pago, pero cuenta una demo gratuita de prueba. Su única función es generar estrategias y ver cómo se comporta el mercado, pero no se pueden configurar indicadores técnicos ni diseñar las estrategias gráficamente. Las acciones de compra y venta las realiza el usuario y sólo puede automatizarlas cuando el activo llegue a algún valor en concreto o cuando pasa un cierto número de días. Los indicadores sólo son informativos, no se permite la ejecución automática de las acciones de compra o venta que proporcionan.

En la figura 1.2 podemos observar la interfaz gráfica de Plus500, es parecida a la de Metatrader, pero algo más sencilla ya que se centra única y exclusivamente a programar señales de compra y venta en el momento que el activo llegue a cierto valor. Esto se realiza en la parte superior de la interfaz.

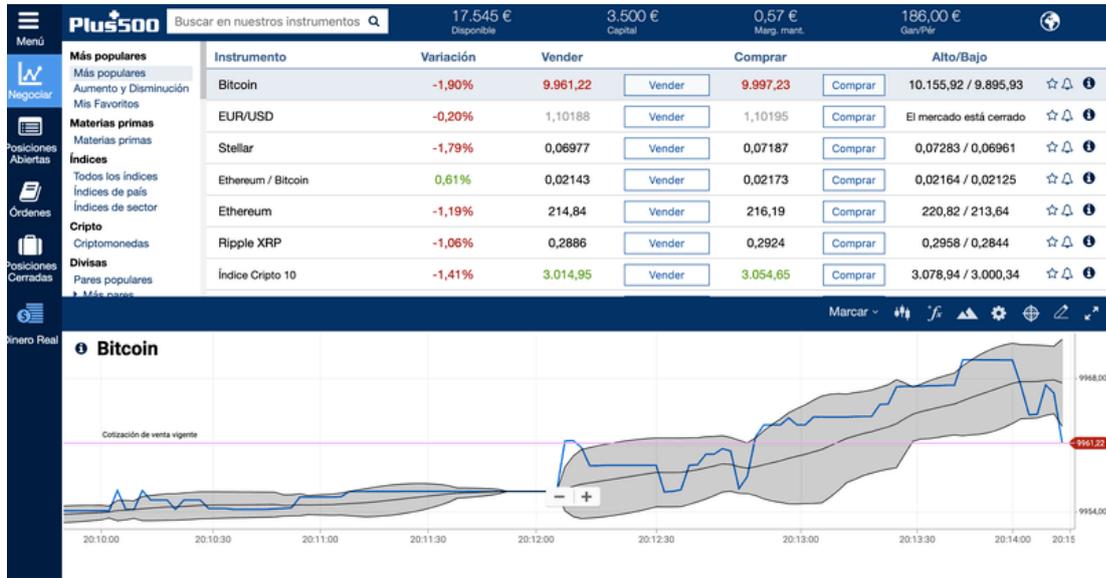


Figura 1.2: Interfaz Plus500.



Figura 1.3: Interfaz TradingView.

TradingView [4] es otra aplicación para analizar el mercado de bolsa. Igual que Plus500, presenta escasez de indicadores técnicos. Es de pago y no necesita instalación, ya que se utiliza desde el navegador. Esta aplicación permite simular estrategias de inversión ya conocidas o crear una propia, pero mediante código de programación, donde ya se tendrían que tener más conocimientos sobre estrategias. Para los indicadores, sólo deja utilizar los que tiene incluidos y no permite configurar sus valores. No permite automatizar acciones de compra y venta con los indicadores, estos son sólo informativos.

En la figura 1.3, podemos observar la compleja interfaz de TradingView. Esta interfaz busca englobar muchos puntos del análisis técnico, lo que afecta negativamente a la visualización de los datos y a la realización de estrategias.

En definitiva, como vemos en la tabla 1.1, la motivación principal de esta aplicación es poder realizar comparaciones de estrategias de manera sencilla. Todo esto, partiendo de un sencillo diseño gráfico, intuitivo y rápido que nos permite generar las estrategias y personalizar los indicadores con libertad sin tener conocimientos sobre ecuaciones matemáticas o programación. El usuario podrá crear estrategias de inversión utilizando diferentes indicadores técnicos y diferentes cantidades iniciales, con lo que podrá comparar las rentabilidades de las estrategias en cualquier punto del periodo temporal elegido.

1.3 Objetivos

El objetivo principal de este proyecto es el desarrollo de una aplicación web que permita realizar estrategias de inversión y compararlas entre sí. Para programar las estrategias, lo único necesario es crear un flujo con los nodos siguiendo un orden definido.

Esta aplicación está destinada a un público con ciertos conocimientos básicos de inversión en bolsa. No hace falta tener conocimientos de programación para utilizar la aplicación, ya que dispone de una pantalla intuitiva que crea estrategias de inversión sólo con la conexión de nodos.

Los objetivos que se pretenden conseguir con esta aplicación son:

- Desarrollo de una aplicación web que permita realizar diferentes estrategias de inversión y poder compararlas entre sí.
- Desarrollo de los indicadores técnicos más conocidos del Análisis Técnico bursátil.
- Implementación de una interfaz gráfica sencilla, dándole el protagonismo a la visualización de las estrategias.

- Capacidad de simular estrategias día a día.
- Disponibilidad de datos cualitativos (a través de gráficas) y cuantitativos (a través de tablas).
- Realización de estrategias con cualquier tipo de activo.
- Exportación de las gráficas en diferentes formatos.

1.4 Estructura de la memoria

En esta sección se describen brevemente los capítulos en los que se ha estructurado la memoria:

- **Capítulo 1: Introducción.** En este capítulo se hace una breve introducción a las inversiones en bolsa, la motivación, los objetivos que se quieren conseguir con este trabajo y la estructura de esta memoria.
- **Capítulo 2: Fundamentos del Análisis Técnico.** Se aborda el tema sobre el que se basa el proyecto, explicando en qué consiste el Análisis Técnico bursátil y cómo se realiza.
- **Capítulo 3: Fundamentos tecnológicos.** Se detallan las herramientas usadas en la realización de este proyecto.
- **Capítulo 4: Metodología.** Se explica en qué consiste una metodología, qué tipos hay, qué elementos tiene y por qué se ha optado por Scrum como metodología de trabajo.
- **Capítulo 5: Planificación y costes.** Se detalla la planificación inicial de duración y costes así como el seguimiento que se ha realizado.
- **Capítulo 6: Análisis de requisitos.** Se describen los requisitos de la aplicación.
- **Capítulo 7: Diseño.** Se explican los aspectos importantes del diseño de la aplicación.
- **Capítulo 8: Implementación.** Se detallan aspectos importantes de la implementación de la aplicación.
- **Capítulo 9: Pruebas.** Se comentan teóricamente las pruebas realizadas de los casos de uso de la aplicación.
- **Capítulo 10: Conclusiones.** Se resumen las funcionalidades de la aplicación y los objetivos que se han cumplido. También se aportan líneas futuras para la implementación de la aplicación.

Fundamentos del Análisis Técnico

En este capítulo nos vamos a centrar en explicar, de manera breve, todos los conceptos que nos van a ser útiles para poder entender el desarrollo de la aplicación.

2.1 Conceptos previos

Para poder analizar y comparar estrategias de inversión, en primer lugar, tenemos que introducir una serie de conceptos fundamentales a la hora de referirnos a inversiones en Bolsa. Una **acción** es una de las partes iguales en las que se divide el capital dentro de una sociedad anónima. Estas partes son poseídas por una persona, que recibe el nombre de accionista, y representa la propiedad que la persona tiene de la empresa.

Una de las características principales que debemos tener en cuenta a la hora de invertir en Bolsa es que, a diferencia de otros sistemas de inversión, su rentabilidad es variable. Los precios de las acciones que se poseen varían con el tiempo, por lo tanto, desconocemos cuál va a ser su rentabilidad, que incluso puede ser negativa. Los precios de las acciones pueden permanecer estables durante cierto periodo de tiempo, o pueden moverse rápidamente. La medida que indica la velocidad con la que fluctúa una acción se conoce como **volatilidad**.

En la cotización diaria la acción toma diferentes valores a lo largo de la jornada bursátil. Es decir, cada día la acción registrará un **precio de apertura**, un **precio máximo**, un **precio mínimo**, un **precio de cierre** y un **volumen**. Los precios son los valores que tiene el activo en este instante, en cambio, el **volumen** es la cantidad de un activo concreto en el que se invierte durante un período de tiempo. Otra manera de invertir de manera global en bolsa es a partir de los índices bursátiles.

Un **índice**[5] es un promedio calculado de precios de acciones seleccionadas que pueden representar un mercado o un sector concreto. Estos índices permiten al inversor realizar un seguimiento de sus acciones dentro de un mercado y sirven como punto de referencia útil para medir el éxito o fracaso de medios de inversión como carteras o fondos.

Existen índices prácticamente para cada sector de la economía y del mercado de acciones. Los más importantes son:

- El **IBEX 35** [6] es el índice español más importante, ya que reúne los principales valores del mercado bursátil nacional. Como ocurre con la mayoría de índices, las empresas con mayor capitalización bursátil tienen mayor peso dentro del índice y sus alzas y bajas influirán en mayor medida en el movimiento final de índice.
- El **Standard & Poor's 500** (S&P 500) [7], considerado el índice más representativo de la situación real del mercado, no sólo de Estados Unidos, sino también del mundo.
- El **Dow Jones Industrial Average** (DJIA) [8] es un índice americano que reúne las 30 compañías industriales más importantes de Estados Unidos y, por lo tanto, del mundo.
- El **NASDAQ** (National Association of Securities Dealers Automated Quotation) [9] es la bolsa de valores electrónica automatizada más grande de los Estados Unidos y la segunda bolsa más grande del mundo.
- El **Euro Stoxx 50** [10] es un índice bursátil que representa el rendimiento de las 50 empresas más grandes entre los 19 supersectores en términos de capitalización de mercado en 11 países de Europa.

El objetivo de cualquier persona que invierte en este mercado es obtener la mayor rentabilidad con el menor riesgo posible. Para ello se emplean diferentes técnicas de análisis, indicadores, instrumentos financieros, etc. En las siguientes secciones desarrollamos algunas de estas ideas detalladamente.

2.2 Análisis bursátil

Los mercados de capitales han supuesto desde sus inicios un gran atractivo para los inversores. Por ello, muchos economistas, investigadores o inversores, han propuesto diferentes

métodos de análisis bursátil con el objetivo de tomar decisiones fundadas en base a información relevante que pueda evitar, en la medida de lo posible, la aleatoriedad de la rentabilidad alcanzada en una inversión.

En definitiva, el análisis bursátil es la disciplina que estudia los movimientos de las cotizaciones bursátiles a través de gráficos, balances, series temporales, series estadísticas u otros tipos de teorías. Cuando hablamos de análisis bursátil estamos haciendo referencia a un campo muy amplio, ya que existen muchos tipos de análisis desde los que podemos abordar el estudio del mercado. La mayoría de los profesionales del sector financiero utilizan el **Análisis Fundamental** y el **Análisis Técnico**.

- El Análisis Fundamental [11] consiste en analizar y estudiar una empresa en función de su valor y del que puede tener en un futuro. Para realizar un buen análisis fundamental se deben tener conocimientos acerca del análisis de balances, cuentas y todo lo relacionado con la empresa, así como estar al tanto de noticias que les pueden afectar.
- El Análisis Técnico 2.3 no se basa en analizar las cuentas de la empresa y su valor, sino en determinar su precio y como se moverá este en un futuro. Consiste en analizar gráficos de cotizaciones y con ello intentar determinar si el precio de un acción subirá o bajará, con la intención de comprar o vender en un buen momento.

2.3 Definición del Análisis Técnico

El Análisis Técnico [12] para Bolsa tiene sus orígenes en 1880 y fue introducido por Charles H. Dow [1]. Esta importante herramienta de análisis bursátil tiene como idea principal el estudio objetivo del **Precio** y del **Volumen** de algún valor. En particular, estudia patrones teniendo en cuenta los valores pasados del activo.

El Análisis Técnico se basa en una serie de premisas:

- El precio y el mercado lo descuentan todo: el Análisis Técnico sostiene que lo único que nos debe importar es si el mercado sube o baja, no las razones por las cuales ha ocurrido.
- La tendencia: este análisis se basa en el estudio de las tendencias y en su identificación, con el fin de obtener la máxima rentabilidad.
- Memoria de mercado: podemos decir que el mercado tiene memoria y nada nos garantiza que algo ocurrido en el pasado no pueda volver a suceder.

Disciplinas del Análisis Técnico

Dentro de este análisis podemos diferenciar dos vertientes, una en relación al estudio del gráfico de un valor y las tendencias de este, y otra vertiente que se basa en indicadores técnicos y modelos que son realizados por algoritmos matemáticos y estadísticos.

El **Análisis Chartista** se basa en el estudio de gráficas de cualquier mercado y las figuras que se forman, dividiéndose a la vez en figuras de **continuación de tendencia** y de **cambio de tendencia** [13]. El **Análisis Cuantitativo** 2.4 nos ayuda a crear estrategias o a apoyarnos en el estudio del análisis chartista a partir de indicadores. Sólo nos ofrece señales para interpretar el mercado.

2.4 Análisis cuantitativos e indicadores técnicos

En este apartado vamos a comentar los indicadores técnicos que vamos a utilizar en nuestro proyecto.

Medias Móviles

También conocida como **MA** [14] es el indicador técnico de tendencia más utilizado en el Mercado de Valores. La Media Móvil en análisis técnico se calcula sumando el valor de un número (n) de cotizaciones y dividiendo su resultado por ese número (n) de cotizaciones. Si este proceso se realiza continuamente da como resultado una gráfica más suavizada que la cotización propia del valor.

La principal finalidad de este indicador es la de suavizar las gráficas y eliminar el ruido producido por movimientos bruscos en el precio, proporcionándonos una visión más limpia de la tendencia del valor. Este indicador lo vamos a usar para confirmar una tendencia, no para detectar cambios en las tendencias ni para detectar figuras en el gráfico. En este proyecto nos centraremos en dos tipos:

- **Media Móvil Simple** o **SMA**. En esta media todos los periodos están ponderados de la misma forma. Debemos tener en cuenta que cuanto mayor sean los (n) periodos de la Media Móvil, más suavizada estará y más tardará en reaccionar a los cambios de tendencia. Esta media tarda más en reaccionar que la Media Exponencial que veremos a continuación. Una buena práctica para utilizar esta media es crear varias para tener información a diferentes temporalidades.

$$SMA = \frac{\sum_{i=1}^n \text{Precio de cierre}(i)}{n}. \quad (2.1)$$

- **Media Móvil Exponencial** o **EMA**. La principal diferencia con la **SMA** es que las últimas cotizaciones tienen mayor peso en la ponderación que las anteriores. Se trata de una ponderación progresiva y ascendente. Es decir, utiliza toda la información de la serie histórica, igual que la simple, pero le da mayor importancia a los últimos datos.

$$EMA(t) = \alpha \times \text{Precio actual} + (1 - \alpha) \times EMA(t - 1), \quad (2.2)$$

donde α es una constante suavizada con un valor entre 0 y 1 y $EMA(t - 1)$ es la EMA para el periodo anterior.



Figura 2.1: Indicador de Medias Móviles.

En la figura 2.1 podemos ver la línea azul que nos indica el activo y dos medias, una **SMA** de color amarillo y una **EMA** de color rojo. Una vez que tenemos calculadas las Medias Móviles, podemos generar ciertas operaciones básicas que nos van a ayudar a la hora de realizar nuestra estrategia:

- Una media móvil nos da una señal de compra en el momento en el que el precio la rompe de abajo hacia arriba.
- Una media móvil nos da una señal de venta en el momento en el que el precio rompe dicha media desde arriba hacia abajo.

RSI

Fue desarrollado por J.Welles Wilder [15] y es uno de los indicadores del Análisis Técnico más usado. También llamado **Índice de Fuerza Bruta** o **RSI** [16], nos indica la fuerza interna que tiene cada activo en cada momento en relación a la tendencia. Su función principal consiste en darnos señales de sobrecompra y sobreventa. Se trata de un oscilador, funciona mejor en el entorno de mercados laterales si lo que queremos es que nos dé señales de entrada y salida al mercado.

$$RSI = 100 - \frac{100}{1 + RS}, \quad (2.3)$$

donde

$$RS = \frac{\text{Promedio ganancias}}{\text{Promedio pérdidas}}. \quad (2.4)$$

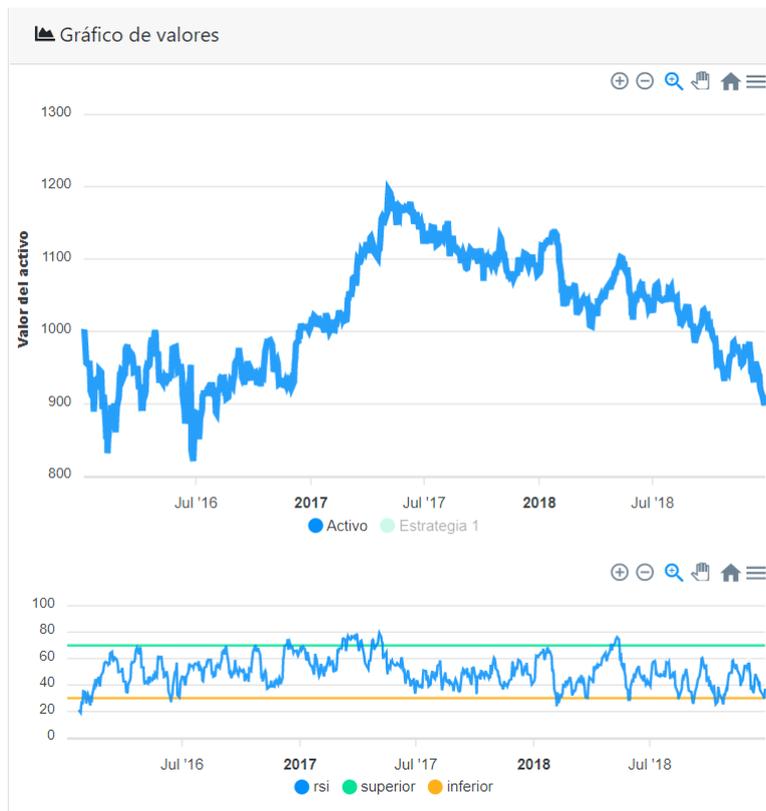


Figura 2.2: Indicador RSI.

En la figura 2.2 podemos observar en la parte superior el activo y en la parte inferior distinguimos 3 líneas, una superior en verde que nos marca el límite superior, una inferior en amarillo que nos marca el límite inferior y en azul la línea del indicador **RSI**.

Es un indicador que nos sirve como apoyo para tomar decisiones acerca de acciones como vender o comprar. Debemos ajustar un límite inferior y un límite superior que, en el momento en el que el indicador *RSI* los corte, tendremos señales de sobrecompra y sobreventa. Por lo tanto, para este indicador identificamos:

- Una señal de venta cuando el indicador cruza el límite superior (por defecto 70).
- Una señal de compra cuando el indicador cruza el límite inferior (por defecto 30).

Un aspecto a considerar es la distancia entre los valores límites. Por ejemplo, considerando 80 y 20 en lugar de los valores por defecto tendremos menos señales de compra y venta pero, a su vez, serán más fiables.

MACD

El *MACD* [17] es un indicador técnico retardado que se forma mediante la diferencia de dos medias móviles exponenciales, una de corto plazo con 12 sesiones y otra de largo plazo con 26. Es un indicador más efectivo en periodos en los que un valor se encuentra en tendencia y en mercados con grandes oscilaciones. Otro parámetro es *Signal* 2.6, que consiste en aplicarle una media móvil exponencial de 9 al propio *MACD* 2.5. Finalmente, el histograma 2.7 consiste en la resta entre los valores obtenidos para *MACD* y *Signal*.

$$\text{MACD} = \text{EMA}(12) - \text{EMA}(26), \quad (2.5)$$

$$\text{Signal} = \text{EMA}(9) \text{ de MACD}, \quad (2.6)$$

$$\text{Histograma} = \text{MACD} - \text{Signal}. \quad (2.7)$$

Con este indicador podemos ejecutar diferentes operaciones dependiendo del componente con el que estemos trabajando. Por lo tanto, podemos diferenciar tres tipos de estrategias que podemos utilizar:

- Si nos fijamos en el histograma, tendremos una señal de compra cuando este pasa de un valor negativo a un valor positivo. Por el contrario, tendremos una señal de venta cuando el histograma pasa de un valor positivo a un valor negativo.
- Si nos fijamos en *Signal*, tendremos una señal de compra cuando la línea pasa por encima del nivel cero. Por el contrario, tendremos una señal de venta cuando la línea pasa por debajo del nivel cero.
- Si nos centramos en el cruce entre *Signal* y el histograma, podemos deducir que tendremos una señal de compra cuando la línea *Signal* sale del histograma por debajo del

nivel cero. Por el contrario, si la línea Signal sale del histograma por encima del nivel cero, tendremos una señal de venta.

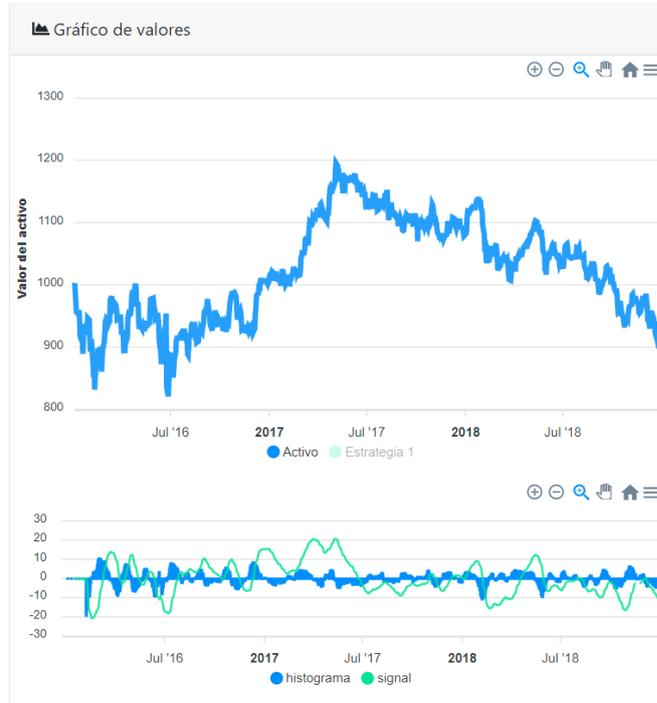


Figura 2.3: Indicador MACD.

En la figura 2.3 podemos observar en la parte superior el activo y en la parte inferior distinguimos: una línea verde que actúa como Signal y un histograma en azul que actúa como el Histograma.

Estocástico

Es uno de los indicadores más utilizados dentro del análisis técnico. Es bastante parecido al *RSI* ya que este indicador nos marca también zonas de sobrecompra y sobreventa [18]. Es un indicador que tiene en cuenta la distancia entre el precio de cierre, el mínimo y el máximo de la sesión. Al igual que *RSI*, está normalizado entre 0 y 100 y se establecen las zonas de sobrecompra en 80 y de sobreventa en 20. En este caso, tendremos otro parámetro llamado %D 2.9 que consiste en una media móvil de n periodos sobre el propio oscilador estocástico 2.8. Ambos se obtienen de la siguiente forma:

$$\%K = 100 \times \frac{C - Ln}{Hn - Ln}, \tag{2.8}$$

$$\%D = SMA(n) \text{ de } \%K, \tag{2.9}$$

donde L_n y H_n son los precios más bajo y más alto, respectivamente, durante las últimas n sesiones y C es el precio de cierre actual.

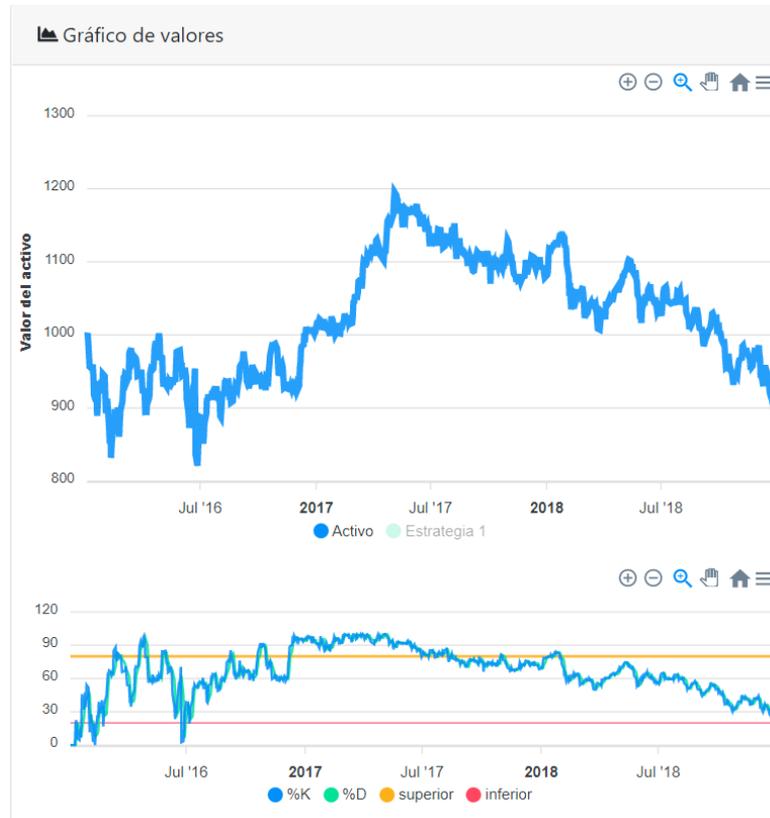


Figura 2.4: Indicador Estocástico.

En la figura 2.4 podemos observar en la parte superior el activo y en la parte inferior distinguimos 4 líneas, una superior en amarilla que nos marca el límite superior, una inferior en rojo que nos marca el límite inferior, una en azul que nos indica la línea %K y una línea en verde que nos indica %D.

El Oscilador Estocástico nos generará señales de venta cuando entre en la zona de sobrecompra, es decir, por encima de 80, y nos dará señales de compra cuando baje de la zona de 20 y entre en sobreventa.

Si el Estocástico %K corta hacia abajo su media, %D nos estará mostrando una señal de venta. Por el contrario, si tenemos que %K corta de forma ascendente a su media, %D nos está ofreciendo una señal de compra.

Bandas de Bollinger

Es un indicador muy utilizado dentro del análisis técnico [19]. Está formado por tres líneas: una superior 2.10, una inferior 2.12 y una intermedia 2.13. Estas líneas son tres medias móviles como las vistas anteriormente, pero con diferentes parámetros cada una:

- La banda superior corresponde a la suma de la media móvil simple de n -periodos y dos desviaciones estándar de las cotizaciones de los últimos n -periodos, es decir, se calcula de la siguiente forma:

$$\text{bandaSuperior} = \frac{\sum_{i=1}^n \text{Precio de cierre}}{n} + 2 \times \sigma, \quad (2.10)$$

donde

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (\text{Promedio}(n) - \text{Precio de cierre}(i))^2}{n}}. \quad (2.11)$$

- La banda inferior corresponde a la diferencia entre las media móvil simple de n -periodos y dos desviaciones estándar de las cotizaciones de los últimos n -periodos, es decir,

$$\text{bandaInferior} = \frac{\sum_{i=1}^n \text{Precio de cierre}(i)}{n} - 2 \times \sigma. \quad (2.12)$$

- La banda intermedia corresponde a la media móvil de n -periodos, y se calcula como:

$$\text{bandaMedia} = \frac{\sum_{i=1}^n \text{Precio de cierre}(i)}{n}. \quad (2.13)$$

Para poder interpretar las Bandas de Bollinger, lo único que tenemos que hacer es observar la distancia entre las bandas. A menor distancia, menor volatilidad; mientras que a mayor distancia, mayor volatilidad.

Para analizar este indicador dentro de la gráfica de valores, debemos tener en cuenta los siguientes puntos:

- Cuando el precio del activo se encuentra por encima de la banda superior, estaríamos hablando de un activo al alza, dando a entender que la tendencia actual continuará ya que hay fuerza alcista en el mercado, obteniendo así una señal de compra.
- Cuando el precio del activo se encuentra por debajo de la banda inferior, estaríamos hablando de un activo a la baja, dando a entender que la tendencia actual continuará ya que hay fuerza bajista en el mercado, obteniendo así una señal de venta.
- La ausencia de volatilidad, la cual vemos con las bandas estrechas, significaría que se estaría preparando para una tendencia alcista.

- Los máximos y mínimos que vemos tanto fuera de las bandas como dentro de ellas nos indican simplemente un cambio de tendencia.



Figura 2.5: Bandas de Bollinger.

En la figura 2.5 podemos observar en azul la línea del activo, en amarillo la banda inferior del indicador, en rojo la banda media y en morado la banda superior.

CCI

El CCI [20] es un indicador de tendencias creado por Donald Lambert en 1980. Su objetivo principal es detectar nuevas tendencias y alertar sobre cambios de volatilidad. La fórmula para calcular este indicador es la siguiente:

$$CCI = \frac{\text{Precio típico} - \text{Media Móvil}}{0.015 \times \sigma}, \quad (2.14)$$

donde el denominado Precio típico en la ecuación anterior se calcula como:

$$\text{Precio típico} = \frac{\text{Máximo} + \text{Mínimo} + \text{Cierre}}{3}, \quad (2.15)$$

y los parámetros en la misma corresponden a:

- σ : ver Eq. 2.11
- Media Móvil: por defecto de 20 sesiones.

- Constante: tiene el valor 0.015 para asegurar que los valores caen principalmente entre -100 y 100 .



Figura 2.6: Indicador CCI.

En la figura 2.6 podemos observar en la parte superior el activo que vamos a analizar y en la parte inferior una línea azul que identifica al indicador CCI. La interpretación de este indicador es muy sencilla ya que, si el indicador supera el límite de $+100$ entramos en una zona de fuerte tendencia alcista. Si el indicador cae por debajo de -100 , entramos, por el contrario, en una zona de fuerte tendencia bajista. La estrategia básica que utilizamos con este indicador es generar señales de compra en el momento en el que supera los límites establecidos. Por lo tanto:

- Tendremos una señal de compra cuando el indicador está por debajo del límite -100 de sobreventa.
- Tendremos una señal de venta cuando el indicador está por encima del límite $+100$ de sobrecompra.

Variación de porcentaje

Este indicador está formado por dos líneas:

- Variación de porcentaje del último **máximo** 2.16: esta línea nos mostrará la variación del porcentaje en relación al último máximo. Se calcula como:

$$VPM = VrM \times \left(1 - \frac{\%M}{100}\right), \quad (2.16)$$

donde VrM es el valor de referencia que tomamos del último máximo y %M es un porcentaje elegido por el usuario.

- Variación de porcentaje del último **mínimo** 2.17: esta línea nos mostrará la variación del porcentaje en relación al último mínimo. Se obtiene de la siguiente forma:

$$VPm = Vrm \times \left(1 + \frac{\%m}{100}\right), \quad (2.17)$$

donde Vrm es el valor de referencia que tomamos del último mínimo y %m es un porcentaje elegido por el usuario.



Figura 2.7: Variación de porcentaje.

En la figura 2.7 podemos observar 3 líneas: una línea azul que nos marca el activo que vamos a analizar, una línea amarilla que identifica a VPM y una línea en rojo que identifica a VPm.

Podemos interpretar este indicador de manera sencilla, ya que llega con centrarnos en el momento en el que el activo coincide con el indicador. Ahí veremos que el precio sube o baja el porcentaje indicado %M o %m, por lo que podremos realizar diferentes operaciones.

Con este indicador podemos establecer señales de compra en el instante en el que el precio coincide con la línea de VPM. Por el contrario, definimos señales de venta en el instante en el que el precio coincide con la línea VPM.

Movimiento por fecha

Se ha incluido el **Movimiento por fecha** dentro del apartado de indicadores aunque no obtenemos con su cálculo ninguna línea que podemos analizar en el gráfico. Simplemente, se ejecutan acciones de compra y venta cada cierto intervalo de tiempo, que viene dado en días.

Ichimoku

El indicador **Ichimoku** [21], también conocido como Ichimoku Kindo Hyo, es un conjunto de medias móviles que actúan con cierta complejidad. Su función principal es identificar la tendencia dominante en un activo financiero. Está formado por 5 líneas diferentes, definidas por las siguientes expresiones:

$$\text{Chikou Span} = \text{Precio actual retrasado 26 periodos}, \quad (2.18)$$

$$\text{Tenkan-Sen} = \frac{\text{Hn}(9) + \text{Ln}(9)}{2}, \quad (2.19)$$

$$\text{Kijun-Sen} = \frac{\text{Hn}(26) + \text{Ln}(26)}{2}, \quad (2.20)$$

$$\text{Senkou-Span-A} = \frac{\text{Tenkan-Sen}(26) + \text{Kijun-Sen}(26)}{2}, \quad (2.21)$$

$$\text{Senkou-Span-B}(26) = \frac{\text{Hn}(52) + \text{Ln}(52)}{2}, \quad (2.22)$$

donde Hn es el máximo más alto dentro del periodo y Ln es el mínimo más bajo en dicho periodo y entre paréntesis detallamos los periodos para la realización de cada línea.

La línea Chikou Span nos sirve para comparar cotizaciones pasadas. La línea Tenkan-Sen o línea de conversión suele ser la señal de entrada. La línea Kijun-Sen o línea base actúa como soporte o resistencia de corto plazo. Por último, tenemos la nube, formada por las líneas Senkou Span A o Extensión principal A y Senkou Span B o Extensión principal B.

Para operar con este indicador, podemos utilizar diferentes estrategias:

- Estrategia con la nube: se define como nube el área comprendida entre las líneas Senkou Span A y Senkou Span B. Cuando el precio está por encima de la nube, tendremos una tendencia alcista y, por lo tanto, obtendremos una señal de compra. Por el contrario, cuando el precio está por debajo de la nube, tendremos una señal de venta, ya que la tendencia es bajista.
- Estrategia de cruces entre Tenkan-Sen y Kijun Sen: tendremos una fuerte señal de compra en el momento en el que el cruce sea alcista, por encima de la nube, y la señal Chikou Span está también por encima de la nube. Por el contrario, si el cruce se encuentra por debajo de la nube y la señal Chikou Span también, tendremos una señal fuerte de venta.



Figura 2.8: Indicador Ichimoku.

En la figura 2.8 podemos observar una línea azul que identifica al activo que vamos a analizar y después una línea en cada color para cada una de las líneas que presenta este indicador.

OBV

El indicador **OBV** [22] es un indicador desarrollado por Joseph Granville [23] en la década del 1960 y que se popularizó al describirlo en su libro [24]. El objetivo principal de este indicador es relacionar la evolución de un activo con el volumen dentro del periodo temporal que se está analizando. Este indicador se crea sumando el volumen positivo o negativo dependiendo de si los precios cierran más altos o más bajos en relación al día anterior. La fórmula para su cálculo es la siguiente:

Si el precio cierra más alto que el día anterior:

$$OBV = OBV(i-1) + VOLUMEN, \quad (2.23)$$

mientras que si el precio cierra más bajo que el día anterior, tenemos:

$$OBV = OBV(i-1) - VOLUMEN. \quad (2.24)$$

Puede ocurrir también que el precio cierra igual que el día anterior y, por lo tanto:

$$OBV = OBV(i-1). \quad (2.25)$$



Figura 2.9: Indicador OBV.

En la figura 2.9 podemos observar en la parte superior el activo que vamos a analizar y en la parte inferior la línea correspondiente al indicador OBV. En este caso, el eje Y está normalizado en millones (M). En este proyecto, vamos a dejar este indicador como informativo, para que nos sirva como apoyo para visualizar las tendencias futuras y, junto con otros indicadores, aplicar las correspondientes operaciones de compra y venta.

Fundamentos tecnológicos

En este capítulo se van a explicar las herramientas utilizadas para el desarrollo del proyecto.

3.1 Control de versiones

Como sistema de control de versiones hemos utilizado **Git** [25] ya que es gratuito y de código abierto. Está diseñado para manejar desde proyectos pequeños hasta muy grandes, con rapidez y eficiencia. Como plataforma para guardar el proyecto se ha utilizado **Bitbucket** [26], ya que nos permite almacenar el proyecto y poder editar y controlar los archivos intuitivamente. Como casi todas las plataformas, nos permite trabajar desde diferentes equipos y guardar una copia de seguridad de todas las versiones. Con la cuenta gratuita se pueden alojar proyectos de forma pública y de forma privada, concediendo permisos de acceso.

3.2 Diseño

Para realizar los diagramas que se incluyen en este documento, se ha utilizado **Draw.io** [27], una aplicación web que permite crear cualquier tipo de diagrama. Esta aplicación da la opción de realizar varios diagramas diferentes y proporciona plantillas que pueden servir como referencia. La principal ventaja de esta aplicación es que no se necesita instalar nada, a diferencia de **Dia**, **MagicDraw** o de otras aplicaciones que requieren de un instalador. Otra ventaja interesante es que tiene conexión directa con la nube, por ejemplo con OneDrive, y permite tener los archivos siempre disponibles.

3.3 Entorno de desarrollo

Se ha utilizado el entorno de desarrollo de **Node-RED** [28] basado en **Node.js** [29]. Puede ejecutarse localmente, en dispositivos hardware o en la nube. Consiste en un editor de flujo en el navegador que facilita la ejecución de flujos mediante un amplio rango de nodos de la paleta.

Con las dependencias necesarias instaladas correctamente, se abre en el navegador la aplicación web con la que se puede realizar el flujo principal del proyecto. Este entorno ofrece nodos ya configurados que resultan útiles a la hora de realizar nuestra aplicación. Los flujos se pueden implementar en tiempo de ejecución con un solo clic. Se pueden definir las siguientes características:

- El editor principal de Node-RED se utiliza para poner los nodos que se conectarán para crear el flujo.
- En la parte izquierda se encuentra la paleta de nodos predefinidos por Node-RED, que permiten realizar múltiples funcionalidades, como funciones en Javascript, unir mensajes, cambiar propiedades...
- La conexión de los nodos se realiza deslizando el punto de salida de un nodo con el punto de entrada del nodo siguiente.
- Con esta conexión entre nodos, se transmite un mensaje entre ellos desde el nodo de salida hasta el nodo de entrada. Estos mensajes tienen diferentes propiedades, pero se puede distinguir el contenido del mensaje y el emisor de este. Para la información que se guarda en estos mensajes se puede utilizar cualquier estructura de datos de Javascript, como pueden ser arrays o objetos.

El motivo principal de realizar este proyecto con Node-RED es que permite programar individualmente cada nodo y unirlos para formar el flujo principal de la aplicación. En definitiva, tener la capacidad de programar un flujo sin tener conocimientos de programación, simplemente uniendo nodos desde una interfaz intuitiva y sencilla.

3.4 Lenguajes de desarrollo

En esta sección se definen los diferentes lenguajes utilizados en este proyecto.

- El principal lenguaje utilizado es **Javascript** [30], que se usa principalmente para crear páginas web dinámicas. En nuestro caso, se utiliza para realizar funciones en algunos nodos del flujo y también para realizar la interfaz web.

- Para visualizar la aplicación web se ha utilizado **HTML** [31], un lenguaje que usa la estructura para dar un sentido al contenido web. Por ejemplo, define párrafos, cabeceras, tablas, gráficas...
- Se ha utilizado **Vue.js** [32], que es un framework que sirve para construir interfaces de usuario. En nuestro caso, se ha utilizado para integrar gráficas, ya que permite utilizarlo conjuntamente con Javascript de manera sencilla.
- Las gráficas se han realizado utilizando **Apexchart** [33], una biblioteca de gráficos moderna que permite generar gráficas de manera sencilla y realizar diferentes funcionalidades sobre ellas, como ampliar, reducir, descargar...
- También se ha utilizado **Bootstrap**[34], un framework sencillo utilizado para la creación de sitios web. En nuestro caso, se ha utilizado para dar estilo a la página principal ya que, apoyado por HTML, permite realizar interfaces de usuario sencillas.

Metodología

Se puede definir metodología de desarrollo software [35] como el proceso que hay que seguir a la hora de diseñar y realizar una solución de un proyecto en concreto. Cualquier metodología consta de una serie de procedimientos y técnicas que nos van a permitir crear software con cierto nivel de calidad. A pesar de permitir cierta flexibilidad en cuanto a la relación de sus objetivos en ciertos puntos del proyecto, su cumplimiento garantiza un producto final que cumple de manera notoria con los objetivos generales inicialmente establecidos.

Existen principalmente dos clases de metodologías: por una parte, tenemos las denominadas *metodologías ágiles*, y, por otro, las denominadas *metodologías clásicas* [36]. En este trabajo nos centraremos en las primeras, ya que permiten una mayor flexibilidad, una mejora continua y una adaptación constante a las necesidades del producto. Con esta metodología se busca darle mayor protagonismo al cliente, ya que su participación en el proyecto aumenta.

Una metodología ágil es una metodología de desarrollo software que mantiene como idea principal el desarrollo iterativo e incremental. Esta metodología nos proporciona respuestas rápidas y flexibles al cambio mediante la planificación adaptativa, la identificación de requisitos y el desarrollo gradual de soluciones. Gracias a la utilización de esta metodología también se obtienen otro tipo de beneficios, como el ahorro de tiempo y coste en la ejecución del proyecto. Al seguir un modelo incremental, se puede mejorar la satisfacción del cliente, la calidad del producto y, a su vez, alertar de forma rápida de errores y de problemas que vayan surgiendo. La base principal es la interacción directa con el cliente proporcionándole entregas y pruebas parciales del trabajo hasta obtener la versión final.

Dentro de este tipo de metodologías se ha optado por una metodología Scrum [37], por su gestión regular de las expectativas del cliente (a través de entregas quincenales o mensuales), la posibilidad de usar el producto antes de que estén implementadas todas sus funcionalidades

o el incremento de motivación que produce el trabajo en equipo planteado en dicha metodología. Como consecuencia, obtenemos una mejor calidad del producto final que satisface las necesidades del cliente.

4.1 Metodología Scrum

Esta metodología es una metodología ágil y flexible que se utiliza principalmente en proyectos en los que tenemos cambios constantes de requisitos que alteran la realización de los mismos. Debido a este nivel de incertidumbre tan elevado, esta metodología se centra en ajustar los resultados a las exigencias exactas del cliente. Como el cliente modifica sus exigencias a corto plazo, se tienen que ir definiendo entregables y revisarlos constantemente.

Esta metodología se basa en la realización de reuniones periódicas donde se regula el cumplimiento de los objetivos parciales establecidos. Los elementos básicos de Scrum son los denominados roles, los eventos y los artefactos. Explicaremos a continuación cada uno de ellos.

4.1.1 Roles

Dentro de Scrum, se establecen tres roles:

- *Product Owner*. Es el responsable del proyecto, que se encarga de maximizar el valor del trabajo del equipo de desarrollo. Debe ser una única persona y es la que se ocupa de interactuar constantemente con el cliente.
- *Scrum Master*. Es una única persona, responsable de que se entienda y se aplique Scrum dentro de la organización.
- *Development Team Member*. Es el miembro del equipo de desarrollo. Este equipo de desarrollo se encarga de realizar las tareas impuestas por el Product Owner. Los miembros del equipo de desarrollo son los encargados de crear el producto para que cumpla los objetivos. Se recomienda que esté formado por 10 personas, pero se puede adaptar dependiendo de la organización.

4.1.2 Eventos

Scrum define cinco tipos de eventos: el que se denomina Sprint y una serie de reuniones, 4 en concreto, asociadas a ese Sprint. El *Sprint* es el periodo, generalmente de quince días o un mes, que corresponde al desarrollo de un producto, potencialmente entregable. Puede considerarse un mini-proyecto con unos objetivos definidos al inicio de cada Sprint. Estos Sprints constituyen la base de Scrum. Aunque tienen una duración máxima de un mes, el

tiempo se decide dependiendo del cliente. Los cuatro tipos de reuniones que se llevan a cabo en un evento, es decir, durante el desarrollo del propio Sprint, son:

- *Sprint Planning*. Es la primera reunión de cualquier Sprint. En ella se planifica el Sprint, se define el tiempo de duración, el objetivo concreto y su correspondiente entregable. Tiene una duración de 8 horas para Sprints de un mes, aunque su duración se ajusta según la extensión del Sprint.
- *Daily Meeting*. Consiste en una reunión diaria del equipo de desarrollo que trabaja en el Sprint en la que se ponen en común las actividades y se sincronizan para elaborar el plan del día. Tiene una duración máxima de 15 minutos y en ella debe tenerse en cuenta lo realizado en el día anterior, lo que se va a realizar en el mismo día y si tengo algún impedimento para ello que demande una solución.
- *Sprint Review*. Esta reunión, en la que se realiza una revisión del Sprint y se estudia el proyecto en cuestión, se hace con el cliente o propietario del proyecto. Tiene una duración de 4 horas para Sprints de un mes y es la única reunión a la que asiste el cliente. Esta reunión sirve para mostrarle lo que se ha realizado y su funcionamiento y este comunica la validez del resultado y/o posibles mejoras adicionales.
- *Sprint Retrospective*. La retrospectiva del proyecto es la última fase de un Sprint en Scrum. Es una reunión que tiene una duración de 3 horas para Sprints de un mes y en la que se evalúa cómo se ha implementado Scrum al finalizar el Sprint. Su finalidad es la revisión del trabajo realizado y la propuesta de ideas para poder mejorar en Sprints subsiguientes.

4.1.3 Artefactos

Los artefactos o contenedores de información que contempla Scrum son los siguientes:

- *Product Backlog*. Es la lista de tareas de las que consta un proyecto, de acuerdo con las necesidades establecidas por el cliente. Cualquier cosa que haga el equipo de desarrollo debe estar incluido en el Product Backlog. El responsable de esta lista es el Product Owner. El equipo de desarrollo elige las tareas del Product Backlog a desarrollar en un Sprint determinado durante el Sprint Planning.
- *Sprint Backlog*. Es la lista de tareas del Product Backlog que son elegidas por el equipo de desarrollo en el Sprint Planning. El Sprint Backlog también incluye un plan para poder desarrollarlas, que se podría cambiar a lo largo del Sprint.
- *Incremento*. Es la parte de producto entregable en cada Sprint, ya terminado y probado.

4.2 Scrum aplicado al proyecto

La metodología explicada anteriormente se usa en la actualidad en organizaciones donde el número de personas es elevado. Para poder realizar el Trabajo de Fin de Grado con esta metodología, se ha tenido que adaptar en algunos puntos para conseguir coherencia, manteniendo siempre las bases de la metodología Scrum.

4.2.1 Roles

- *Scrum Master*. Este rol va a ser llevado a cabo por Paula María Castro Castro, codirectora de este Trabajo de Fin de Grado.
- *Product Owner*. Este rol va a ser llevado a cabo por José Pablo González Coma, codirector de este Trabajo de Fin de Grado.
- *Development Team*. Debido a que no disponemos del número de personas recomendado por Scrum (10), hemos tenido que adaptar este rol para que pueda ser realizado sólo por una persona, en este caso, el alumno Alejo Cedeira Vidal.

4.2.2 Eventos

Al inicio de cada uno de los Sprints se ha mantenido una reunión *Sprint Planning* que ha servido para establecer la duración del Sprint y definir las tareas y objetivos del mismo. Los objetivos de cada Sprint se revisan en la reunión *Sprint Review* realizada al terminar el Sprint correspondiente con el Product Owner, en este caso, José Pablo González Coma. Para evaluar como se ha implementado Scrum, tenemos una reunión *Sprint Retrospective* que realizamos junto al Scrum Owner, en este caso, Paula María Castro Castro. A todas estas reuniones asistirá el Development Team, en este caso, Alejo Cedeira Vidal.

Se definen en el proyecto los siguientes Sprints:

Sprint 0: Investigación

Este Sprint representa la fase inicial del proyecto. Su finalidad es la búsqueda de información sobre inversiones en Bolsa y sobre todos los conceptos relacionados cuyo manejo será necesario para la elaboración del proyecto. En esta fase también se realiza una búsqueda de las herramientas y tecnologías más adecuadas para el desarrollo de las funcionalidades necesarias para alcanzar los objetivos del proyecto. Además se realiza una planificación del tiempo necesario para la realización del trabajo y del coste asociado a su desarrollo.

Una vez terminado este Sprint, tendríamos el conocimiento necesario para abordar el proyecto, se habrán definido las herramientas y tecnologías que vamos a utilizar en nuestro pro-

yecto y se tendrá también una estimación inicial del coste total y del tiempo necesario para su realización.

Sprint 1: Desarrollo de indicadores básicos

Este Sprint es de los más importantes del proyecto, ya que consiste en la recopilación de información sobre los indicadores que se utilizarán en el trabajo, y su implementación.

Una vez terminado el Sprint, se habrán implementado los indicadores de estrategias de mercado requeridos por el Product Owner.

Sprint 2: Implementación de estrategias.

En este Sprint se realizan los casos de uso relacionados con la configuración de las estrategias. Además, se desarrolla la interfaz web de la aplicación.

A partir de la aplicación obtenida en el Sprint 1, se obtiene un entregable que implementa distintas estrategias con un mismo valor inicial de inversión.

Sprint 3: Pruebas y comparación de estrategias.

En este Sprint se realizan todos los indicadores que faltaban y se prueban sus funcionalidades, utilizando también configuraciones. Se prueban además todas las estrategias posibles, que se añaden a una tabla final que permite el análisis comparativo de las mismas.

Sprint 4: Realización de la memoria.

En este Sprint se realiza la memoria del proyecto. Como resultado de este Sprint, se obtiene el documento técnico completo y una versión final de la aplicación que incluye todas las funcionalidades, ya probadas, y que cumple con los objetivos iniciales establecidos por el Product Owner.

4.2.3 Artefactos

- *Product Backlog*. En este apartado se indican las funcionalidades de los productos intermedios que forman el proyecto.
 1. Conocimiento del problema y de los conceptos relacionados con el análisis bursátil.
 2. Identificación y definición de las herramientas y tecnologías a utilizar.
 3. Planificación inicial de tiempo y costes.
 4. Presentación de los indicadores del Análisis Técnico que se van a utilizar.

5. Implementación de un conjunto de indicadores.
 6. Realización de las primeras implementaciones de las estrategias.
 7. Desarrollo de la interfaz web.
 8. Realización de todos los indicadores.
 9. Prueba de todas las funcionalidades de los indicadores.
 10. Realización de comparativas de resultados.
 11. Redacción, incluyendo detalladamente todo el trabajo realizado, de la memoria.
- *Sprint Backlog*. Se definen las siguientes tareas del Product Backlog en cada uno de los Sprints, de la siguiente forma:

Sprint 0: Investigación

1. Conocimiento del problema y de los conceptos relacionados con el análisis bursátil.
En esta tarea se recopila toda la información posible acerca del tema a abordar, desde lo que es una acción hasta cómo interpretar unos gráficos de Bolsa apoyándonos en el Análisis Técnico.
2. Identificación y definición de las herramientas y tecnologías a utilizar.
En esta tarea se realiza una revisión de las posibles herramientas y tecnologías disponibles para la realización del proyecto, eligiendo aquellas que mejor se adaptan a las exigencias del mismo.
3. Planificación inicial de tiempo y costes.
En esta tarea se realiza una planificación temporal inicial de la realización del proyecto y de los costes asociados.

Sprint 1: Desarrollo de indicadores básicos

4. Presentación de los indicadores del Análisis Técnico que se van a utilizar.
En esta tarea se definen los indicadores principales que se van a utilizar en primer lugar en la aplicación para el análisis de las estrategias.
5. Implementación de un conjunto de indicadores.
Se implementan estos indicadores principales para poder ir realizando las primeras pruebas.

Sprint 2: Implementación de estrategias.

6. Realización de las primeras implementaciones de las estrategias.

En este Sprint se han integrado los indicadores ya implementados para poder elaborar una estrategia y obtener sus resultados.

7. Desarrollo de la interfaz web.

En este Sprint también se implementa la visualización de los datos y las gráficas en nuestra interfaz con HTML y VueJS.

Sprint 3: Pruebas y comparación de estrategias.

8. Realización de todos los indicadores.

En este Sprint se han implementado los indicadores técnicos más avanzados.

9. Prueba de todas las funcionalidades de los indicadores.

En este Sprint se han probado las diferentes funcionalidades que podemos presentar en una estrategia, así como todo tipo de configuraciones para los indicadores.

10. Realización de comparativas de resultados.

En este Sprint también se ha añadido una tabla resumen donde podemos comparar diferentes estrategias apoyándonos en la gráfica.

Sprint 4: Realización de la memoria.

11. Redacción, incluyendo detalladamente todo el trabajo realizado, de la memoria.

- *Incremento.* En el proyecto se definen cuatro incrementos, uno para cada uno de los Sprints, exceptuando el inicial:

HITO 1: Primera versión de la aplicación.

Se obtiene una aplicación que tiene implementados los indicadores principales que podremos representar y comparar entre sí en una gráfica.

HITO 2: Elaboración de estrategias.

Partiendo de la aplicación obtenida en el anterior Sprint, se pueden utilizar todos los indicadores técnicos que se van a utilizar en este proyecto y se permite realizar diferentes estrategias con el mismo valor de inversión total.

HITO 3: Pruebas.

Partiendo de la aplicación obtenida en el Sprint anterior, se han probado todas las posibilidades de estrategias que se pueden crear y se ha añadido la tabla resumen final para poder compararlas. También se han implementado los indicadores técnicos más avanzados.

HITO 4: Documentación de la memoria.

Se finaliza el documento por completo y se obtiene una versión final de la aplicación con todas las funcionalidades probadas y con todos los objetivos iniciales cumplidos.

Planificación y costes

En este capítulo vamos a presentar de manera detallada la planificación y los costes iniciales que se han estimado para la realización de este proyecto, así como el seguimiento una vez finalizado el mismo.

5.1 Planificación del proyecto

La planificación es el proceso de organizar los objetivos requeridos en un tiempo y espacio. Es importante tener una planificación inicial y hacer un seguimiento, ya que facilita la identificación de las desviaciones que ocurrieron durante el proyecto.

5.1.1 Planificación inicial

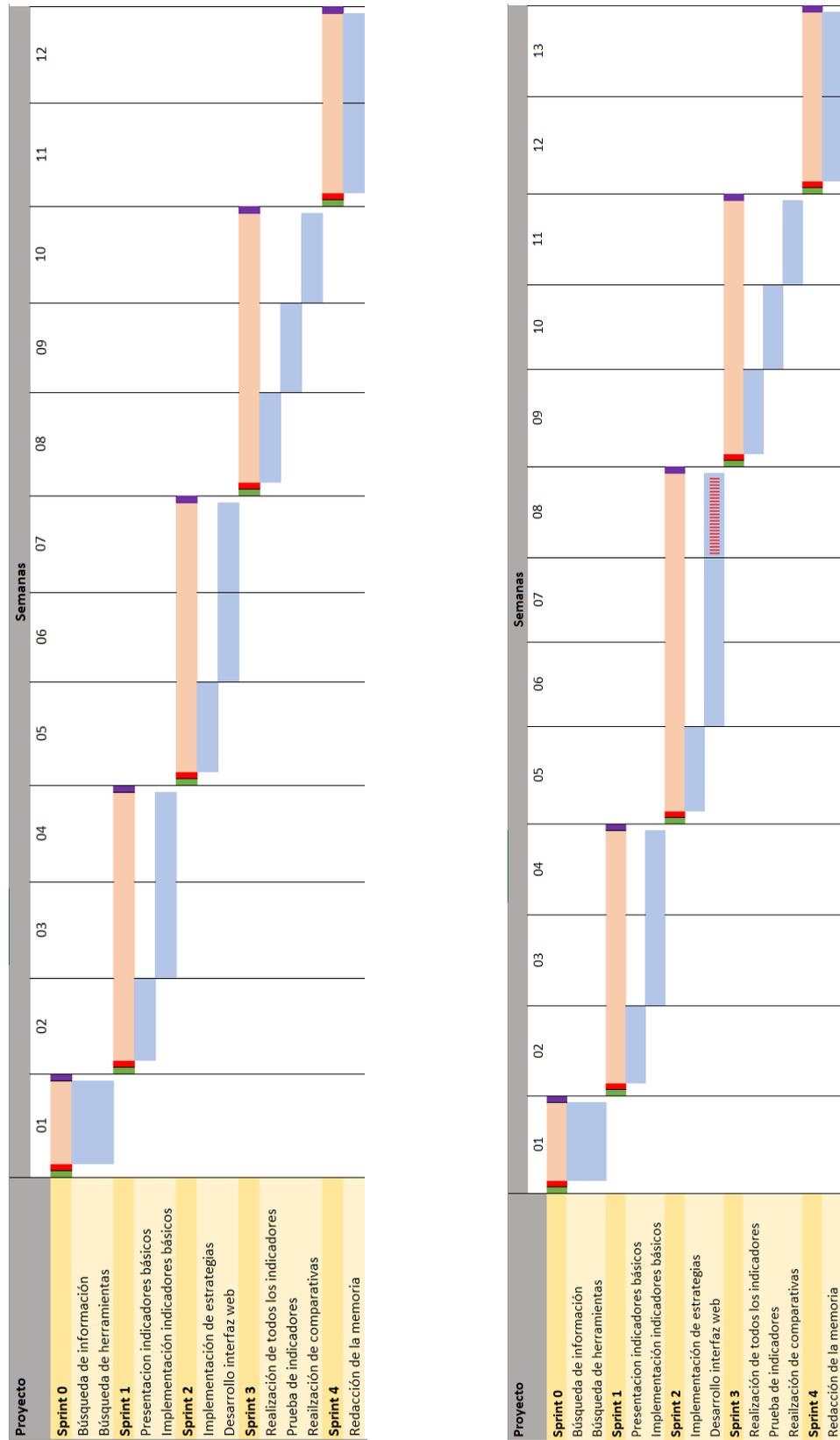
Para la planificación, se han utilizado, de forma aproximada, periodos temporales de una semana, asumiendo que un Sprint comienza inmediatamente después de que acabe otro, tal y como recoge la metodología Scrum que se plantea en el proyecto.

Mediante el diagrama de Gantt representado en la figura 5.1a se puede ver la duración total del proyecto, la duración de cada Sprint y la duración de cada una de las tareas que constituyen los Sprints estimada inicialmente.

En la figuras 5.1 se identifica una línea verde al principio de cada Sprint que hace referencia a la *Sprint Review*, una línea roja que representa la *Sprint Planning* y una línea morada al final de cada Sprint que representa la *Sprint Retrospective*.

5.1.2 Seguimiento

En la figura 5.1b se puede ver el seguimiento una vez finalizado el proyecto. Las desviaciones entre el diagrama de la planificación inicial y el seguimiento ocurren en la semana 8, donde hubo un retraso de una semana debido a la implementación de la interfaz web, lo que supuso un atraso de todo el proyecto de una semana más.



(a) Planificación inicial.

(b) Seguimiento.

37
Figura 5.1: Diagramas de Gantt

5.2 Recursos utilizados

Un recurso es una fuente de la cual se produce beneficio. Por lo tanto, en este proyecto los recursos tienen una gran importancia ya que son la base para el desarrollo de la aplicación. Durante el desarrollo del trabajo se han utilizado los siguientes recursos hardware y software.

Recursos hardware

- Ordenador portátil
 - Modelo: Xiaomi Mi Notebook Pro 15.6”.
 - CPU: Intel Core i5-8250U CPU 1.60GHz.
 - Disco Duro: 250GB SSD.
 - Memoria RAM: 8GB.
 - Sistema Operativo: Windows 10 Home x64bits.
 - Tarjeta gráfica: NVIDIA GeForce MX150.

Recursos software

- Node-RED.
- Bitbucket.

5.3 Estimación de costes

Este apartado incluye una estimación del coste total del proyecto. Para ello, dividiremos este coste total en tres partes: hardware, software y humano.

- *Estimación de coste de los recursos hardware.* Para el coste relacionado con el hardware, como sólo se ha utilizado el ordenador portátil, sólo se va a tener en cuenta en valor estimado de este en relación a los meses que ha llevado la realización del proyecto (3 meses). Por lo tanto, se parte de una estimación de 900 euros que se divide entre los años de vida media de un portátil, que es de 4. Una vez realizado esto, es necesario dividir el valor anual entre 4, ya que nos interesa el valor trimestral. El cálculo final sería $900/4 \times 1/4 = 56,25 \text{ €}$.
- *Estimación de coste de los recursos software.* Para el coste relacionado con el software, se analizan a continuación individualmente cada una de las herramientas utilizadas, y mencionadas en el apartado anterior:
 - *Node-RED.* Es una herramienta gratuita, por lo tanto su coste es nulo.

- *Bitbucket*. Se ha utilizado la versión gratuita de esta herramienta, por lo tanto su coste es nulo.
- *Estimación de coste de los recursos humanos*. En este apartado se van a diferenciar tres categorías: la de Development Team, desempeñada por el alumno Alejo Cedeira Vidal, la de Scrum Master, que lo realiza la profesora Paula María Castro Castro y el Product Owner, realizado por el profesor José Pablo González Coma.

Las jornadas de trabajo han sido aproximadamente de 6 horas al día, de lunes a viernes. En este tiempo de trabajo se han incluido las reuniones de equipo con una duración aproximada de una hora. Se han estimado 360 horas de trabajo para el programador, como vemos en la tabla 5.1 y 24 horas para los directores de proyecto.

Sprint	Objetivo	Fecha Inicio	Fecha Fin	Duración (horas)
0	Investigación	16/03/2020	23/03/2020	30
1	Desarrollo de indicadores	24/03/2020	14/04/2020	90
2	Implementación de estrategias	15/04/2020	06/05/2020	90
3	Pruebas y comparación de estrategias	07/05/2020	28/05/2020	90
4	Realización de la memoria	29/05/2020	08/06/2020	60

Tabla 5.1: Resumen de la duración del proyecto en horas.

Partiendo de que el sueldo medio de un miembro de un Development Team en la actualidad es de 23 €/h, de un Scrum Master es de 30 €/h y el de un Product Owner de aproximadamente 28 €/h, se puede ver la estimación de costes totales en la tabla 5.2.

Puesto	Salario	Horas	Total
Development Team	23 €/h	360	8280 €
Scrum Master	30 €/h	24	720 €
Product Owner	28 €/h	24	672 €
			9672 €

Tabla 5.2: Estimación del coste de los recursos humanos.

Por último, en la tabla 5.3 podemos ver el coste total de la realización del proyecto.

Recursos	Total
Hardware	56,25 €
Software	0 €
Humanos	9672 €
	9728,25 €

Tabla 5.3: Coste total del proyecto.

5.3.1 Seguimiento de los costes

Una vez finalizado el proyecto, los costes han aumentado debido al retraso que ha ocurrido en el desarrollo de la interfaz web. Este desajuste de la planificación implica sumar más horas de trabajo para todos los puestos, por lo tanto:

Sprint	Objetivo	Fecha Inicio	Fecha Fin	Duración (horas)
0	Investigación	16/03/2020	23/03/2020	30
1	Desarrollo de indicadores	24/03/2020	14/04/2020	90
2	Implementación de estrategias	15/04/2020	13/05/2020	120
3	Pruebas y comparación de estrategias	14/05/2020	04/06/2020	90
4	Realización de la memoria	05/06/2020	12/06/2020	60

Tabla 5.4: Seguimiento de costes.

Puesto	Salario	Horas	Total
Development Team	23 €/h	390	8970 €
Srum Master	30 €/h	30	900 €
Product Owner	28 €/h	30	840 €
			10710 €

Tabla 5.5: Seguimiento de costes humanos.

Recursos	Total
Hardware	56,25 €
Software	0 €
Humanos	10710 €
	10766,25 €

Tabla 5.6: Seguimiento del coste total del proyecto.

Análisis de requisitos

En este capítulo vamos a definir todos los objetivos que se pretenden conseguir en la aplicación, analizando para ello los casos de uso y las funcionalidades que se van a poder realizar.

6.1 Requisitos funcionales

En este apartado se van a identificar unos requisitos funcionales a partir de los objetivos marcados al principio del proyecto. Se ha dividido esta sección en dos partes: una, con los casos de uso que son necesarios para la generación del flujo, y otra, con los casos de uso que se pueden ejecutar en la pantalla principal de visualización de los datos.

6.1.1 Generación del flujo principal

En este apartado vamos a presentar los casos de uso necesarios para la generación del flujo principal de la aplicación. Estos casos de uso son los siguientes:

- CU-01 Seleccionar índice.
- CU-02 Definir estrategia.
- CU-03 Seleccionar indicadores.
- CU-04 Configurar indicador.
- CU-05 Realizar estrategia.
- CU-06 Inyectar datos.

6.1.2 Visualización de los datos.

En este apartado se van a presentar los casos de uso que se pueden lanzar a la hora de visualizar los datos. Son los siguientes:

- CU-07 Avanzar día a día.
- CU-08 Avanzar al último día.
- CU-09 Mostrar estrategias individualmente.
- CU-10 Mostrar resultados de las estrategias.
- CU-11 Reiniciar evaluación.

6.2 Requisitos no funcionales

Es necesario seguir los pasos marcados para la configuración de las estrategias, ya que, en caso contrario, no se mostrará nada en la pantalla principal. Si alguna configuración de estrategia falla, esta no se muestra en ella, pero las demás realizadas correctamente sí. Realizar el flujo principal es muy sencillo, pero para hacerlo correctamente hay que seguir unas pautas establecidas. Como la información va desde un nodo hacia otro por medio de su conexión, esto nos permite una gran integridad asegurando que la información va a transcurrir correctamente por el flujo si las conexiones son las correctas. La fiabilidad de la aplicación también recae sobre la unión correcta de los nodo. Siempre que estas estén bien, la aplicación funcionará sin errores.

6.3 Actores

En esta aplicación no consideramos gestión de usuarios y, por lo tanto, sólo definimos un único actor, que es el que puede configurar una estrategia y visualizar gráficamente los resultados.

6.4 Historias de usuario

En este apartado se detallan las historias de usuario que se han obtenido a partir de los requisitos funcionales definidos previamente.

En primer lugar, se ha realizado la tabla 6.1, donde podemos ver las distintas prioridades que se le han asignado a todas las tareas. Conviene destacar que una prioridad menor implica una mayor importancia en el proyecto .

Nombre	Prioridad
CU-01 Seleccionar índice	1
CU-02 Definir estrategia	1
CU-03 Seleccionar indicadores	1
CU-04 Configurar indicador	1
CU-05 Realizar estrategia	1
CU-06 Inyectar datos	1
CU-07 Avanzar día a día	2
CU-08 Avanzar al último día	2
CU-09 Mostrar estrategias individualmente	3
CU-10 Mostrar resultados de las estrategias	3
CU-11 Reiniciar evaluación	3

Tabla 6.1: Prioridades de los casos de uso.

6.4.1 Flujo

En este apartado se detallan los casos de uso necesarios para generar el flujo principal.

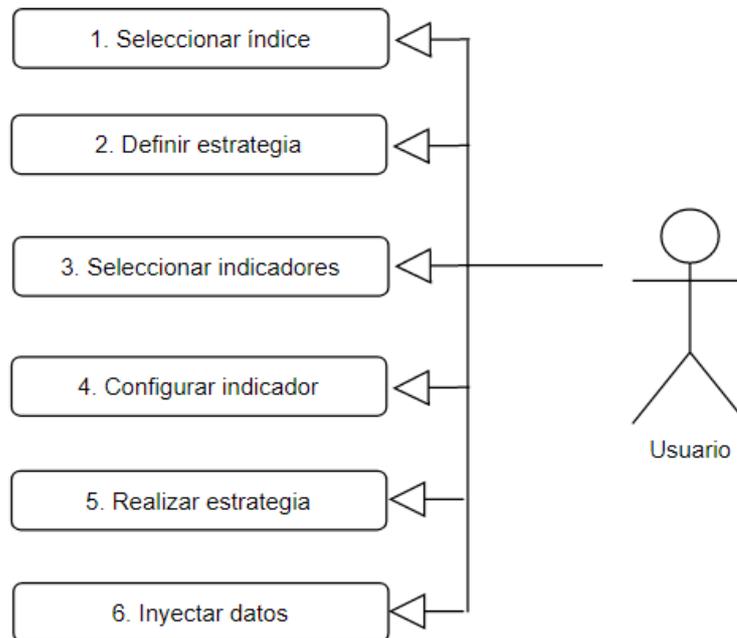


Figura 6.1: Historias de usuario en el flujo,

CU-01 Seleccionar índice. Para poder construir una estrategia, lo primero que hay que hacer es elegir qué índice vamos a utilizar. Una vez seleccionado, hay que unir este índice con el nodo *Definir estrategia*.



Figura 6.2: Seleccionar índice.

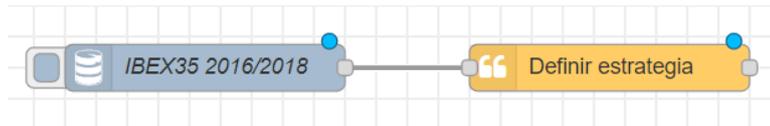


Figura 6.3: Conexión.

CU-02 Definir estrategia. Una vez que tenemos el índice seleccionado, se debe establecer la liquidez total que se va a tener en la inversión. Para ello hay que seleccionar el nodo *Definir estrategia* y poner en la variable "Inversión total" una cantidad igual a la de la figura 6.4.

Inversión total

Figura 6.4: Inversión total.

CU-03 Seleccionar indicadores. Después de haber definido los parámetros iniciales de la estrategia, lo siguiente es seleccionar los indicadores que vamos a utilizar. Para ello es necesario unir la salida del nodo *Definir estrategia* con los nodos de los indicadores que se van a usar, como puede verse en la figura 6.5. Cada indicador se define en un nodo diferente y permite ajustar sus parámetros para configurarlo de diferentes maneras.

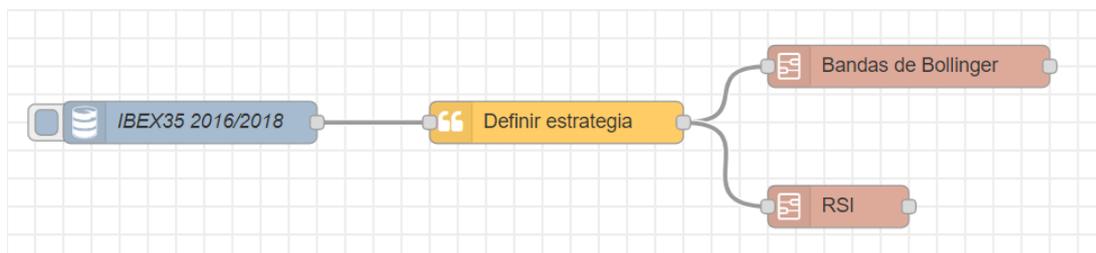


Figura 6.5: Seleccionar indicador/es.

Name	<input type="text" value="Name"/>
Periodos	<input type="text" value="20"/>
Señal venta	<input type="text" value="si"/>
Vender el	<input type="text" value="35"/>
Señal compra	<input type="text" value="si"/>
Cantidad nueva	<input type="text" value="100"/>

Figura 6.6: Parámetros indicador/es.

CU-04 Configurar indicador. Como se ve en la figura 6.5, cada indicador está definido en un nodo y, por lo tanto, cada indicador tiene diferentes parámetros (ver figura 6.6) que podemos configurar o dejar por defecto. Estos parámetros pueden ser diferentes para cada indicador, aunque los más comunes son: periodos, señales de compra, señales de venta, cantidad de compra... Por último, se debe unir el indicador elegido con el nodo *Realizar estrategia*, como puede observarse en la figura 6.7.

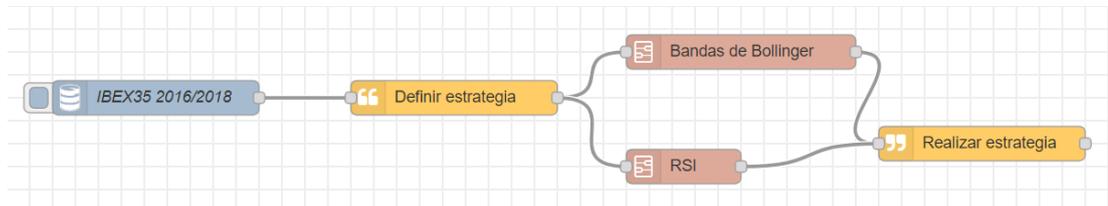


Figura 6.7: Unión con Realizar estrategia.

CU-05 Realizar estrategia. En este caso de uso se realiza la estrategia, teniendo en cuenta los puntos de compra y de venta que proporciona cada indicador. Es necesario que el usuario configure qué inversión inicial quiere para empezar la estrategia, como vemos en la figura 6.8. Por último, hay que unir cada *Realizar estrategia* con el nodo *Gráfica*, tal y como se representa en la figura 6.9.

Inversion inicial

Figura 6.8: Inversión inicial.

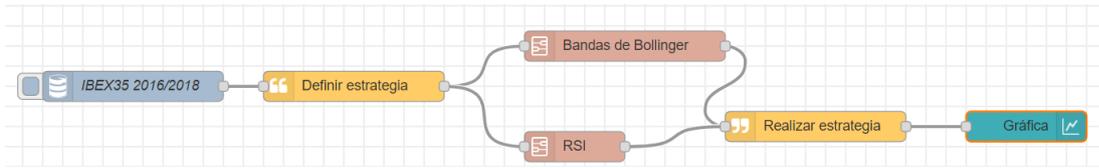


Figura 6.9: Flujo completo.

CU-06 Inyectar datos. Una vez realizadas las estrategias que se van a visualizar en el gráfico, lo único que se tiene que hacer es inyectar los datos pulsando en el botón del nodo que se muestra en la figura 6.10. Una vez realizado esto, ya tenemos los datos de nuestras estrategias disponibles para poder visualizarlos en la interfaz web.



Figura 6.10: Inyectar datos.

6.4.2 Interfaz

En este apartado se detallan los casos de uso necesarios para interactuar en la interfaz web. Estos casos de uso son:

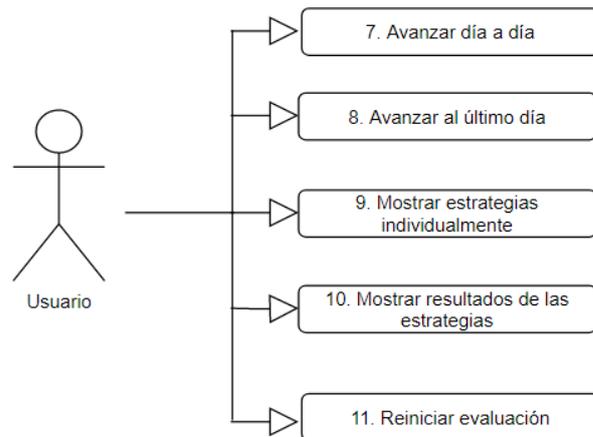


Figura 6.11: Historias de usuario en la interfaz.

CU-07 Avanzar día a día. Este caso de uso consiste en un botón *Avanzar* en la pantalla principal de la aplicación web que, al ser pulsado, permite ir visualizando día a día la evolución de cada indicador, el precio del activo y el resultado que se va obteniendo en cada estrategia. También se añadirá en la tabla de acciones, la acción correspondiente cuando el día actual presente una señal de compra o de venta.

CU-08 Avanzar al último día. Al pulsar el botón *Directo* se avanza hasta el último día de la estrategia, donde se verán los resultados finales de esta, la evolución completa de los indicadores utilizados y el comportamiento seguido por el activo. También se observan las acciones totales llevadas a cabo por cada indicador.

CU-09 Mostrar estrategias individualmente. Una vez que se haya decidido si avanzar diariamente o hasta el final de forma directa, al pulsar el botón *Mostrar* se puede visualizar únicamente una estrategia a la vez, lo que ayuda a ver mejor tanto la gráfica principal como la tablas correspondientes a las acciones ejecutadas.

CU-10 Mostrar resultados de las estrategias. Este caso de uso funciona igual que el **CU-09** ya que, una vez elegido, si se avanza diariamente o hasta el final de forma directa, se puede pulsar el botón *Estrategias* para visualizar únicamente los resultados de las estrategias configuradas.

CU-11 Reiniciar evaluación. En este caso de uso basta con pulsar el botón *Reiniciar* para reiniciar toda la evaluación desde el principio, pudiendo lanzar otra vez los casos de uso **CU-07** y **CU-08**.

Capítulo 7

Diseño

En este capítulo se van a explicar las decisiones relativas al diseño para el desarrollo de la aplicación.

En primer lugar, conviene destacar que es complicado pensar en un diseño de la aplicación cuando se usa Node-RED, ya que no sigue el estándar de diseño que pueden tener otro tipo de aplicaciones, ya sea en Java, Python o cualquier otro lenguaje. Node-red es un entorno de desarrollo que se basa en la unión de diferentes nodos para hacer un flujo que nos permita realizar las funcionalidades requeridas por la aplicación. La salida de cada nodo funciona como la entrada del nodo siguiente, el cual debe tratar el mensaje recibido y procesar la salida correspondiente. Estos mensajes tienen unas propiedades que nos permiten utilizar el contenido y el emisor de este, que nos sirve para saber de dónde procede el mensaje y así tratarlo de una manera específica. Por lo tanto, la implementación de los nodos se basa en recorrer la información que viene en el mensaje enviado por el nodo anterior y, a partir de ahí, realizar diferentes operaciones.

7.1 Estructura

Para poder entender el diseño de esta aplicación, previamente tenemos que definir una clasificación de las diferentes partes que lo constituyen.

7.1.1 Fuente de datos

La mayoría de aplicaciones tienen una fuente de datos de donde obtener datos para realizar operaciones. Esta puede presentarse de diferentes formas, como una base de datos o una lectura de fichero. Con Node-RED también podemos utilizar las funcionalidades anteriores, ya que están presentes en el entorno de desarrollo que nos proporciona pero, en este caso, se

ha decidido utilizar un nodo que permite inyectar datos. Este nodo se conoce como *inject* y nos permite meter datos a nuestro flujo en el instante en el que se accione el botón que está presente en el mismo nodo. Gracias a esto, se puede dejar que el usuario realice su estrategia completa uniendo los nodos para crear el flujo y, una vez que la tenga hecha, pueda inyectar los datos para poder visualizarlos en la interfaz del usuario.

7.1.2 Back-end

El Back-end es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione. Se trata del conjunto de acciones que pasan en una web pero que no vemos. En nuestro caso, es la unión de los nodos, que nos permite crear el flujo de la estrategia para poder visualizarla en la interfaz web.

De nuevo, al ser Node-RED un lenguaje con marcadas diferencias respecto a otras alternativas convencionales, exige un tratamiento particular en cuanto a la ubicación de las funcionalidades requeridas en la aplicación. En nuestro caso, dónde se van a calcular los indicadores y simular las estrategias de inversión. Node-RED nos proporciona la creación de nodos *subflow* que nos permiten encapsular diferentes nodos en uno sólo y definir variables globales que usaremos en su interior. Otra de las ventajas de utilizar los nodos *subflow* de Node-RED es que los nodos nos van a aparecer en la paleta de nodos lateral. Por lo tanto, el usuario podrá ir escogiendo los nodos que desee y arrastrarlos al editor central para crear el flujo.

En nuestro flujo nos podemos encontrar con tres tipos de nodos:

- Dentro de las posibilidades que nos ofrece Node-RED, la mejor opción consiste en crear un nodo para cada indicador. Estos nodos encapsulan la funcionalidad, y se encargan de realizar todas las operaciones relacionadas con ese indicador. Es decir, presentan diferentes opciones para ayudar al usuario a elegir de qué manera quiere implementar cada indicador. En la figura 7.1 vemos el nodo *subflow* definido para un indicador.

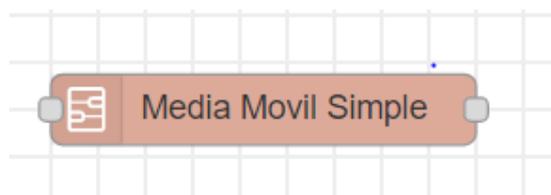


Figura 7.1: Nodo Indicador.

- Una vez determinado cómo se va a realizar cada indicador, hay que pensar en cómo se va a implementar la estrategia sobre el activo que queremos analizar, y las acciones disparadas en base a las señales proporcionadas por los indicadores que se han seleccionado. Para este fin, se ha definido el nodo llamado *Definir estrategia*, que se presenta en la figura 7.2. Este nodo debe ir colocado después del índice seleccionado y antes de los indicadores.

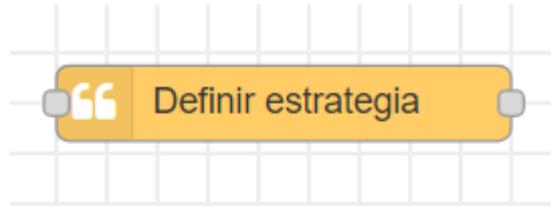


Figura 7.2: Nodo *Definir estrategia*.

- En la figura 7.3, se puede ver el nodo *Realizar estrategia*. Como su nombre indica, es el nodo encargado de realizar la estrategia, teniendo en cuenta las acciones de compra y venta que nos proporcionan las señales de los indicadores. Este nodo debe ir situado justo después de los indicadores y sirve como punto de unión de estos para realizar la estrategia. De este modo, podemos incorporar en cada estrategia las acciones de compra y venta de uno o más indicadores de manera sencilla, ya que toda la información necesaria se encapsula en un único nodo.

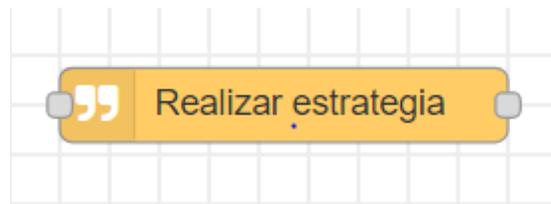


Figura 7.3: Nodo *Realizar estrategia*.

7.1.3 Front-end

El Front-end es la parte del desarrollo web que interactúa directamente con el usuario. En nuestro caso es la pantalla donde se visualizan las estrategias mediante gráficas y tablas.

Figura 7.4: Nodo *Gráfica*.

Para realizar la parte de desarrollo web Front-end, se ha utilizado el nodo UIBuilder de Node-RED que nos permite hacer la interfaz web apoyándonos en HTML, Bootstrap y Vue.js. Este apartado se encuentra en el nodo *Gráfica 7.4*, que es un nodo *subflow* de Node-RED donde está el nodo UIBuilder.

Implementación

En este capítulo se comentan los aspectos más importantes que se han seguido en la implementación de la aplicación.

8.1 Implementación

Para realizar la implementación de la aplicación, se ha intentado aprovechar todas las funcionalidades que nos facilita Node-RED, las cuales citamos en el capítulo 3. Por lo tanto, la base de nuestra aplicación va a ser crear un flujo principal realizado mediante la conexión de nodos y así interpretar los mensajes que se envían entre ellos para analizar la información y realizar las operaciones requeridas.

8.1.1 Índices y acciones

Para cada acción o índice bursátil se ha implementado un inyector diferente, al cual tenemos que pasarle los datos del activo en formato JSON. Estos nodos están siempre fijos en el editor principal de Node-RED y el usuario debe seleccionar uno. El funcionamiento de estos nodos es sencillo ya que sirven para meter los datos del activo seleccionado en el flujo.

Estos datos son extraídos de la página web de **Yahoo Finanzas** [38]. Esta página permite obtener datos históricos de prácticamente todos los índices y acciones del mercado de bolsa. Es una página web gratuita con todo tipo de información relativa a la inversión en bolsa. Una vez que se ha seleccionado el índice o acción, se puede elegir el rango de fechas deseado y descargar un archivo CSV con esos datos.

Para el nodo *inject* se necesita el contenido del fichero en formato JSON. Para eso, se convierte el fichero CSV a formato JSON. Esto se puede realizar de muchas maneras, pero se ha optado por utilizar un conversor online como, por ejemplo, **CSVJSON**[39].

Una vez convertido el fichero, se pega el contenido en el interior del nodo *inject*. En nuestra aplicación se dispone de unos cuantos ejemplos de índices y acciones que el usuario puede usar en sus pruebas pero, si le interesa introducir alguno en concreto, lo único que tiene que hacer es seguir los pasos anteriores.

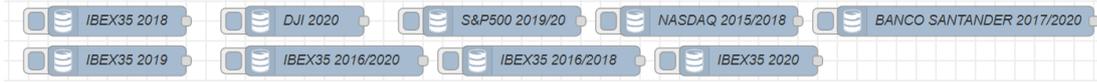


Figura 8.1: Índices y acciones.

En la figura 8.1 se pueden ver los índices en el editor de Node-RED.

```

1 - [
2   {
3     "Date": "2016-01-04",
4     "Open": 9429.799805,
5     "High": 9440.900391,
6     "Low": 9263.700195,
7     "Close": 9313.200195,
8     "Adj Close": 9313.19043,
9     "Volume": 279173100
10  },
11  {
12    "Date": "2016-01-05",
13    "Open": 9431.5,
14    "High": 9446.200195,
15    "Low": 9260.200195,
16    "Close": 9335.200195,
17    "Adj Close": 9335.19043,
18    "Volume": 208637600
19  },
20  {
21    "Date": "2016-01-06",
22    "Open": 9330.299805,
23    "High": 9330.299805,
24    "Low": 9122.599609,
25    "Close": 9197.400391,
26    "Adj Close": 9197.390625,
27    "Volume": 288658500
28  },
29  {
  
```

Figura 8.2: Valores requeridos en el inyector.

En la figura 8.2 podemos observar el contenido de los nodos inyectores que, para cada índice o acción, tendrán esos valores en formato JSON.

8.1.2 Indicadores

Se han implementado los indicadores técnicos de manera que a cada uno de ellos le corresponde un nodo *subflow*.

Estructura

La estructura interior de cada indicador es similar en todos. Como vemos en la figura 8.3, tenemos el nodo central con el nombre del indicador, que es donde se aplica la fórmula de cada

uno para calcular los valores que se van a mostrar en la gráfica de la interfaz del usuario. Otro nodo importante es el *Incidencias*, que calcula los días en los que debemos ejecutar acciones de compra y venta. Cada indicador tiene una manera de calcular estas incidencias, como se ha visto en el capítulo 2.4. Los nodos *Añadir datos*, *Parser* y *Señales* no tienen gran importancia ya que se utilizan para cambiar la estructura del contenido del mensaje para que sea más sencillo realizar operaciones con los datos en el siguiente nodo.

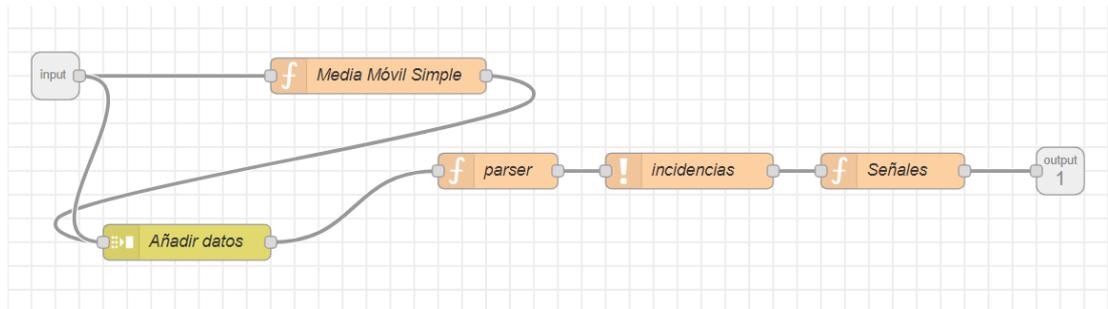


Figura 8.3: Interior del indicador.

Procedimiento

Para implementar cada indicador se han seguido una serie de pasos, que son comunes para todos los indicadores:

- En primer lugar, debemos calcular la línea o líneas del indicador que vamos a visualizar en la gráfica de la interfaz web. Esto se realiza en el nodo central con el nombre del indicador, que es un nodo *function* de Node-RED que permite realizar una función en Javascript. La entrada de este nodo es un mensaje que contiene los datos del activo.
- Una vez que se ha calculado el indicador, se utiliza el nodo *Añadir datos* para combinar los datos del activo y el indicador para que así formen parte de un único mensaje.
- Cuando ya se tiene el mensaje agrupado con los datos del activo y el indicador, lo siguiente que hay que hacer es pasar esta salida por el nodo *Parser*, que es otro nodo *function* de Node-RED que sirve para cambiar la estructura del contenido del mensaje. Se utiliza este tipo de nodos con la finalidad de cambiar el formato de los datos del mensaje, en concreto para pasar de un tipo objeto a una lista, todo ello en Javascript.
- Se calculan las incidencias en el nodo *Incidencias*. Cada indicador tiene una forma de calcularlas pero, en definitiva, todos tienen que analizar el mensaje de entrada y verificar si hay cruces entre indicador y activo, entre diferentes líneas del indicador o si el indicador traspasa cierto valor.

- Por último, en el nodo *Señales*, se incorporan en una lista los días en los que se producen acciones de compra, mientras que en otra lista se almacenan los datos correspondientes a las ventas.

Variables

Ya hemos visto que Node-RED nos deja declarar variables globales para un *subflow* y que sólo los nodos de su interior las pueden utilizar. Por lo tanto, para cada indicador, tenemos una serie de variables que podemos configurar. Estas pueden estar relacionadas con la formación de las líneas del indicador o con las acciones de compra y venta. Estas variables pueden cambiar dependiendo del indicador.

- En la figura 8.4 podemos observar las variables de los indicadores **SMA**, **EMA**, **Bandas de Bollinger** y **CCI**.

Periodos	<input type="text" value="25"/>
Vender el	<input type="text" value="30"/>
Señal venta	<input type="text" value="si"/>
Señal compra	<input type="text" value="si"/>
Cantidad nueva	<input type="text" value="500"/>

Figura 8.4: Variables de un indicador.

Cada variable tiene un formato que se debe cumplir:

- Periodos: valor numérico que representa el intervalo de tiempo a partir del cual queremos realizar el indicador.
- Señal venta: admite valores 'si' o 'no' dependiendo de si queremos realizar o no las señales de venta que propone el indicador.
- Vender el: valor numérico que indica el porcentaje de la cantidad invertida que queremos vender en las acciones de venta.
- Señal compra: admite valores 'si' o 'no' dependiendo de si queremos realizar o no las señales de compra que propone el indicador.
- Cantidad nueva: valor numérico que indica la cantidad que queremos añadir a nuestra inversión cuando se ejecuten acciones de compra.

- Para los indicadores **Ichimoku** y **MACD**, en lugar de definir un sólo periodo, debemos definir uno para cada línea que compone el indicador. A su vez, le damos al usuario la posibilidad de poder seleccionar con qué líneas del indicador quiere hacer las acciones de compra y venta (ver figuras 8.5 y 8.6).

Periodos de MACD	<input type="text" value="12"/>
Periodos de MACD 2	<input type="text" value="26"/>
Periodos señal	<input type="text" value="9"/>
Señal de compra	<input type="text" value="si"/>
Cantidad nueva	<input type="text" value="100"/>
Señal de venta	<input type="text" value="si"/>
% de venta	<input type="text" value="25"/>
Usar histograma	<input type="text" value="si"/>
Usar signal	<input type="text" value="si"/>
Histograma + signal	<input type="text" value="si"/>

Figura 8.5: Variables del indicador MACD.

Nube	<input type="text" value="a_z si"/>
Cruce de medias	<input type="text" value="a_z no"/>
Señal compra	<input type="text" value="a_z si"/>
Cantidad nueva	<input type="text" value="0_9 100"/>
Vender el	<input type="text" value="0_9 25"/>
Señal venta	<input type="text" value="a_z si"/>

Figura 8.6: Variables del indicador Ichimoku.

- Para el indicador **Estocástico**, en lugar de definir un sólo periodo, debemos definir uno para cada línea que compone el indicador. A mayores, debemos definir el valor que queremos para los límites de sobrecompra y sobreventa. A su vez, le damos al usuario la posibilidad de poder seleccionar con qué líneas del indicador quiere hacer las acciones de compra y venta (ver figura 8.7).

Periodos para K	<input type="text" value="0_9 5"/>
Periodos para D	<input type="text" value="0_9 3"/>
Señal compra	<input type="text" value="a_z si"/>
Cantidad nueva	<input type="text" value="0_9 500"/>
Señal venta	<input type="text" value="a_z si"/>
Vender el	<input type="text" value="0_9 35"/>
Inferior	<input type="text" value="0_9 20"/>
Superior	<input type="text" value="0_9 80"/>
Cruce con límites	<input type="text" value="a_z si"/>
Cruces entre %D y %K	<input type="text" value="a_z si"/>

Figura 8.7: Variables del indicador Estocástico.

- Con el indicador **RSI** seguimos el ejemplo de variables que tiene el indicador **SMA**. La única diferencia es que se define una variable para cada línea de sobrecompra y sobreventa (ver figura 8.8).

Periodos	<input type="text" value="14"/>
Superior	<input type="text" value="70"/>
Inferior	<input type="text" value="30"/>
Señal compra	<input type="text" value="si"/>
Cantidad nueva	<input type="text" value="100"/>
Señal venta	<input type="text" value="si"/>
Vender el	<input type="text" value="25"/>

Figura 8.8: Variables del indicador RSI.

- El indicador **Variación de porcentaje** no tiene en cuenta ningún periodo. En este caso, se define una variable de porcentaje máximo y una variable de porcentaje mínimo (ver figura 8.9).

Porcentaje maximo	<input type="text" value="10"/>
Señal venta	<input type="text" value="si"/>
Vender el	<input type="text" value="50"/>
Porcentaje minimo	<input type="text" value="5"/>
Señal compra	<input type="text" value="si"/>
Cantidad nueva	<input type="text" value="250"/>

Figura 8.9: Variables del indicador Variación de porcentaje.

- Para el indicador **Movimiento por fecha** lo único que debemos indicar es cada cuántos días queremos realizar acciones de compra o venta(ver figura 8.10).

Comprar cada	<input type="text" value="20"/>
Nueva cantidad	<input type="text" value="100"/>
Vender cada	<input type="text" value="30"/>
Vender el	<input type="text" value="50"/>

Figura 8.10: Variables del indicador Movimiento por fecha.

- El indicador **OBV** no presenta ninguna variable.

A estas variables se puede acceder en cualquiera de los nodos que forman el *subflow*. En la siguiente imagen vemos cómo acceder al valor de estas variables dentro de un nodo.

```
1 var period = env.get("Periodos")
2 var senalVenta = env.get("Señal venta")
3 var senalCompra = env.get("Señal compra")
4 var cantidadNueva = env.get("Cantidad nueva")
5 var porcenVenta = env.get("Vender el")
```

8.1.3 Estrategias

Una estrategia es un conjunto de procedimientos y operaciones que nos permiten evaluar la rentabilidad de una inversión. Estas estrategias dependen principalmente de acciones de compra y venta, que permiten variar la cantidad invertida en ese instante. Para las estrategias que se realizan en esta aplicación, se parte de una cantidad inicial y se van ejecutando las diferentes acciones de venta y compra. Para las acciones de compra, no podemos excedernos de la cantidad que se fija como inversión total. Para realizar las estrategias, hay que tener en cuenta dos nodos del flujo principal:

Definir estrategia

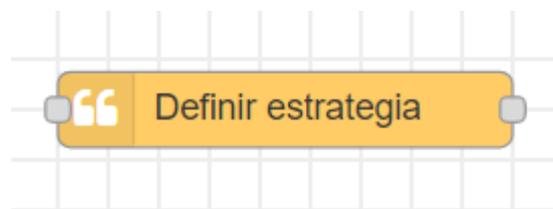


Figura 8.11: *Definir estrategia*.

Este nodo marca el comienzo de la estrategia. Tiene como único objetivo definir la inversión total que quiere aplicar el usuario, como puede verse en la figura 8.12. Para ello se utilizan las variables globales del nodo *subflow*. Ubicar este nodo antes de los indicadores permite normalizar los datos del activo en función de esta inversión total, que va a facilitar más adelante la manera en la que se visualizan las estrategias en la gráfica de la interfaz del usuario.

Inversión total

Figura 8.12: Inversión total.

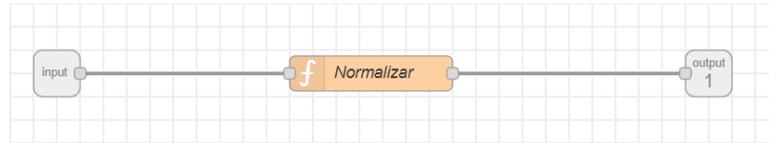


Figura 8.13: Contenido *Definir estrategia*.

En la figura 8.13 vemos el nodo *Normalizar*, dentro del *subflow Definir estrategia*. Este nodo normaliza los datos del activo realizando la siguiente operación:

$$\text{Activo}[n] = \frac{\text{Activo}[n]}{\text{Activo}[0] \times \text{Total}}, \quad (8.1)$$

donde *Activo* es el conjunto de valores del activo que se obtienen de la fuente de datos y *Total* es el valor de la inversión total que define el usuario.

Realizar estrategia

Este nodo permite realizar la estrategia, haciendo diferentes compras y ventas, dependiendo de lo que marquen las incidencias de los indicadores. A su vez, el usuario tiene que configurar en las variables del nodo, como puede verse en la figura 8.15, el parámetro de inversión inicial, que se usa como punto de partida de la estrategia.

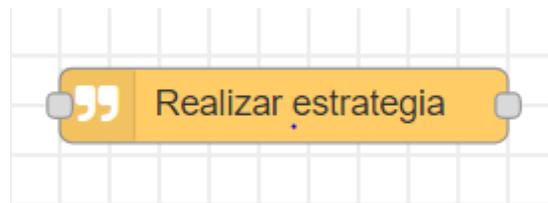


Figura 8.14: *Realizar estrategia*.

Inversion inicial

Figura 8.15: Inversión inicial.

El nodo *Realizar estrategia* es un *subflow*, por lo tanto tiene un grupo de nodos en su interior. En la figura 8.16 pueden verse los nodos que lo conforman.

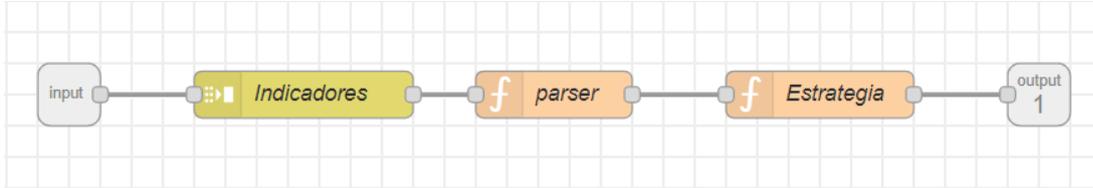


Figura 8.16: Contenido *Realizar estrategia*.

Cada nodo cumple una función específica:

- El nodo *Indicadores* tiene como función agrupar todos los indicadores que se utilizan en la estrategia.
- El nodo *Parser* modifica el contenido del mensaje para que sea más sencillo trabajar con él en los siguientes nodos.
- El nodo *Estrategia* es donde se realizan todas las operaciones de compra y venta, así como el resultado de la estrategia. Los datos de entrada de este nodo son el activo y los puntos de compra y venta que proporciona cada indicador. La operativa de este nodo es muy sencilla y se puede describir en estos pasos:

1. En primer lugar, hay que diferenciar entre el depósito y la liquidez. Se entiende como depósito la cantidad de dinero que está invertida. Por lo tanto, la liquidez es la diferencia entre la cantidad total de la inversión y el depósito.
2. Después, hay que recorrer todos los días en los que hay activo y verificar si en ese día hay que ejecutar alguna acción de compra o venta definida por algún indicador. Si es así, se realiza la acción y se actualiza el valor del depósito y de la liquidez.
3. Por otra parte, es necesario actualizar el valor del depósito en relación al día anterior, para ello utilizamos la siguiente fórmula:

$$\text{Deposito} = \text{Deposito} \times (1 + \text{Factor}), \quad (8.2)$$

donde:

$$\text{Factor} = \frac{\text{Actual} - \text{Anterior}}{\text{Anterior}}, \quad (8.3)$$

donde Actual es el valor de cierre del activo en el día actual y Anterior es el valor de cierre del activo del día anterior.

4. A continuación, hay que calcular el valor de la estrategia en cada día en función del depósito y de la liquidez. Para ello se suma el valor del depósito y el valor de la liquidez.

5. Por último, hay que calcular la rentabilidad de la estrategia para cada día. Para ello se utiliza la siguiente fórmula:

$$\text{Rentabilidad} = \frac{\text{Deposito} + \text{Liquidez} - \text{Total}}{\text{Total}} \times 100. \quad (8.4)$$

8.1.4 Interfaz



Figura 8.17: Gráfica.

La interfaz del usuario se desarrolla en el nodo Gráfica, un *subflow* que tiene como contenido (ver figura 8.18):

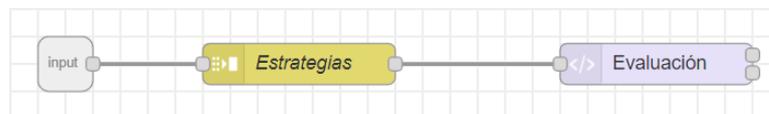


Figura 8.18: Contenido nodo Gráfica.

- El nodo *Estrategias* permite agrupar en un único mensaje todas las estrategias que se le quieren pasar a la interfaz del usuario.
- *Evaluación* es el nodo "UIBuilder" que permite crear la interfaz web del usuario en una URL determinada. Este nodo agrupa los ficheros necesarios para la realización de la interfaz, como pueden ser ficheros HTML, CSS o de Javascript. La entrada de datos que recibe este nodo es el conjunto de estrategias. La funcionalidad principal es recorrer todos los datos de entrada y mostrarlos mediante gráficas interactivas y tablas.

También va a permitir interactuar entre el flujo y la interfaz del usuario, utilizando para ello la librería Vue.js. En el fichero **index.js** se define un objeto Vue que nos permite obtener la información de los mensajes recibidos por el flujo, realizar operaciones con ello y mostrarlos en gráficas y tablas. Para las gráficas vamos a utilizar Apexchart, que nos permite generar gráficas interactivas pasándole los datos como una lista.

La comparación de estrategias se refleja en la interfaz del usuario gracias a la tabla resumen donde vemos los resultados de cada una en comparación con una estrategia base llamada **Buy&Hold**, que consiste en invertir la cantidad total desde el primer instante y mantenerlo hasta el último día, sin alterar la cantidad invertida durante todo el proceso. Esta comparación se visualiza en la gráfica, con una línea para cada estrategia y con el valor del activo reflejando la estrategia Buy&Hold.

En la tabla acciones, se muestran todas las acciones de compra o venta que se ejecutan durante la simulación, ordenadas por fecha. Esta información nos permite conocer en qué día fueron ejecutadas, qué indicador las lanza, a qué estrategia pertenece y el motivo por el que se ejecuta.

8.2 Interfaz Node-RED

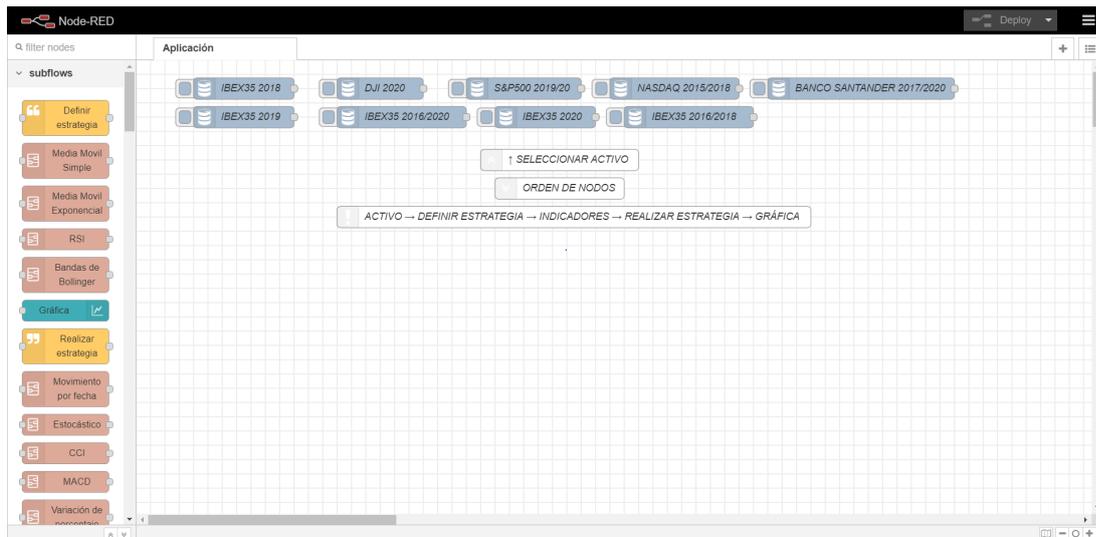


Figura 8.19: Interfaz Node-RED.

En la figura 8.19 podemos observar el editor de la interfaz Node-RED. En la parte lateral izquierda tenemos la paleta con los nodos *subflow* de nuestra aplicación. Estos nodos se arrastran hasta el editor central para realizar el flujo principal. Si damos doble clic en estos nodos, podemos editar las variables de configuración. La unión entre los nodos para hacer posible el intercambio de mensajes entre ellos se realiza conectando la salida de un nodo con la entrada del nodo siguiente. Es importante desplegar el flujo una vez esté finalizado, antes de inyectar los datos. Para ello es necesario pulsar el botón *Deploy* de la parte superior derecha (ver figura 8.20).

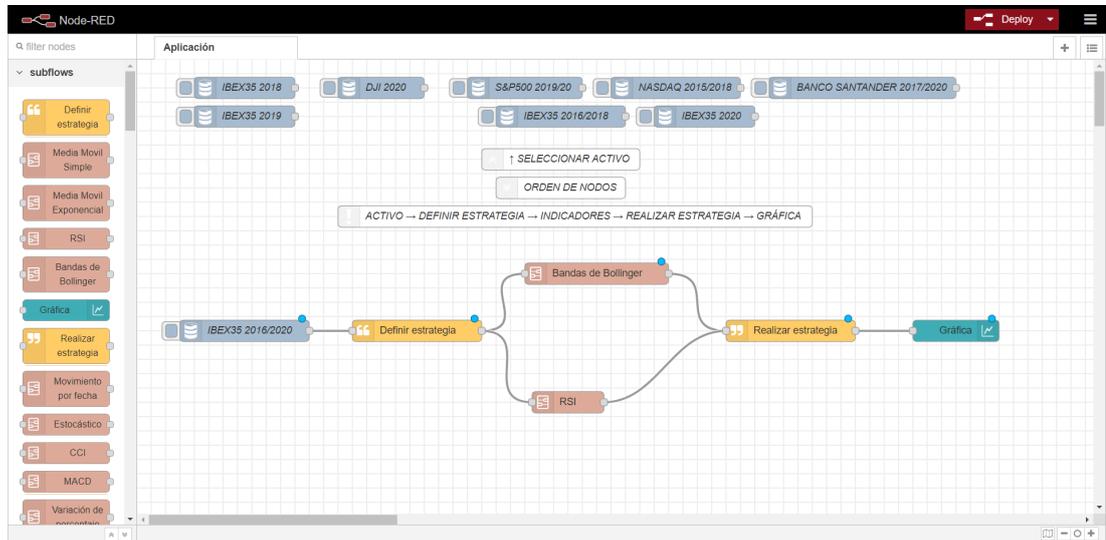


Figura 8.20: Flujo antes de pulsar *Deploy*.

El orden que se debe seguir para realizar cualquier estrategia es el siguiente:

- Seleccionar un índice o acción de todos los que hay en el editor central de Node-RED.
- Unir el índice o acción con el nodo *Definir estrategia*, que marca el inicio de la estrategia. Es posible definir un valor total para la inversión o se puede dejar por defecto.
- Después, tenemos que elegir los indicadores que queremos incluir en la estrategia. Para ello unimos el nodo *Definir estrategia* con el indicador que queramos incluir. Se permite configurar las variables de cada indicador o las podemos dejar por defecto.
- Una vez configurados los indicadores, tenemos que unir cada uno con el nodo *Realizar estrategia*, que es donde se va a hacer la estrategia. En este nodo tenemos que configurar con que inversión inicial vamos a empezar la inversión.
- Tenemos que conectar el nodo *Realizar estrategia* con el nodo *Gráfica* para poder visualizarlos en la interfaz del usuario.
- Por último, una vez esté acabado el flujo, tenemos que pulsar el botón *Deploy* para desplegar la aplicación. Ahora ya podemos inyectar los datos en el botón del nodo del activo seleccionado.

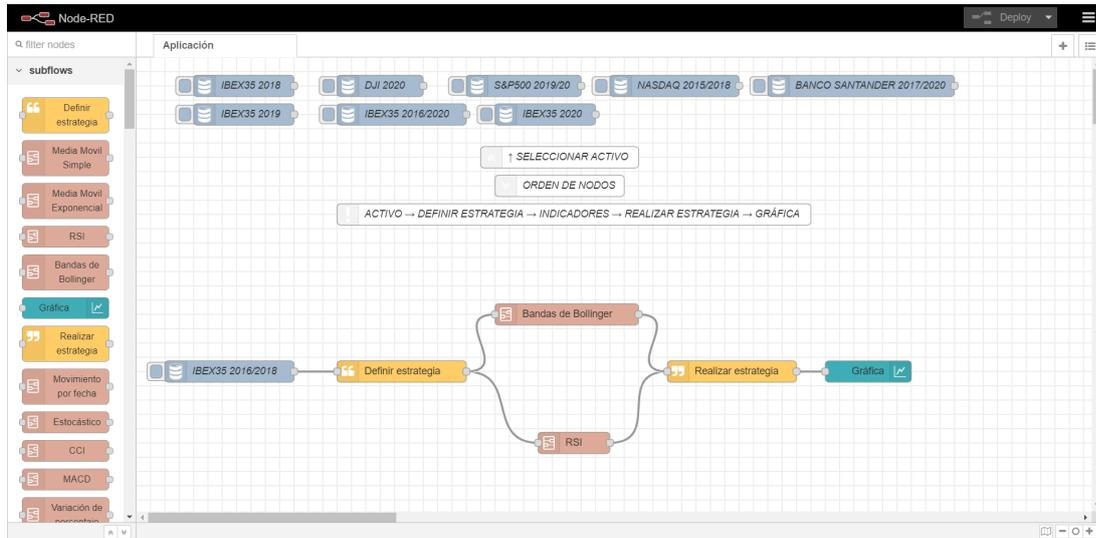


Figura 8.21: Implementación de una estrategia.

En la figura 8.21 podemos observar un ejemplo de un flujo que implementa una estrategia siguiendo los pasos anteriores.

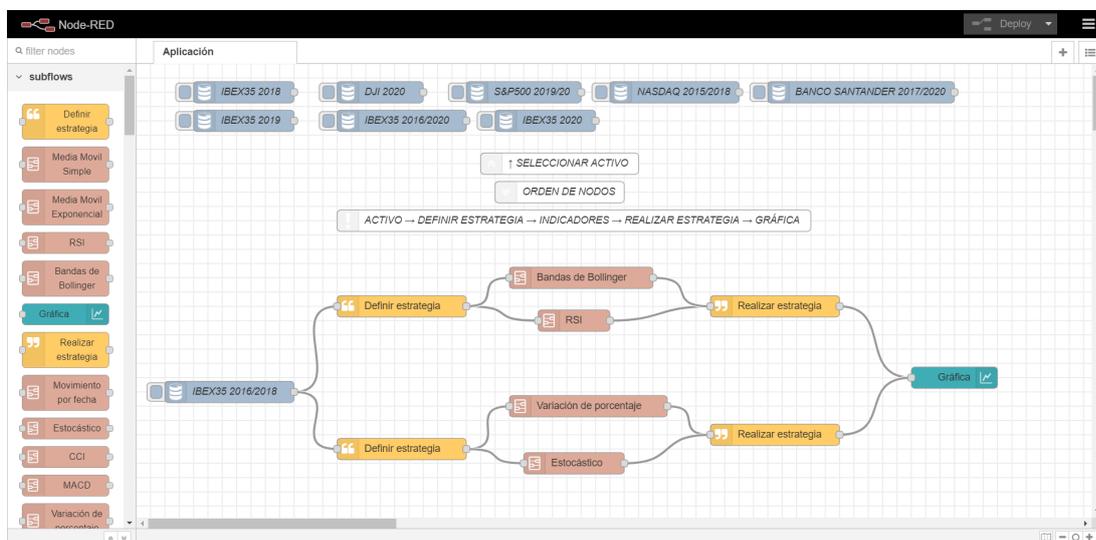


Figura 8.22: Varias estrategias.

En la figura 8.22 podemos observar un ejemplo de un flujo que implementa varias estrategias de inversión a la vez. Hay que tener en cuenta que si hacemos esto, el valor de inversión total de los nodos *Definir estrategia* debe ser igual, ya que queremos comparar estrategias sobre el mismo valor del activo, por lo tanto no tiene sentido poner valores diferentes en esta variable de los nodos.

8.3 Interfaz del usuario

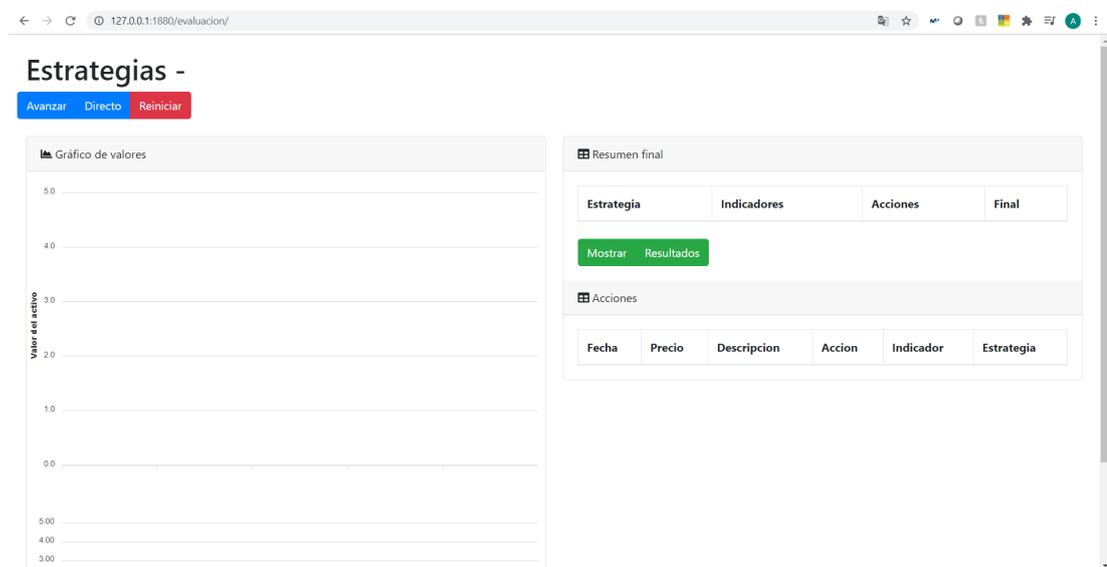


Figura 8.23: Interfaz del usuario.

En la figura 8.23, podemos observar la interfaz del usuario que se encuentra en la URL 127.0.0.1:1880/evaluacion. Aquí se pueden encontrar las gráficas y las tablas para poder analizar y comparar estrategias.

Se ha intentado dar el máximo protagonismo a la visualización de las gráficas, por lo tanto la interfaz es sencilla e intuitiva. Una vez se han inyectado los datos, estos se cargan en la pantalla automáticamente. Para poder iniciar la simulación, es necesario elegir si se hace día a día o directamente, pulsando el botón correspondiente.

Si la simulación se ha realizado directamente (ver figura 8.24), se pueden ver los resultados en la tabla resumen. Para visualizar sólo las líneas de las estrategias en las gráficas sólo hay que pulsar el botón *Resultados*. Si se quiere visualizar una estrategia completamente, con todos sus indicadores, se puede pulsar el botón *Mostrar* para ir cambiando la estrategia que aparece en la gráfica.

Si la simulación se ha realizado día a día (ver figura 8.25), también se pueden utilizar los botones *Resultados* y *Mostrar* en cualquier instante. Las tablas resumen y acciones se van rellenando a medida que va avanzando la simulación.

Para volver a visualizar todo por completo otra vez, basta con pulsar el botón *Directo*.

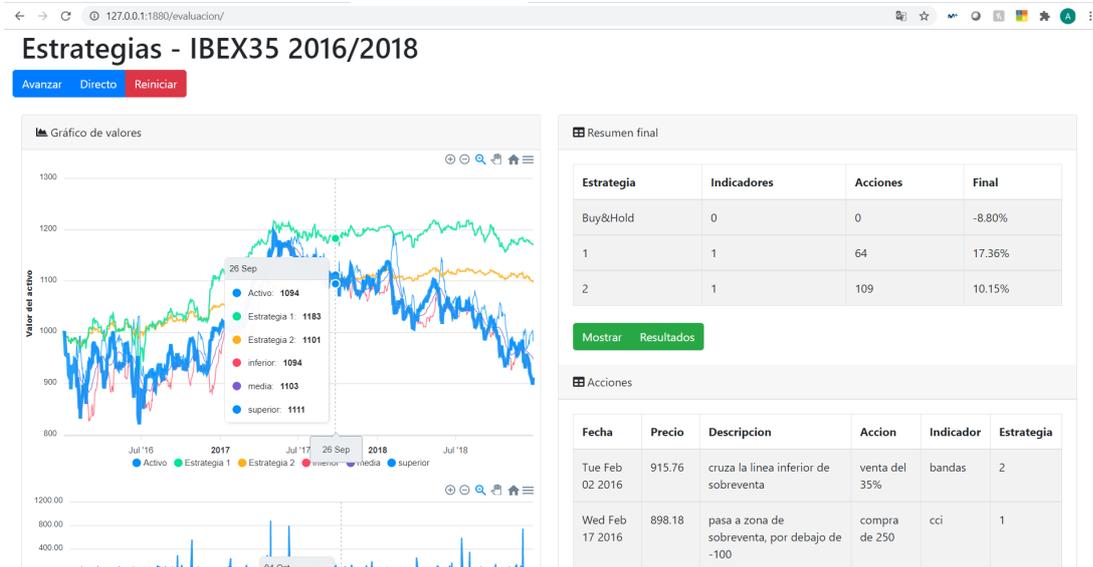


Figura 8.24: Directo.

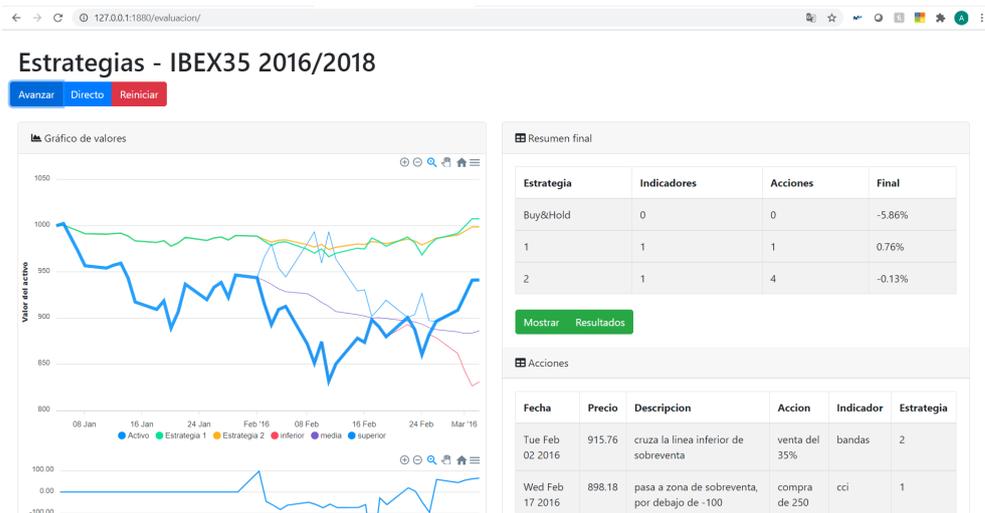


Figura 8.25: Día a día.

Se puede reiniciar la simulación con los mismos datos y las mismas estrategias sólo con pulsar el botón *Reiniciar*. Si ya ha finalizado una simulación y se quiere generar otras estrategias para evaluar, lo único que hay que hacer es repetir el proceso para generar el flujo e inyectar los datos de nuevo. La aplicación está programada para que en el momento en el que entran nuevos datos desde el flujo, se eliminen los anteriores y se haga la evaluación con los últimos datos recibidos.

8.4 Instalación

Para poder ejecutar la aplicación, es necesario correr el servidor Node-RED para que nos permita acceder al puerto 1880 de la dirección localhost (127.0.0.1). Node-RED proporciona una guía de instalación utilizando la librería **npm** desde un terminal de cualquier sistema operativo.

Para ahorrarnos todos esos pasos y que la aplicación sea más portable, se va a utilizar Docker [40], que nos va a permitir ejecutar una imagen del servidor Node-RED. Es necesario tener Docker instalado en el terminal donde se va a desplegar la aplicación. Se ha utilizado la versión gratuita, que permite utilizar las funcionalidades básicas de Docker en cualquier equipo.

La aplicación se encuentra en un directorio comprimido llamado **app.zip**, el cual debemos descomprimir en una ruta que conozcamos. Una vez realizado esto, hay que ejecutar el siguiente comando en el terminal:

```
1 sudo docker run -it -p 1880:1880 -v /ruta-del-directorio:/data  
   --name TFG nodered/node-red
```

En el comando, después de la opción **-v**, hay que poner la ruta donde se encuentra la carpeta **app** del proyecto de Node-RED. En este ejemplo, la ruta de la carpeta app es **/home/alejo/app**.

En la figura 8.26 se puede observar el servidor Node-RED corriendo. A partir de aquí, ya podemos buscar en el navegador la dirección localhost del dispositivo en el puerto 1880 (**127.0.0.1:1880**). En esta URL se nos va a abrir la interfaz principal de Node-RED con la que vamos a poder crear nuestro flujo. Una vez abierta esta página, es necesario abrir también la interfaz de usuario alojada en la URL **127.0.0.1:1880/evaluacion**, donde vamos a visualizar las estrategias realizadas en el flujo.

```
alejo@alejo-VirtualBox:~$ sudo docker run -it -p 1880:1880 -v /home/alejo/app:/data --name TFG nodered/node-red
> node-red-docker@1.0.6 start /usr/src/node-red
> node $NODE_OPTIONS node_modules/node-red/red.js $FLOWS "--userDir" "/data"
25 Jun 00:02:17 - [info]
Welcome to Node-RED
=====
25 Jun 00:02:17 - [info] Node-RED version: v1.0.6
25 Jun 00:02:17 - [info] Node.js version: v10.20.1
25 Jun 00:02:17 - [info] Linux 5.4.0-37-generic x64 LE
25 Jun 00:02:17 - [info] Loading palette nodes
25 Jun 00:02:19 - [info] +-----+
25 Jun 00:02:19 - [info] | uibuilder initialised:
25 Jun 00:02:19 - [info] |   root folder: /data/uibuilder
25 Jun 00:02:19 - [info] |   version . . : 2.0.7
25 Jun 00:02:19 - [info] |   packages . . : vue,bootstrap,bootstrap-vue,socket.io
25 Jun 00:02:19 - [info] +-----+
25 Jun 00:02:19 - [info] Settings file : /data/settings.js
25 Jun 00:02:19 - [info] Context store : 'default' [module=memory]
25 Jun 00:02:19 - [info] User directory : /data
25 Jun 00:02:19 - [warn] Projects disabled : editorTheme.projects.enabled=false
25 Jun 00:02:19 - [info] Flows file      : /data/flows.json
25 Jun 00:02:19 - [info] Creating new flow file
25 Jun 00:02:19 - [warn]
-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
25 Jun 00:02:19 - [info] Starting flows
25 Jun 00:02:19 - [info] Started flows
25 Jun 00:02:19 - [info] Server now running at http://127.0.0.1:1880/
```

Figura 8.26: Comando Docker.

Capítulo 9

Pruebas

En este capítulo se explican las pruebas realizadas en la aplicación.

La mayoría de aplicaciones programadas en lenguajes como Java o Python, tienen diferentes herramientas, como JUnit o Unittest, que permiten al programador realizar pruebas globales de la misma, comprobando así todas las funcionalidades presentes.

Por lo tanto, se ha optado por realizar pruebas funcionales de cada parte del flujo principal donde se realizan las estrategias y también de la interfaz de usuario donde visualizamos estas estrategias.

9.1 Pruebas funcionales

Se han llevado a cabo un conjunto de pruebas funcionales para comprobar todas las funcionalidades de la aplicación. En este apartado comentaremos algunas.

9.1.1 Inyectores

La prueba de estos nodos se realiza en cada ejecución de la aplicación ya que, para poder realizar todas las operaciones de la aplicación, es necesario inyectar los datos en el flujo principal. Esto se hace con el botón que tienen en el propio nodo y, siempre que los datos definidos en este nodo estén en el formato adecuado, la funcionalidad de este nodo siempre va a ser correcta.

9.1.2 Indicadores

Con los indicadores se han podido realizar diferentes pruebas, ya que su funcionamiento cambia dependiendo de la definición de variables que hagamos.

Para todos los indicadores se han probado diferentes opciones para crear las líneas del indicador. Esto se comprueba variando el valor de los periodos de cada línea. Para cada uno, también se ha probado a incorporarlo en una estrategia con las acciones de venta y compra que este proporciona.

Por otra parte, se han realizado pruebas en donde se configuraban los indicadores para que no proporcionaran acciones de compra y venta y, por lo tanto, estas acciones no se ejecutaban a lo largo de la estrategia aunque el indicador estuviera presente y se visualizara.

Además de estas pruebas, los indicadores **MACD**, **Estocástico** y el **Ichimoku** permiten realizar más pruebas, ya que podemos seleccionar con qué líneas del indicador queremos realizar las acciones de compra y venta.

9.1.3 Estrategias

Las pruebas realizadas en relación a la elaboración de las estrategias tienen que ver con la evaluación de diferentes configuraciones de valores para la inversión total y la cantidad inicial. También se tienen en cuenta las diferentes combinaciones de indicadores que se pueden dar. Una estrategia no tiene límite de indicadores para su formación. Un aspecto a tener en cuenta es probar que la estrategia no realice acciones de compra cuando no tenemos liquidez, ya que esto no tiene sentido y reduciría la fiabilidad de nuestra aplicación. Esto se ha probado, por ejemplo, añadiendo una cantidad inicial para la inversión muy próxima a la inversión total y, con ayuda de algún indicador que nos proporcione un número elevado de acciones de compra, comprobar que esas acciones no se realizan.

9.1.4 Visualización

Las pruebas realizadas en la interfaz web del usuario dependen del funcionamiento de los botones establecidos para los casos de uso. Por lo tanto, se han probado diferentes estrategias, realizando la simulación tanto día a día como directamente. También se ha probado el botón para mostrar sólo los resultados de las estrategias y el botón para mostrar una estrategia completa. Estas últimas se han realizado en cualquier día de la simulación y también en el último día, con un funcionamiento correcto. Estas pruebas de visualización se han realizado con el Product Owner, para verificar si cumplían con lo requerido.

Conclusiones

En este trabajo se ha elaborado una aplicación que permite al usuario realizar estrategias de inversión por medio de la conexión de nodos para formar un flujo. Estas aplicaciones son importantes ya que sirven para analizar el mercado de bolsa y ayudan a concienciar de lo complicado que es invertir en este tipo de mercados.

Tras acabar este Trabajo Fin de Grado, podemos decir que se ha conseguido:

- Elaborar indicadores del Análisis Técnico bursátil.
- Realizar una aplicación que nos permita crear estrategias de inversión y compararlas.
- Utilizar herramientas innovadoras y distintos estilos de programación.

Los resultados obtenidos por esta aplicación son muy satisfactorios. Los gráficos creados tienen una apariencia similar a los obtenidos con otras aplicaciones. Durante todo el proceso de realización, se ha priorizado la sencillez a la hora de mostrar los resultados. Esto es, las gráficas se han creado sólo con la información necesaria en cada momento, no mostrando datos que empeoren la visualización y no ayuden en la interpretación de las estrategias.

Los resultados dependen principalmente de la actuación del usuario en el momento de cargar los datos del activo que desee. Estos datos tienen que tener el formato especificado a lo largo de la memoria y es la base de la aplicación. El usuario también tiene que realizar correctamente la unión de los nodos. Si esto no se cumple, no se van a obtener los resultados esperados. El funcionamiento se ha explicado de manera detallada en esta memoria pero, en la interfaz de Node-RED donde se realiza el flujo, se ha incluido una guía de comentarios para poder realizar las estrategias de manera correcta.

Un aspecto importante es cómo se realizan las estrategias. Ninguna otra aplicación permite que la lógica de la estrategia se realice mediante la conexión de diferentes nodos. Esto

aporta al usuario visibilidad en todo momento de lo que está haciendo. Esta aplicación permite al usuario programar una estrategia de forma intuitiva, ágil y rápida sin tener conocimientos de ecuaciones matemáticas, de algoritmos ni de programación.

La realización de este trabajo ha permitido al alumno ganar experiencia en el análisis de requisitos de un proyecto y de sus casos de uso. También en el desarrollo de una aplicación web con un entorno poco conocido en la actualidad como Node-RED. Por otra parte, también ha conseguido adquirir conocimientos acerca de las inversiones en bolsa y de todos los conceptos básicos relativos a este mercado.

10.1 Líneas futuras de trabajo

La aplicación está lista para que todas sus funcionalidades puedan ser utilizadas en un entorno real. Esta aplicación es escalable ya que nos permite añadir funcionalidades fácilmente.

En el futuro, se podrían considerar los siguientes puntos:

- Añadir más indicadores técnicos para el análisis. Esto se podría realizar creando un nodo para cada indicador y permitiendo su visualización en la interfaz del usuario.
- Poder incorporar más gráficas para analizar diferentes estrategias en diferentes periodos al mismo tiempo. Esto da posibilidades al usuario para comprobar simultáneamente qué estrategia puede ir mejor en cada periodo.
- Permitir comparar estrategias con varios índices a la vez. Esto puede ayudar a interpretar en qué mercado puede encajar mejor una estrategia.
- Añadir diferentes operativas para ejecutar acciones de compra y venta. Un ejemplo puede ser "ponerse corto", que significa apostar cuando la bolsa va a caer (sea una acción o un índice), de forma que uno gana dinero si realmente la bolsa cae, pero si la bolsa sube, se perdería dinero.

Lista de acrónimos

CCI Commodity Channel Index. 17, 18

EMA Exponential Moving Average. 11

MA Moving Average. 10

MACD Moving Average Convergence Divergence. 1, 13, 72

OBV On Balance Volume. 1, 21, 22

RSI Relative Strength Index. 1, 12–14

SMA Simple Moving Average. 10, 11

Bibliografía

- [1] A. de la Loma, *Biografía de Charles Henry Dow*, 26/02/2012. [Online]. Available: <https://www.rankia.com/blog/bolsa-desde-cero/210307-biografia-charles-henry-dow>
- [2] *Metatrader*. [Online]. Available: <https://www.metatrader5.com/es>
- [3] *Plus500*. [Online]. Available: <https://www.plus500.es/>
- [4] *TradingView*. [Online]. Available: <https://es.tradingview.com/>
- [5] *Indices bursátiles*. [Online]. Available: <https://www.ig.com/es/indices/explicacion-indices>
- [6] *IBEX 35*. [Online]. Available: <https://es.finance.yahoo.com/quote/%5EIBEX?p=%5EIBEX>
- [7] *S&P 500*. [Online]. Available: <https://es.finance.yahoo.com/quote/%5EGSPC?p=%5EGSPC&.tsrc=fin-srch>
- [8] *DJI*. [Online]. Available: <https://es.finance.yahoo.com/quote/%5EDJI?p=%5EDJI&.tsrc=fin-srch>
- [9] *NASDAQ*. [Online]. Available: <https://es.finance.yahoo.com/quote/NQ=F?p=NQ=F&.tsrc=fin-srch>
- [10] *Euro Stoxx 50*. [Online]. Available: <https://es.finance.yahoo.com/quote/%5ESTOXX50E?p=%5ESTOXX50E>
- [11] F. León, *¿Cómo realizar un análisis fundamental de una empresa?*, 26/05/2020. [Online]. Available: <https://www.rankia.cl/blog/analisis-ipsa/3501534-como-realizar-analisis-fundamental-empresa>
- [12] J. I. D. Ugarte, *El Análisis Técnico Bursátil*. Díaz de Santos, 2015.
- [13] R. C. C., *Introducción al Análisis Chartista*. [Online]. Available: <https://www.tecnicadetrading.com/2009/10/introduccion-al-analisis-chartista.html>

- [14] *Medias Móviles*. [Online]. Available: <https://www.rankia.com/informacion/medias-moviles>
- [15] *Entrevista a Welles Wilder*. [Online]. Available: <http://biogramoney.blogspot.com/2015/12/entrevista-welles-wilder.html>
- [16] *RSI*. [Online]. Available: <https://admiralmarkets.es/education/articles/forex-indicators/como-sacar-beneficio-del-indicador-de-trading-rsi>
- [17] *MACD*. [Online]. Available: <https://admiralmarkets.es/education/articles/forex-indicators/macd>
- [18] *Estocastico*. [Online]. Available: <https://admiralmarkets.es/education/articles/forex-indicators/indicador-estocastico>
- [19] *Bandas de Bollinger*. [Online]. Available: <https://admiralmarkets.es/education/articles/forex-strategy/bandas-de-bollinger>
- [20] *cci*. [Online]. Available: <https://www.rankia.com/blog/bolsa-desde-cero/3907277-analizando-indicador-commodity-channel-index-cci>
- [21] *Ichimoku*. [Online]. Available: <https://compraraccionesdebolsa.com/formacion/tecnico/indicadores/ichimoku/>
- [22] *OBV*. [Online]. Available: <https://compraraccionesdebolsa.com/formacion/tecnico/indicadores/volumen/obv-on-balance-volume/>
- [23] *Biografía de Joseph Granville*. [Online]. Available: https://en.wikipedia.org/wiki/Joseph_Granville
- [24] J. Granville, *New Strategy Of Daily Stock Market Timing*.
- [25] *Git*. [Online]. Available: <https://git-scm.com/>
- [26] *Bitbucket*. [Online]. Available: <https://bitbucket.org/>
- [27] *Draw.io*. [Online]. Available: <https://drawio-app.com/>
- [28] *Node-RED*. [Online]. Available: <https://nodered.org/>
- [29] *Node.js*. [Online]. Available: <https://nodejs.org/es/>
- [30] *Javascript*. [Online]. Available: <https://www.javascript.com/>
- [31] *HTML*. [Online]. Available: <https://html.spec.whatwg.org/>

- [32] *Vue.js*. [Online]. Available: <https://vuejs.org/>
- [33] *Apexchart*. [Online]. Available: <https://apexcharts.com/>
- [34] *Bootstrap*. [Online]. Available: <https://getbootstrap.com/>
- [35] J. F. G. y. L. H. Carlos Ignacio Rivas, Verónica Paola Corona, “Metodologías actuales de desarrollo de software,” *Tecnología e Innovación*, vol. 2, no. 5, pp. 980–986, Diciembre 2015.
- [36] J. D. C. Bryan Molina Montero, Harry Vite Cevallos, “Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software,” *Espirales*, Junio 2018.
- [37] J. L. M. Galiano, *IMPLANTAR SCRUM CON ÉXITO*. Sonia Poch, 2016.
- [38] *Yahoo Finances*. [Online]. Available: <https://es.finance.yahoo.com/>
- [39] *CSVJSON*. [Online]. Available: <https://csvjson.com/>
- [40] *Docker*. [Online]. Available: <https://www.docker.com/>

