



Universidad
Carlos III de Madrid



Universidad
Carlos III de Madrid
www.uc3m.es

Máster Universitario en Tecnologías de la Computación
Aplicadas al Sector Financiero 2016-2017

Trabajo Fin de Máster

Sistema de recomendación para la toma de decisiones de inversión basado en análisis técnico

Autor: Rubén García Sánchez

Tutor: Antonio Manuel Aguilar Arévalo

Cotutor: Tomás Eduardo de la Rosa Turbides



Agradecimientos

A mis padres, sobre todo por aguantarme en los momentos de estrés, pero también por ofrecerme todo lo que está en su mano, apoyarme en mis decisiones y ofrecerme la posibilidad de estudiar lo que quería. Gracias por todo.

A mi novia, por aguantarme y apoyarme en los peores momentos, por estar ahí siempre que lo necesito y por mucho más. Gracias por todo.

A Antonio, por la acogida en RDSistemas, por ofrecerme la posibilidad de hacer este trabajo y por guiarnos y enseñarnos durante el desarrollo del mismo.

A Victor, cuyo trabajo estaba muy ligado al mío y lo hemos desarrollado apoyándonos mutuamente.

A Tomás, por colaborar en el desarrollo del trabajo.

A mi hermana, a su novio y a mis amigos por interesarse y ayudarme a desconectar.



Resumen

Los fondos de inversión son un vehículo de inversión muy importante en la actualidad del mundo financiero y que, cada vez con más frecuencia, son más utilizados para canalizar el ahorro. Estos mueven grandes cantidades de dinero que los distintos participantes aportan con el objetivo de obtener una rentabilidad y evitar tener el dinero parado sin generar ningún tipo de riqueza. Este dinero que los partícipes invierten en un fondo debe ser gestionado por alguien que tenga unos conocimientos mínimos en finanzas y en los mercados financieros de forma que con sus decisiones de inversión genere riqueza para los partícipes.

Para conseguir riqueza o una rentabilidad positiva, los gestores deben basarse en algo que les pueda ayudar a invertir en un activo financiero u otro. Esta tarea es una tarea compleja ya que nadie sabe con certeza que va a pasar en el futuro. Sin embargo, existen diferentes herramientas que facilitan a los inversores realizar esta tarea de predicción o, en algunos casos, de cobertura, basándose en distintos indicadores que permiten orientar a los gestores en sus decisiones.

Estos indicadores pertenecen a diversas categorías y son de distinto tipo. Podemos agruparlos en 5 categorías: macroeconómicos, microeconómicos, coyunturales, técnicos y fundamentales. En este trabajo nos centraremos en los indicadores que pertenecen al análisis bursátil, que son aquellos pertenecientes a las corrientes de análisis técnico y análisis fundamental.

A grosso modo, el análisis técnico es aquel que se basa en el estudio de las cotizaciones y volúmenes de negociación históricos de cada activo para predecir movimientos futuros, ya que se piensa que lo que ocurre en el pasado volverá a ocurrir en el futuro. Dentro del análisis técnico existen también dos categorías de análisis frecuentemente usadas en conjunto, el análisis chartista, que se basa en el uso de gráficos y reconocimiento de patrones o figuras, y el cuantitativo. Por su parte, el análisis fundamental se basa en el estudio de la empresa (sus cuentas, su contabilidad, sus ratios, su competencia o sus productos) para detectar posibles infravaloraciones o sobrevaloraciones y actuar en consecuencia. Ambas corrientes tienen sus propios indicadores y son corrientes igual de válidas.

Los gestores, a la hora de invertir, normalmente hacen uso de indicadores de una de estas dos vertientes de análisis comentadas. Por ello, se quiere realizar una aplicación software que ayude a los gestores en la toma de decisiones de inversión, en este caso, basándose en indicadores de análisis técnico.



Abstract

Nowadays, investment funds are one of the most important investment entities in the financial environment. These entities manage large amounts of money which is contributed by all the investors participating in the investment fund with the objective of obtaining a return. This contributed money, must be managed by someone who is an expert in the task, someone who has deep knowledge in finances and in financial markets because this person will be the one that, with his decisions, will generate profits or losses for the investors.

To achieve profits, fund's managers must base their investment decisions on something, something that helps them or indicate them in which securities may be better to invest or which securities should be sold. There are two main analytical techniques used for achieving this purpose, both based in the use of indicators: technical analysis and fundamental analysis.

Both techniques, which are obviously susceptible to failure, are completely different each other. Technical analysis uses securities historical data for calculating the different indicators because it's thought that past situations will be repeated in the future. In the other hand, fundamental analysis is based on present data of the listed companies in order to detect underestimate or overrated securities.

Fund's managers usually use one of this two analysis techniques to support their investment decisions. This analysis is their main tool for working and is something that they use every day, therefore, to easy this process of investment for them, we decided to develop this technical analysis application.



Índice de contenido

Agradecimientos	1
Resumen.....	2
Abstract	3
Índice de ilustraciones.....	7
Índice de Tablas.....	8
1. Introducción	10
1.1 Motivación	10
1.2 Objetivos	12
1.3 Estructura del documento.....	13
1.4 Acrónimos y abreviaturas.....	14
2. Estado del arte	15
2.1 Análisis bursátil de mercados.....	15
2.1.1 Análisis técnico.....	15
2.1.1.1 Indicadores de momentum.....	16
2.1.1.1.1 RSI.....	17
2.1.1.1.2 Oscilador Estocástico	18
2.1.1.2 Indicadores de tendencia	21
2.1.1.2.1 Medias móviles	21
2.1.1.2.2 MACD	23
2.1.1.3 Indicadores de volatilidad	25
2.1.1.3.2 Bandas de Bollinger.....	25
2.1.1.4 Indicadores de volumen.....	27
2.1.1.4.1 EMV	27
2.1.2 Análisis fundamental.....	30
2.1.3 Comparativa	31
2.2 Antecedentes en software de análisis técnico.....	32
2.2.1 Finviz.....	32
2.2.2 Investing	33
2.3 Obtención de los datos	34
2.3.1 Bloomberg.....	35
2.3.2 Reuters	35
2.3.3 Yahoo Finance	35
2.3.4 Los mercados.....	36
2.4 Backtesting	36
2.5 Arquitectura SOA.....	37



2.6	Sistemas de soporte a la decisión	38
2.6.1	Trading algorítmico	39
3.	Análisis del sistema	40
3.1	Detalle de la aplicación	40
3.1.1	Recomendaciones de compraventa	41
3.1.2	Estrategias	41
3.1.3	Backtesting	42
3.1.4	Autoajuste	43
3.2	Requisitos de Usuario	44
3.2.1	Requisitos de capacidad	45
3.2.2	Requisitos inversos	48
3.2.3	Requisitos de restricción	48
3.3	Casos de uso	50
3.4	Requisitos Software	56
3.4.1	Requisitos no funcionales	56
3.4.2	Requisitos funcionales	59
3.4.3	Requisitos de interfaz	64
4.	Diseño	66
4.1	Arquitectura del sistema	66
4.2	Especificación del entorno tecnológico	68
4.3	Diseño de la base de datos	68
4.4	Diseño de la interfaz	77
4.5	Decisiones de diseño	84
5.	Implementación	86
5.1	Obtención de cotizaciones	86
5.2	Implementación de la lógica de compraventa en R	87
5.3	Implementación de los Web Services	89
5.3.1	Web service para la obtención de cotizaciones	89
5.3.2	Web service para los procesos de cálculo	89
5.3.3	Web service para consulta de históricos	93
5.3.4	Web service para el tratamiento de las configuraciones	94
5.4	Implementación del servlet	96
5.5	Implementación de los handlers	97
5.6	Implementación de la interfaz	102
6.	Pruebas	103
6.1	Pruebas unitarias	103



6.2 Pruebas de integración	109
7. Planificación	115
8. Presupuesto.....	120
9. Marco regulador.....	122
9.1 Java	122
9.2 R.....	122
9.3 Rserve.....	122
9.4 Yahoo finance	122
9.5 PostgreSQL	123
9.6 Apache Tomcat.....	123
10. Evaluación	124
10.1 Verificación de requisitos de usuario	124
10.2 Ejemplos	125
11. Conclusiones.....	129
11.1 Conclusiones.....	129
11.2 Líneas futuras	130
11.3 Relación del trabajo con el máster cursado	131
12.Referencias.....	132

Índice de ilustraciones

<i>Ilustración 1 - RSI zonas de tendencia</i>	18
<i>Ilustración 2 - Ejemplo del oscilador estocástico. En azul K, en naranja D.</i>	20
<i>Ilustración 3 - Ejemplo MACD</i>	24
<i>Ilustración 4 - Bandas de Bollinger (Verde = Banda superior, rojo = banda inferior, marrón = banda medi, azul = precio de cierre del activo)</i>	26
<i>Ilustración 5 - Equivolume box</i> <i>http://stockcharts.com/school/doku.php?id=chart_school:chart_analysis:equivolume</i>	28
<i>Ilustración 6 - EMV(azul) y maEMV(naranja)</i>	29
<i>Ilustración 7 - Ejemplo de análisis técnico en la versión gratuita de Finviz [12]</i>	32
<i>Ilustración 8 - Ejemplo de análisis técnico en Investing [13]</i>	33
<i>Ilustración 9 - Casos de uso</i>	50
<i>Ilustración 10 - Diagrama de la arquitectura</i>	66
<i>Ilustración 11 - Esquema de base de datos</i>	69
<i>Ilustración 12 - Interfaz: Login</i>	77
<i>Ilustración 13 - Interfaz: Menú de Navegación</i>	77
<i>Ilustración 14 - Interfaz: Menú Análisis Técnico</i>	78
<i>Ilustración 15 - Interfaz: Pantalla de recomendaciones</i>	78
<i>Ilustración 16 - Interfaz: Pantalla de recomendaciones específica de un activo</i>	79
<i>Ilustración 17 - Interfaz: Pantalla Histórico de Recomendaciones</i>	80
<i>Ilustración 18 - Interfaz: Pantalla Histórico de Recomendaciones de estrategias autoajustadas</i>	80
<i>Ilustración 19 - Interfaz: Pantalla de inversiones</i>	81
<i>Ilustración 20 - Interfaz: Pantalla de configuración</i>	81
<i>Ilustración 21 - Interfaz: Pantalla de Configuración de Estrategias</i>	82
<i>Ilustración 22 - Interfaz: Pantalla de Configuración de las Órdenes Automáticas</i>	83
<i>Ilustración 23 - Interfaz: Ventana emergente</i>	83
<i>Ilustración 24 - Ejemplo Indicador en R</i>	88
<i>Ilustración 25 - Planificación Inicial del TFM</i>	117
<i>Ilustración 26 - Planificación Real del TFM</i>	118
<i>Ilustración 27 - Planificación del Proyecto</i>	119
<i>Ilustración 28 - Tabla de base de datos</i>	125
<i>Ilustración 29 - Resultado ejecución recomendación de compraventa.</i>	126
<i>Ilustración 30 - Bandas de Bollinger</i>	126
<i>Ilustración 31 - MACD</i>	127
<i>Ilustración 32 - RSI</i>	127
<i>Ilustración 33 - Ejecución backtesting</i>	128



Índice de Tablas

<i>Tabla 1-Ejemplo requisito de usuario</i>	44
<i>Tabla 2-RUC 01</i>	45
<i>Tabla 3-RUC 02</i>	45
<i>Tabla 4- RUC 03</i>	45
<i>Tabla 5- RUC 04</i>	45
<i>Tabla 6- RUC 05</i>	45
<i>Tabla 7-RUC 06</i>	46
<i>Tabla 8-RUC 07</i>	46
<i>Tabla 9-RUC 08</i>	46
<i>Tabla 10-RUC 09</i>	46
<i>Tabla 11-RUC 10</i>	46
<i>Tabla 12-RUC 11</i>	47
<i>Tabla 13-RUC 12</i>	47
<i>Tabla 14-RUC 13</i>	47
<i>Tabla 15-RUC 14</i>	47
<i>Tabla 16-RUC 15</i>	47
<i>Tabla 17-RUI 01</i>	48
<i>Tabla 18-RUR 01</i>	48
<i>Tabla 19-RUR 02</i>	48
<i>Tabla 20-RUR 03</i>	48
<i>Tabla 21-RUR 04</i>	48
<i>Tabla 22-RUR 05</i>	49
<i>Tabla 23-RUR 06</i>	49
<i>Tabla 24-RUR 07</i>	49
<i>Tabla 25-RUR 08</i>	49
<i>Tabla 26-RUR 09</i>	49
<i>Tabla 27-CU 01</i>	51
<i>Tabla 28-CU 02</i>	51
<i>Tabla 29-CU 03</i>	51
<i>Tabla 30-CU 04</i>	52
<i>Tabla 31-CU 05</i>	52
<i>Tabla 32-CU 06</i>	53
<i>Tabla 33-CU 07</i>	53
<i>Tabla 34-CU 08</i>	54
<i>Tabla 35-CU 09</i>	54
<i>Tabla 36-CU 10</i>	54
<i>Tabla 37-CU 11</i>	55
<i>Tabla 38-CU 12</i>	55
<i>Tabla 39-CU 13</i>	55
<i>Tabla 40 - Ejemplo tabla requisito software</i>	56
<i>Tabla 41 RNF-01</i>	56
<i>Tabla 42 RNF-02</i>	57
<i>Tabla 43 RNF-03</i>	57
<i>Tabla 44 RNF-04</i>	57



<i>Tabla 45 RNF-05</i>	57
<i>Tabla 46 RNF-06</i>	57
<i>Tabla 47 RNF-07</i>	58
<i>Tabla 48 RNF-08</i>	58
<i>Tabla 49-RNF 09</i>	58
<i>Tabla 50-RNF 10</i>	58
<i>Tabla 51-RF 01</i>	59
<i>Tabla 52-RF 02</i>	59
<i>Tabla 53-RF 03</i>	59
<i>Tabla 54-RF 04</i>	59
<i>Tabla 55-RF 05</i>	60
<i>Tabla 56-RF 06</i>	60
<i>Tabla 57-RF 07</i>	60
<i>Tabla 58-RF 08</i>	60
<i>Tabla 59-RF 09</i>	61
<i>Tabla 60-RF 10</i>	61
<i>Tabla 61-RF 11</i>	61
<i>Tabla 62-RF 12</i>	61
<i>Tabla 63-RF 13</i>	62
<i>Tabla 64-RF 14</i>	62
<i>Tabla 65-RF 15</i>	62
<i>Tabla 66-RF 16</i>	62
<i>Tabla 67-RF 17</i>	63
<i>Tabla 68-RF 18</i>	63
<i>Tabla 69-RF 19</i>	63
<i>Tabla 70-RF 20</i>	63
<i>Tabla 71-RF 21</i>	64
<i>Tabla 72-RF 22</i>	64
<i>Tabla 73-RI 01</i>	64
<i>Tabla 74-RI 02</i>	65
<i>Tabla 75-RI 03</i>	65
<i>Tabla 76-RI 04</i>	65
<i>Tabla 77-RI 05</i>	65
<i>Tabla 78-Tabla instrumentos</i>	70
<i>Tabla 79-Tabla cotizaciones</i>	70
<i>Tabla 80 - Tabla carteraGestor</i>	70
<i>Tabla 81-Tabla carteras</i>	71
<i>Tabla 82-Tabla patrimonio</i>	71
<i>Tabla 83-Tabla gestores</i>	71
<i>Tabla 84-Tabla Parámetros para órdenes</i>	72
<i>Tabla 85- Tabla indicadores</i>	72
<i>Tabla 86-Tabla histórico de indicadores</i>	72
<i>Tabla 87-Tabla de estrategias</i>	73
<i>Tabla 88-Tabla definición de estrategias</i>	73
<i>Tabla 89-Tabla histórico de recomendaciones</i>	74
<i>Tabla 90-Tabla de órdenes</i>	74
<i>Tabla 91-Tabla backtesting</i>	75
<i>Tabla 92- Tabla brokers</i>	75
<i>Tabla 93 - Tabla operaciones</i>	76
<i>Tabla 94 - Tabla autoajuste</i>	76
<i>Tabla 95 - Planificación inicial TFM</i>	116
<i>Tabla 96 - Planificación real TFM</i>	118
<i>Tabla 97 - Planificación del proyecto</i>	119



Tabla 98 - Presupuesto: Personal.....	120
Tabla 99 - Presupuesto: Equipos.....	120
Tabla 100 - Presupuesto: Software.....	120
Tabla 101 - Presupuesto: Material Fungible.....	120
Tabla 102 - Presupuesto: Otros Gastos.....	120
Tabla 103 - Presupuesto: Resumen.....	121
Tabla 104 - Verificación de requisitos.....	125

1. Introducción

1.1 Motivación

Los fondos de inversión son una de las principales opciones que existen para que los pequeños inversores puedan invertir su dinero, depositando su capital en esta estructura buscando optimizar sus inversiones. Los fondos de inversión consisten en una agregación de capitales procedente de distintos inversores con el objetivo de juntar un capital de mayor cuantía para realizar inversiones mejores, más complejas y completas, ahorrando también en costes a la hora de invertir (comisiones). Por ejemplo, la mayoría de las personas de clase media no expertas en el sector financiero no tienen conocimientos financieros suficientes ni capacidad económica suficiente para crear una cartera bien diversificada. Sin embargo, si deposita su dinero en un fondo junto con el de otros inversores esta restricción económica que se comentó anteriormente desaparece. La otra restricción mencionada, la falta de conocimientos de inversión, es la que cubrirá el gestor del fondo de inversión.

Los gestores de fondos son personas que fundamentalmente se dedican a gestionar el patrimonio del fondo para el que trabajan, en nombre de los partícipes. El objetivo fundamental de los gestores es conseguir el mejor rendimiento posible en concordancia con el riesgo que aceptan correr los partícipes [1]. Esta tarea no es sencilla ni es algo trivial, tienen que decidir en que invierten una gran cantidad de dinero, por ello es necesario que sean expertos conocedores de los mercados de inversión a los que acuden, así como tener unas características psicológicas adecuadas para el trabajo.

Los gestores, conocedores de los mercados en los que invierten, normalmente hacen uso de indicadores de distinta índole para estudiar el mercado y decidir en que invertir. En este trabajo nos centraremos únicamente en el análisis bursátil. Dentro del análisis bursátil existen dos corrientes bien diferenciadas y frecuentemente enfrentadas por los expertos del sector financiero, el análisis técnico y el análisis fundamental

De forma muy general, el análisis técnico consiste en seleccionar acciones de empresas observando la evolución histórica. El análisis fundamental, por su parte,

consiste en estimar cuál es el valor intrínseco de una empresa basándose en su estado actual [2].

En la actualidad, los gestores de fondos hacen uso de herramientas de análisis fundamental y análisis técnico para ayudarles en la toma de decisiones de inversión. Existen numerosas herramientas software en el mercado para realizar estos análisis, en las que para un activo dado puedes obtener la información de numerosos indicadores pertenecientes a ambos tipos de análisis.

Si ya existe, ¿por qué queremos hacer una herramienta de soporte a la decisión basado en análisis técnico? Este proyecto tiene dos características que le hacen ser diferentes. Dentro de la empresa en la que se quiere llevar a cabo este proyecto existe una diversidad de soluciones software dedicado para gestoras de fondos de inversión que abarcan desde el front-office financiero hasta el back-office financiero, pero carece de una herramienta de ayuda a la toma de decisiones para el gestor. Todas las operaciones que realiza el gestor quedan registradas en estos sistemas disponiendo de toda la información histórica y actual de las distintas carteras. Esta información es algo que las demás herramientas de análisis técnico del mercado no disponen ya que únicamente se dedican a la obtención de los datos históricos en mercado de un activo y muestran los resultados dados por cada indicador.

Gracias a esta información, la integración de la herramienta de soporte a la decisión basado en análisis técnico va a permitir al gestor, además de su utilidad típica, ver si la estrategia seguida basándose en unos u otros indicadores está resultando beneficiosa o no. Además, se desarrollará una funcionalidad que permitirá reajustar las estrategias creadas por el gestor para intentar optimizar los resultados obtenidos con una determinada estrategia.

De esta forma, se pretende añadir al sistema existente un módulo del que carece y que seguramente los gestores utilizan a través de otra plataforma con el añadido de permitir al gestor combinar indicadores creando estrategias ponderadas, ver si las estrategias de inversión seguidas y creadas son buenas o no y enviar órdenes de compra o venta al mercado, a través de un bróker, tras una recomendación.



1.2 Objetivos

El objetivo fundamental de este trabajo es diseñar y desarrollar un sistema que ayude a los gestores de fondos de inversión a la toma de decisiones en la gestión de sus carteras con activos de renta variable.

El programa implementará un conjunto de indicadores de análisis técnico que el gestor podrá consultar y visualizar gráficamente. Además, el gestor podrá definir sus propias estrategias agrupando varios de estos indicadores y otorgándole un peso de influencia a cada uno. El programa dará un feedback de compra o venta en base al activo consultado y al indicador o estrategia seleccionado.

Otras características son:

- El gestor podrá probar (backtesting) las estrategias creadas con datos del pasado para comprobar si su creación es, al menos en el pasado, válida.
- Envío de órdenes a mercado a través de un bróker, en base a una recomendación y a una configuración previa.
- Visualización de las inversiones realizadas y ver las pérdidas o ganancias que está teniendo con las mismas.
- Autoajuste de estrategias con el objetivo de optimizar la estrategia original del gestor.

1.3 Estructura del documento

En este apartado se listan los distintos capítulos que forman este documento y una breve descripción de cada uno de ellos:

- 1. Introducción: describe la motivación y los objetivos de este proyecto.
- 2. Estado del arte: describe de forma teórica, la historia y actualidad del entorno que rodea al proyecto, así como la relación de este proyecto con otros existentes.
- 3. Análisis del sistema: describe los distintos requisitos que debe tener el sistema y las restricciones del mismo.
- 4. Diseño del sistema: describe la arquitectura adoptada por el sistema, diseño de base de datos, diseño de interfaz de usuario y decisiones de diseño.
- 5. Implementación: describe la solución adoptada para desarrollar el sistema y cumplir los requisitos iniciales.
- 6. Pruebas: describe las distintas pruebas a realizar sobre el software para verificar su correcta implementación.
- 7. Planificación: describe la planificación inicial y final del TFM, también una posible planificación real del proyecto.
- 8. Presupuesto: contiene el cálculo de costes detallado de la realización del proyecto.
- 9. Marco regulador: contiene las referencias de los distintos aspectos legales del software empleado.
- 10. Evaluación: Verificación de los requisitos de usuario cumplidos y ejemplos del funcionamiento de la herramienta.
- 11. Conclusiones: conclusiones del proyecto, posibles trabajos futuros y relación del TFM con el máster cursado.



1.4 Acrónimos y abreviaturas

RSI: Índice de fuerza relativa

SMA: Simple Moving Average, Media móvil simple.

EMA: Exponential Moving Average, Media móvil exponencial.

MACD: Moving Average Convergence Divergence, Convergencia/Divergencia del Promedio Móvil.

EMV: Ease of Movement, Facilidad de movimiento.

PER: Price-to-earnings ratio, ratio precio-beneficio.

ROI: Return On Investment, Retorno sobre la inversión.

WS: Web service, Servicio Web.

SOA: Service Orientated Architecture, Arquitectura Orientada a Servicios.

GLP: General Public License

2. Estado del arte

En este apartado se analizará la situación actual de las tecnologías que se pueden utilizar para desarrollar este proyecto, así como la situación actual en lo referente al análisis bursátil de los mercados.

2.1 Análisis bursátil de mercados

El análisis bursátil consiste en el estudio de los mercados financieros con el objetivo de intentar predecir el valor y la evolución de los precios de los activos financieros. Está dividido en dos tipos de análisis diferentes: análisis técnico y análisis fundamental, los cuales se tratarán con mayor profundidad a continuación.

2.1.1 Análisis técnico

Definición

El análisis técnico financiero es una metodología de análisis de activos para predecir la dirección de los precios. Consiste en seleccionar activos financieros observando la evolución histórica de los movimientos del precio de la acción y del volumen de contratación, a través de gráficos, fórmulas matemáticas y estadísticas. Se basa en el supuesto de que los precios de las acciones se mueven según unas tendencias determinadas que se van repitiendo en el tiempo [2].

Los analistas técnicos se basan en dos premisas. La primera de ellas es que todo aquello que pueda afectar al precio de un activo, se verá reflejado en su precio de mercado. Esto significa que los movimientos de los precios deberían reflejar los cambios de la oferta y la demanda, si la demanda supera la oferta, los precios deberían subir, si la oferta supera la demanda, los precios deberían bajar. Si todo lo que afecta al precio de mercado queda reflejado en dicho precio, entonces lo único que hace falta es estudiar ese precio de mercado, haciendo uso de gráficos e indicadores [3].

La segunda premisa es que los precios se mueven por tendencias. El único objetivo que tiene la representación gráfica de los movimientos de los precios es el hecho de ser capaces de identificar tendencias en su etapa más reciente con el fin de que las transacciones vayan en la dirección que indican las tendencias [3].

Diferencia del análisis técnico y el chartista

Dependiendo de la literatura consultada, hay autores que incluyen el análisis chartista dentro del análisis técnico y otros que los separan en categorías diferentes. Indistintamente de la clasificación, a continuación, se tratará de comentar la diferencia fundamental entre ambas técnicas.

El analista chartista utiliza como herramienta principal los gráficos, todo lo demás es secundario. Esto es algo subjetivo y, por tanto, el éxito depende en gran medida de la habilidad del individuo que lleva a cabo el análisis. Por otro lado, el analista técnico además de hacer uso de los gráficos y su correspondiente parte subjetiva, los cuantifica, los prueba y los optimiza con el propósito de desarrollar sistemas mecánicos de transacción [3]. Se hace uso de fórmulas matemáticas y estadísticas con el fin último de conseguir generar una simple señal que indique si comprar, vender o mantenerse. Con esto se pretende evitar la subjetividad y la divergencia que ocasiona el análisis chartista [4].

Indicadores

El análisis técnico se basa en el uso de indicadores técnicos los cuales se fundamentan en un conjunto de fórmulas matemáticas sobre las series históricas de precios de mercado. Con ayuda de estos indicadores, se construyen unas señales de entrada y salida del mercado en el tiempo, recomendando al inversor cuando y como es mejor invertir.

Dentro del análisis técnico existen un gran número de indicadores diferentes que se pueden clasificar en cuatro grupos diferentes. A continuación, se hablará sobre cada uno de estos grupos y se hablará en detalle de alguno de los indicadores más importantes y utilizados según la literatura y que serán utilizados en la aplicación a desarrollar.

2.1.1.1 Indicadores de momentum

Este grupo de indicadores técnicos mide la aceleración y desaceleración en los cambios de precios en un periodo de tiempo fijado respecto al precio de cierre actual. Se obtiene restando al cierre de hoy el cierre de hace N días, donde N es el periodo del momento. Dependiendo de si queremos trabajar a corto-medio plazo o a medio-largo plazo, se elijará un momento más lento o rápido, es decir una n mayor o menor [4] [5].

La existencia del momentum se debe a una anomalía del mercado que la teoría financiera trata de explicar. Se habla de anomalía ya que, el hecho de que un activo vea incrementado su precio en algún momento no debería significar, con cierto grado de certeza, una subida posterior. Según la hipótesis de eficiencia del mercado, este incremento de precio debería estar sólo justificado por cambios en la oferta y en la demanda o por nueva información que se agrega al mercado. Sin embargo, los

expertos financieros creen que la existencia de esta anomalía se debe al sesgo cognitivo producido por el comportamiento irracional de los inversores [6].

Interpretación

De forma simplificada y general, el momento se representa en un gráfico que se mueve en torno a una línea central. Si el momento está por encima de la línea central, la tendencia al alza de los precios se está acelerando y será más acelerada cuanto más arriba este de la línea central. En otras palabras, los precios del cierre de hoy son superiores a los del cierre de hace N días. Un momento ascendente que cruza la línea central es señal de compra [5].

De igual forma sucede para el caso contrario. Si el momento está por debajo de la línea central, la tendencia a la baja se está acelerando en mayor o menor medida dependiendo de su distancia respecto a la línea central. Los precios de cierre de hoy son inferiores a los de hace N días. Un momento descendente que cruza la línea central es señal de venta [5].

Cuando el momento se aplana o regresa en dirección a la línea central, independientemente de si es un momento positivo o negativo, se produce una desaceleración y puede significar un cambio de tendencia [5].

En los valores extremos del momento, se produce una señal de compra si se llega a un valor del extremo inferior del momento, es decir, si encontramos un estado de sobreventa. De igual forma, se produce una señal de venta si se llega a un valor del extremo superior del momento, es decir, si encontramos un estado de sobrecompra [5].

[2.1.1.1.1 RSI](#)

Definición

El RSI o índice de Fuerza Relativa es un indicador de momento, desarrollado por J. Wells Wilder, que mide las variaciones de los precios en un periodo de tiempo determinado [4]. Su representación es lineal y está normalizado ya que se mueve entre valores de 0 a 100. Normalmente este rango de 0 a 100 se divide mediante 3 líneas horizontales situadas en los valores 30, 50 y 70 [6]. Esta división es muy común y permite representar diferentes estados o señales como se muestra en la siguiente imagen:

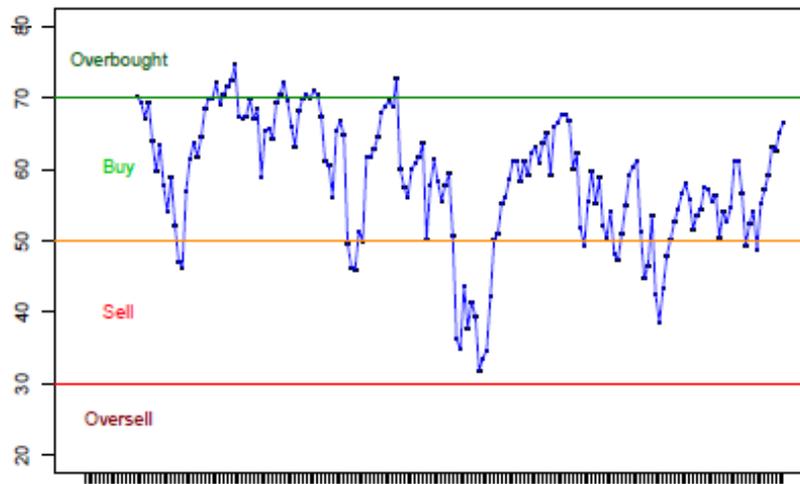


Ilustración 1 - RSI zonas de tendencia

El indicador se define matemáticamente mediante la siguiente expresión:

$$RSI = 100 - \frac{100}{1 + \left(\frac{Gm}{Pm}\right)}$$

Siendo Gm la ganancia media en periodos de subida y Pm la pérdida media en periodos de bajada [6].

Interpretación

- Un valor igual a 50 indica ausencia de tendencia.
- Un valor superior a 50 indica que las ganancias son mayores que las pérdidas.
- Un valor inferior a 50 indica que las pérdidas son mayores que las ganancias.
- Un valor entre 0 y 30 indica sobreventa, posibilidad de compra cuando se traspasa de forma ascendente este límite.
- Un valor entre 30 y 50 indica venta.
- Un valor entre 50 y 70 indica compra.
- Un valor entre 70 y 100 indica sobrecompra, posibilidad de venta cuando se traspasa de forma descendente este límite.

Su utilidad fundamental es como indicador de sobrecompra o sobreventa de un valor [4].

2.1.1.1.2 Oscilador Estocástico

El oscilador estocástico es un indicador de momento desarrollado por George C. Lane y tiene las siguientes características básicas [6]:

- Es capaz de capturar la velocidad el movimiento de un precio.
- Muestra la localización de un precio de cierre con respecto al máximo y al mínimo producidos en una venta temporal.

- Normalmente, el *momentum* cambia antes que el precio por lo que puede usarse para anticipar cambios de tendencia.
- El gráfico se compone de dos líneas que se mueven entre un valor de 0 y 100. Denotadas típicamente como K y D.
- Tiene dos líneas horizontales que indican la sobrecompra o sobreventa del activo. Normalmente estas líneas se sitúan en valores de 20 para marcar la sobreventa y de 80 para marcar la sobrecompra.

El indicador se define matemáticamente de la siguiente forma:

K se obtiene para un día **n** perteneciente a la ventana temporal **nK** como:

$$K = 100 * \left(\frac{\text{Cierre} - \text{min}}{\text{max} - \text{min}} \right)$$

Dónde:

- **Cierre**: es el precio de cierre del activo
- **min**: es el precio mínimo que ha habido dentro de la ventana temporal **nK**, es decir, en los **nK** días anteriores.
- **max**: es el precio máximo que ha habido dentro de la ventana temporal **nK**, es decir, en los **nK** días anteriores.
- **K**: es la posición del precio de cierre del día considerado, con respecto al rango total (**min,max**) en la ventana **nK**, expresada como porcentaje.

D se obtiene a partir de **K** de la siguiente forma:

$$\text{temp} = \text{SMA}(K, N)$$

$$D = \text{SMA}(\text{temp}, M)$$

Dónde:

- **N**: ventana temporal de **N** días.
- **SMA**: Media móvil simple. Este indicador se explicará más adelante ya que pertenece a otra categoría.
- **temp**: es el resultado de la media móvil de la serie **K** con una ventana temporal de **N** días.
- **M**: ventana temporal de **M** días.
- **D**: el resultado de la media móvil simple de la serie **temp** con una ventana temporal de **M** días.

Interpretación

El oscilador estocástico K se encarga de situar el precio actual con respecto al máximo y al mínimo en un periodo anterior de tiempo. Como se ha comentado, K es un porcentaje y, por tanto, su valor estará comprendido entre 0 y 100. Valores por encima de 50 indican que el precio es mayor que el valor medio entre el máximo y el mínimo,

por debajo de 50 indican que el precio es menor. Un valor de 100 indica que el precio actual es el máximo del periodo y un valor del 0 indica que es el mínimo [6].

Por otro lado, D, se construye mediante dos medias móviles sobre K con el objetivo de suavizar K. De esta forma tenemos una señal, K, que genera más señales ya que se mueve más y una señal, D, que está suavizada y genera menos señales ya que se mueve menos [6].

Si K cruza su media móvil D en sentido ascendente, significa que la tendencia del precio es más pronunciada hoy que en pasados días. Si K cruza D en sentido descendente, significa que la tendencia es menos pronunciada que en pasados días [6].

Se considera que, por encima de un nivel de 80 para K y D, el activo está sobrecomprado, por lo que se espera que su precio baje. Por debajo de 20 el activo está sobrevendido por lo que se espera que el precio suba. Con toda la información recogida anteriormente podemos diseñar las siguientes señales [6].

En base a esta información, se puede afirmar que:

- Cuando K cruza D hacia arriba y ambas líneas están por debajo de 20, el indicador sugiere la compra del activo.
- Cuando K cruza D hacia abajo y ambas líneas están por encima de 80, el indicador sugiere la venta del activo.

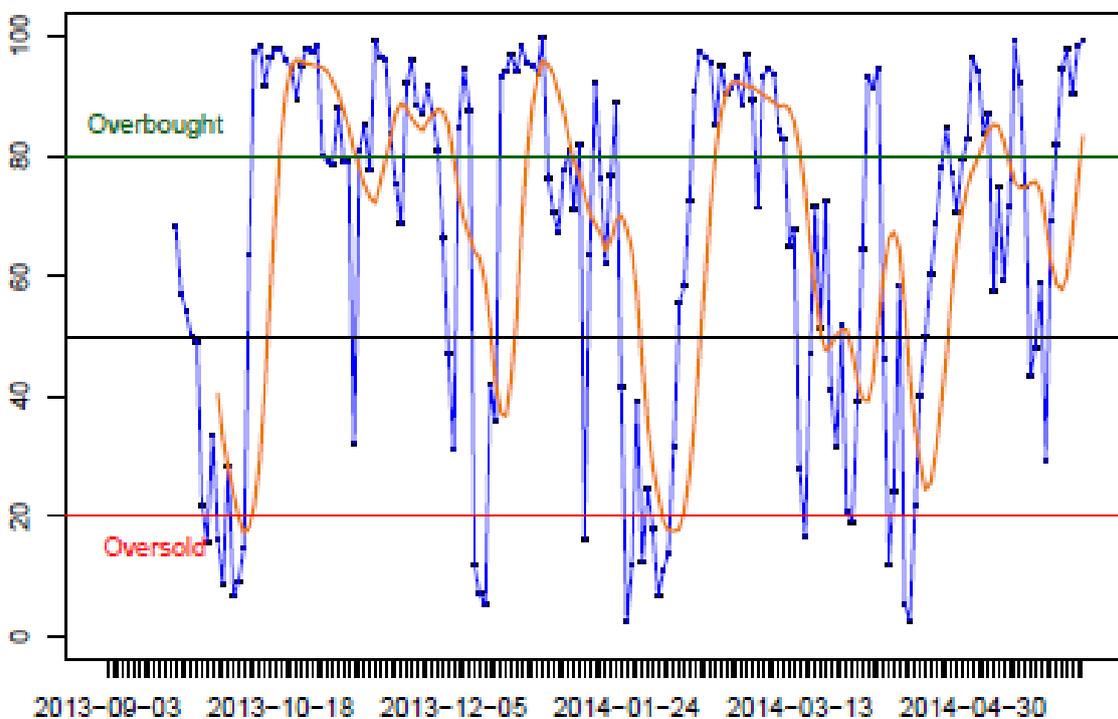


Ilustración 2 - Ejemplo del oscilador estocástico. En azul K, en naranja D.

2.1.1.2 Indicadores de tendencia

Definición

Una tendencia se produce cuando los precios suben o bajan durante un determinado periodo de tiempo. La tendencia será alcista cuando en sucesivos periodos el precio de un activo alcanza un máximo más alto que el precedente. La tendencia será bajista si en sucesivos periodos la bajada desciende hasta un mínimo más bajo que el de la bajada precedente. La tendencia será lateral (también conocido como mercado lateral) cuando el precio de un activo se mueve lateralmente, sin dirección [7].

Atendiendo a la anterior definición de tendencia bursátil, un indicador de tendencia no es más un indicador para saber la dirección del mercado o dirección de la tendencia.

2.1.1.2.1 Medias móviles

Definición

Una media móvil es un indicador técnico de tendencia que muestra el valor medio de los datos comprendidos en su ventana temporal. Por tanto, la media móvil depende de dos factores, los valores que se están promediando y el horizonte temporal elegido [7].

La media móvil es el indicador más simple y utilizado en el análisis técnico y típicamente sirven como base de cálculo para otros indicadores. Normalmente se calcula haciendo uso de los precios de cierre de mercado [7].

Una media móvil no anticipa o predice nada ya que es un indicador de seguimiento, es decir, primero se produce el cambio de tendencia y posteriormente la media móvil reacciona y lo confirma [4].

Media Móvil Simple

Como su nombre indica es la media móvil más sencilla de calcular y por ello una de las más utilizadas aunque no por ello es la mejor. La media móvil simple es aquella que asigna los mismos pesos a todos los precios de la ventana temporal. Se calcula de la siguiente forma [6]:

$$SMA = \frac{P1 + P2 + \dots + PN}{N}$$

Dónde:

- P: son los distintos precios de cierre dentro de la ventana temporal.
- N: es el tamaño de la ventana temporal, normalmente representa los días.

Esta media móvil tiene un defecto y es que cambia dos veces cuando entra un dato nuevo en la serie ya que cambia al insertar un nuevo precio y al extraer el último precio que había en la serie y que es reemplazado por el nuevo. Este problema se soluciona haciendo uso de las medias móviles exponenciales [7].

Media Móvil Exponencial

La media móvil exponencial es una media móvil en la que se asignan diferentes pesos a los distintos valores de la ventana temporal, estos pesos decrecen exponencialmente desde el valor más reciente al más antiguo. Esta media móvil mejora la media móvil simple en tres aspectos [7]:

- Da más peso a los datos recientes.
- Responde a los cambios más rápidamente.
- No cambia dos veces al introducir un dato nuevo en la serie. En la media móvil simple el último precio se extraía de golpe y tenía el mismo peso en el cálculo que todos los demás precios. Sin embargo, en la media móvil exponencial el dato que se extrae del cálculo en reemplazo del nuevo es el que menos peso tiene en el cálculo (por ser el más antiguo) y por ello su salida afecta en muchísima menor medida al cálculo que como lo hacía en la media móvil simple.

La media móvil exponencial se calcula de la siguiente forma [6]:

$$EMA_i = (1 - \alpha) * EMA_{i-1} + \alpha * p_i$$

Dónde:

- P: es el precio de cierre.
- α : típicamente en análisis técnico, $\alpha = \frac{2}{n+1}$, siendo n el tamaño de la ventana temporal.
- El primer valor de la serie EMA se asigna como: $EMA_n = SMA_n$.

Sustituyendo, la fórmula final quedaría de la siguiente forma:

$$EMA_i = \frac{n-1}{n+1} * EMA_{i-1} + \frac{2}{n+1} * p_i$$

Interpretación

Lo más importante de una media móvil es la dirección de su pendiente, si sube muestra el optimismo de los inversores (compradores), si baja el pesimismo de los inversores (vendedores) [6].

2.1.1.2.2 MACD

Definición

El MACD o Convergencia/Divergencia de Medias Móviles es un indicador técnico de tendencia creado por Gerald Appel a final de los años 70 que proporciona información sobre cómo varía el precio a lo largo del tiempo. El indicador se compone de tres series temporales e informa sobre la tendencia de un activo, señalado su fuerza, dirección y duración [7].

La primera serie temporal se denomina igual que el indicador, MACD, y se construye como la diferencia de dos medias móviles exponenciales con distintos periodos. Representa una estimación de la primera derivada del precio respecto al tiempo [6].

$$MACD = EMA(p, 12) - EMA(p, 26)$$

Dónde:

- p : es la serie temporal del precio.
- 12 y 26 son los periodos por defecto.

La segunda serie temporal se llama señal y se construye como la media móvil exponencial sobre la serie MACD con un periodo por defecto de 9 días [6].

$$señal = EMA(MACD, 9)$$

La última serie temporal se llama divergencia y es la diferencia entre la serie MACD y la señal. Se representa mediante un histograma. Representa una estimación de la segunda derivada del precio respecto al tiempo [6].

$$divergencia = MACD - señal$$

Interpretación

Las señales que proporciona el MACD vienen determinadas por la serie temporal de la divergencia y en menor medida por la serie temporal MACD. Los cambios de signo en la línea de la divergencia indican potenciales tendencias alcistas o bajistas de la siguiente forma [6]:

- Cuando la divergencia pasa de ser negativa a ser positiva o la línea MACD cruza la señal de abajo a arriba, se genera una señal de compra. Se produce una aceleración de precios y se da una potencial tendencia alcista.
- Cuando la divergencia pasa de ser positiva a ser negativa o la línea MACD cruza la señal de arriba abajo se genera señal de venta. Se produce una desaceleración en el precio y se da una potencial tendencia bajista.

Por su parte, la serie temporal MACD es capaz de ofrecer información por sí misma, aunque de menor relevancia [6]:

- Cuando la línea MACD pasa de negativo a positivo, se genera señal de compra.
- Cuando la línea MACD pasa de positivo a negativo, se genera señal de venta.

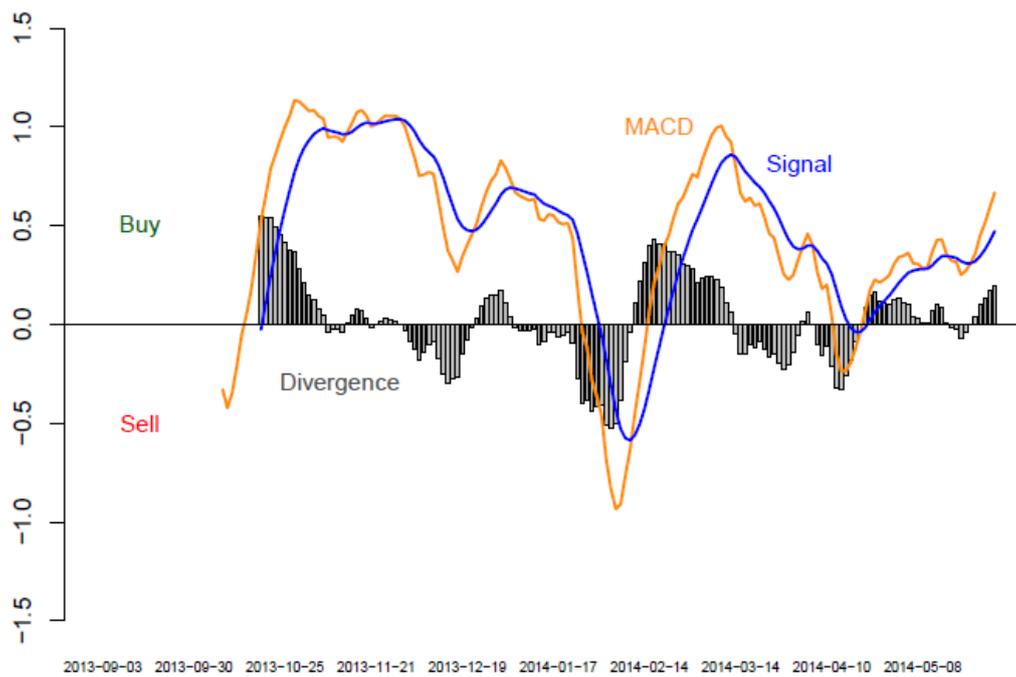


Ilustración 3 - Ejemplo MACD

2.1.1.3 Indicadores de volatilidad

La volatilidad es la medida de la frecuencia e intensidad de los cambios en un precio. Cuando la volatilidad es muy alta, los precios varían mucho en el tiempo, mientras que, si es baja, los precios son más estables [6].

Un indicador de volatilidad es un indicador que permite medir la volatilidad de una serie de precios para tomar decisiones. Una mayor volatilidad se traduce en un mayor riesgo de pérdida, pero en caso de ganancia, se espera un mayor retorno. Menor volatilidad implica menos riesgo, pero menor retorno positivo. Este tipo de indicadores no se bastan por si mismos para realizar operaciones de compraventa y suelen utilizarse en conjunto con indicadores de tendencia y de momento [6].

2.1.1.3.2 Bandas de Bollinger

Definición

Las Bandas de Bollinger es un indicador técnico de volatilidad, desarrollado por John Bollinger en los años 80, que es capaz de medir la volatilidad del precio del activo. El indicador se compone de tres líneas diferentes:

- Línea media: se define como la media móvil simple del precio diario calculada sobre un periodo de n días (típicamente n = 20). El precio diario (**PM**) será igual a la media entre el máximo, mínimo y el precio de cierre:

$$PM = \frac{\text{max} + \text{min} + \text{cierre}}{3}$$

$$\text{Línea Media} = SMA(PM, n)$$

- Las líneas superior e inferior se calculan como un múltiplo, m, de la desviación estándar del precio de n días sobre la media móvil anteriormente calculada.

$$\text{Línea superior} = \text{Línea Media} + m * \sigma(PM, n)$$

$$\text{Línea inferior} = \text{Línea Medi} - m * \sigma(PM, n)$$

Interpretación

El precio del activo generalmente oscilará entre la banda superior e inferior, pero cuando se cruzan estas líneas es cuando obtenemos las señales que proporciona el indicador.

Cuando el precio alcanza la línea superior se genera una señal de venta ya que el mercado se encuentra en sobrecompra.

Cuando el precio alcanza la línea inferior se genera una señal de compra ya que el mercado se encuentra en sobreventa.

Por otro lado, la anchura creada entre la banda inferior y la banda superior también proporciona información, a mayor anchura, mayor volatilidad en los precios.

Toda esta información que nos ofrecen las Bandas de Bollinger se puede resumir en una sola ecuación:

$$B = \frac{PM - Inf}{Sup - Inf}$$

Dónde:

- PM: es la media móvil simple del precio diario.
- Sup: es el valor de la banda de Bollinger superior.
- Inf: es el valor de la banda de Bollinger inferior.

B representa la proporción de la anchura de banda donde se encuentra el precio.

- Si $B > 1$ indica que el precio está por encima de la banda superior y se genera señal de venta.
- Si $B < 0$ el precio está por debajo de la banda inferior y se genera señal de compra.

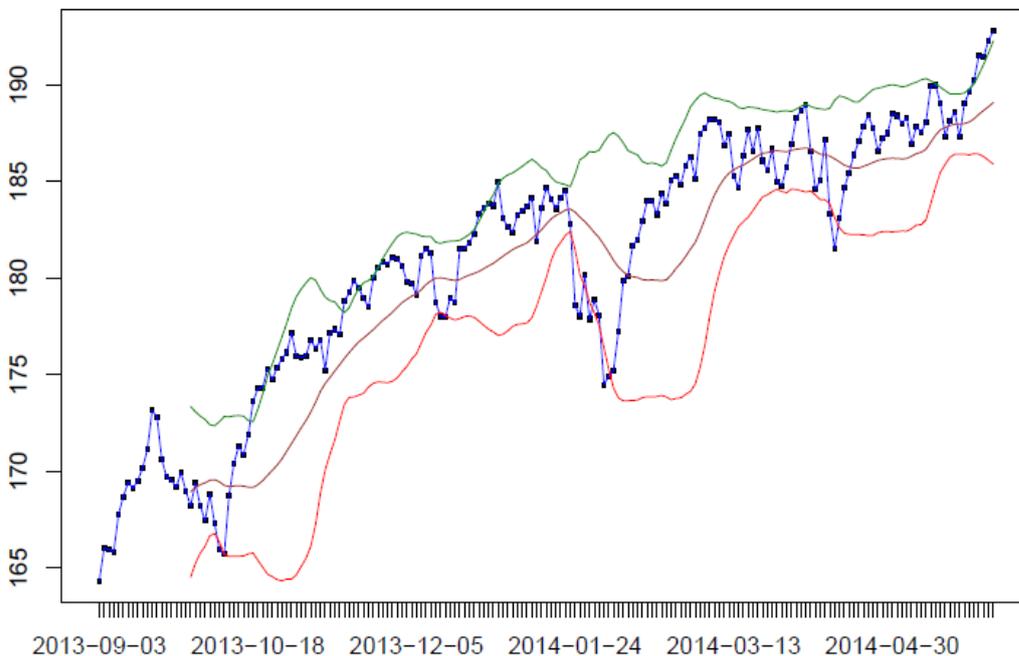


Ilustración 4 - Bandas de Bollinger (Verde = Banda superior, rojo = banda inferior, marrón = banda medi, azul = precio de cierre del activo)

2.1.1.4 Indicadores de volumen

El término volumen hace referencia a la cantidad de intercambios de un mismo activo que se ha realizado en un determinado periodo de tiempo. Normalmente, cuando el volumen es más alto se debe a que el precio de un activo está cambiando. Además, un gran volumen implica una gran liquidez en el mercado.

Los indicadores de volumen son aquellos indicadores que hacen uso de los datos referentes al volumen negociado de un activo para predecir y generar una señal. Los indicadores anteriormente descritos hacían uso siempre del precio de cierre, del precio máximo diario y del precio mínimo diario, sin embargo, estos hacen uso del volumen negociado.

El volumen representa la actividad de los inversores y traders en el mercado. Algunos inversores creen que los precios ya reflejan toda la información de los activos del mercado, otros entienden que el volumen es un elemento que les permite comprender los mercados más profundamente y operar mejor. Como es de utilidad para algunos inversores conviene en este proyecto introducirlos [7].

2.1.1.4.1 EMV

Definición

El indicador de facilidad de movimiento EMV es un indicador técnico de volumen desarrollado por Richard W. Arms Jr. Se trata de un oscilador basado en el volumen diseñado para medir la relación entre el precio y el volumen. El EMV se mueve alrededor de una línea central situada en cero para cuantificar la facilidad de los movimientos del precio. Cuando el EMV es positivo, los precios tienen facilidad para subir y cuando el EMV es negativo los precios tienen facilidad para bajar.

El cálculo del indicador se realiza de la siguiente forma:

Cálculo del punto medio diario como la media entre el máximo y mínimo diario:

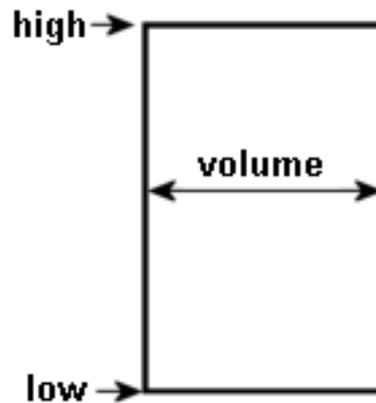
$$\mathbf{midpoint} = \frac{\mathbf{max} + \mathbf{min}}{2}$$

Con este valor se calcula la variación de este midpoint entre dos días consecutivos (momentum del midpoint):

$$\mathbf{var_midpoint} = \mathbf{midpoint}_t - \mathbf{midpoint}_{t-1}$$

A continuación, se calcula el cociente entre la anchura y la altura del *equivolume box*, que es el cociente entre el volumen diario (normalizado para hacer el indicador más legible) y la diferencia entre el máximo y el mínimo diario:

$$bratio = \frac{\text{volumen}}{\frac{\text{vol. divisor}}{\text{max} - \text{min}}}$$



Equivolume Box

Ilustración 5 - Equivolume box http://stockcharts.com/school/doku.php?id=chart_school:chart_analysis:equivolume

Finalmente, el indicador EMV se calcula como el cociente entre *var_midpoint* y *bratio*:

$$EMV = \frac{\text{var_midpoint}}{\text{bratio}}$$

Para facilitar la generación de señales, este indicador se suele suavizar haciendo uso de una media móvil de la siguiente forma:

$$maEMV = SMA(EMV, n)$$

Con $n = 9$, típicamente.

Interpretación

Como se ha descrito anteriormente, el EMV se construye como el cociente entre el momentum del precio medio y el boxratio. Por tanto, el EMV será positivo cuando el precio medio suba y negativo cuando el precio medio baje. Además, los valores del EMV serán altos cuando la subida o bajada de precio sea muy fuerte y/o cuando el boxratio sea pequeño, es decir, cuando el volumen es pequeño o la diferencia de precios entre el máximo y el mínimo es grande.

Gracias a la suavización del EMV mediante una media móvil simple podemos generar dos señales de compra y venta:

- Cuando *maEMV* cruza hacia los valores positivos se genera señal de compra.
- Cuando *maEMV* cruza hacia los valores negativos se genera señal de venta.

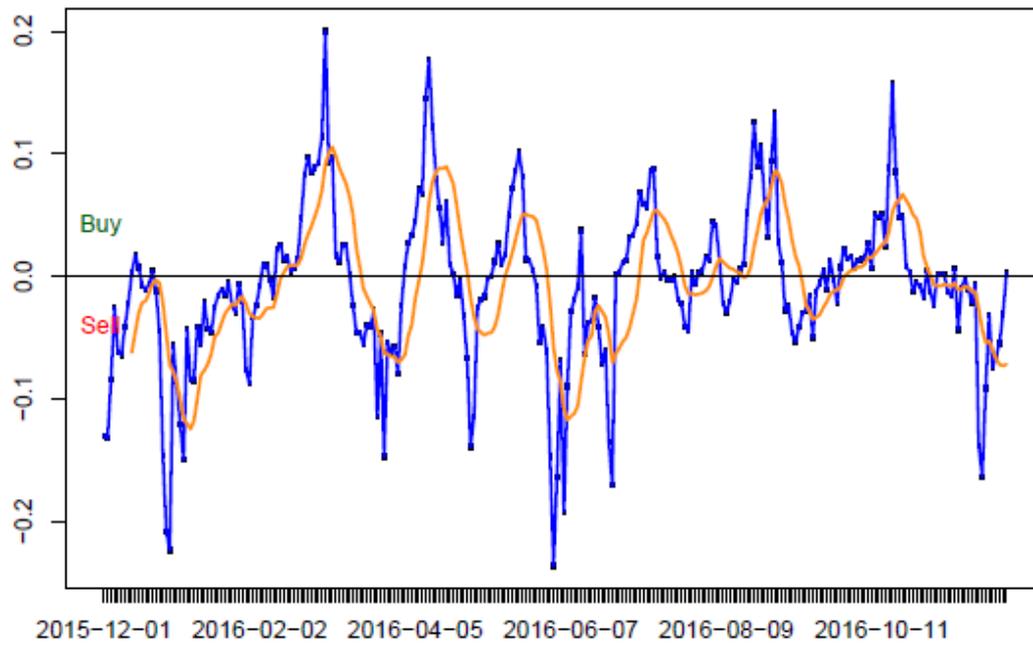


Ilustración 6 - EMV(azul) y maEMV(naranja)

2.1.2 Análisis fundamental

Definición

El análisis fundamental se basa en examinar todos aquellos factores relevantes que afectan al precio de una acción en el mercado para determinar su valor intrínseco, esto es, el valor real de la empresa en cuestión según la ley de la oferta y la demanda. Para ello se hace uso de datos como pueden ser sus cuentas de resultados, su contabilidad, sus ratios, productos o la competencia, entre otros.

El funcionamiento básico de este análisis es el siguiente: si el valor intrínseco de una empresa se encuentra por debajo del precio actual de mercado, quiere decir que el activo o empresa en cuestión está sobrevalorada y debe venderse. Por otro lado, si el valor intrínseco se encuentra por encima del precio actual de mercado, quiere decir que el activo está infravalorado y debería comprarse [3].

Algunos Indicadores

Los factores relevantes comentados anteriormente son consultados en el análisis fundamental haciendo uso de indicadores fundamentales. Algunos de estos indicadores son los siguientes [9]:

- PER: el ratio Precio-Beneficio expresa lo que paga el mercado por cada unidad monetaria de beneficio. Representa la valoración que realiza el mercado de la capacidad de generar beneficios de una empresa.
- RoI: el retorno sobre la inversión mide la eficiencia de una inversión o compara la eficiencia de diferentes inversiones.
- Price Cash-Flow: relaciona el precio de cotización en mercado de la compañía con los fondos generados por la misma. Mide en qué medida el mercado está valorando la evolución futura de la compañía en términos de flujos de caja.



2.1.3 Comparativa

El análisis técnico y el análisis fundamental son dos corrientes igual de válidas para el análisis bursátil de los mercados, pero como es observable en lo anteriormente descrito, son dos corrientes bien distintas. El análisis técnico sostiene que los precios ya reflejan todos los factores económicos subyacentes y cree que las respuestas de los inversores a los movimientos de los precios predicen nuevos movimientos. Por su parte, el análisis fundamental cree que los mercados pueden errar en el precio de un activo a corto plazo y que a largo plazo este precio será corregido [6].

Otra diferencia fundamental entre el análisis técnico y el análisis fundamental reside en la temporalidad de los datos utilizados. El análisis técnico se nutre de datos históricos para todos sus cálculos mientras que el análisis fundamental utiliza datos actuales y vigentes en la mayoría de sus cálculos, aunque también puede hacer uso de datos pasados.

La frecuencia de obtención de los datos utilizados también es muy distinta. Mientras que el análisis técnico tiene nuevos precios de cierre cada día, en el análisis fundamental, en el mejor de los casos, tendrá nuevos datos de las empresas cada trimestre y normalmente esta información se presentará con retraso.

Existe mucha literatura de ambos tipos de análisis y hay una gran cantidad de defensores y detractores de cada tipo de análisis. Normalmente el analista técnico es detractor del fundamental y el analista fundamental es detractor del técnico, pero no existe realmente nada que permita afirmar el buen hacer de uno sobre otro. En el análisis bursátil todo se basa en predecir, ya sea basándose en un tipo de análisis u otro.

2.2 Antecedentes en software de análisis técnico

Desde la década de los 90, se comenzó a desarrollar diversos sistemas automáticos que tratan de satisfacer la necesidad humana de interpretar los indicadores técnicos en busca de señales de entrada y salida del mercado [6].

Tal y como se mencionó al principio de este documento, los gestores de fondos hacen un gran uso del análisis técnico y fundamental para guiarse en sus inversiones. Existen multitud de herramientas que realizan los cálculos correspondientes a numerosos indicadores y permiten visualizar sus gráficas y resultados. A continuación, se describen dos ejemplos que ofrecen la utilidad del análisis técnico:

2.2.1 Finviz

Finviz es una aplicación web que permite realizar análisis técnico, análisis fundamental y visualizar el histórico y la cotización actual de activos cotizados en los mercados americanos. Tiene una modalidad de pago y una gratuita. En la modalidad gratuita en lo que al análisis técnico se refiere te permite visualizar gráficamente la cotización de un activo en diferentes frecuencias temporales además de varias medias móviles. Sin embargo, del resto de indicadores técnicos que implementa solo proporcionan el valor del indicador a día de hoy y la recomendación de inversión, sin posibilidad de visualizar gráficamente el mismo [12].

reener | publish chart | save to portfolio | create alert Type: candle | line | **advanced** | interactive | perf Timeframe: intraday | **daily** | weekly | mon



P/E	9.71	EPS (ttm)	4.78	Insider Own	0.70%	Shs Outstand	516.11M
Forward P/E	8.67	EPS next Y	5.35	Insider Trans	-13.93%	Shs Float	516.11M
PEG	4.45	EPS next Q	0.51	Inst Own	80.40%	Short Float	5.10%
P/S	0.60	EPS this Y	-56.50%	Inst Trans	1.09%	Short Ratio	4.15
P/B	6.30	EPS next Y	15.29%	ROA	5.30%	Target Price	54.27
P/C	3.76	EPS next 5Y	2.18%	ROE	62.20%	52W Range	24.66 - 50.53
P/FCF	42.05	EPS past 5Y	18.20%	ROI	13.00%	52W High	-8.25%
Quick Ratio	0.70	Sales past 5Y	10.90%	Gross Margin	35.60%	52W Low	88.01%
Current Ratio	0.70	Sales Q/Q	1.70%	Oper. Margin	13.20%	RSI (14)	49.25
Debt/Eq	6.43	EPS Q/Q	-89.10%	Profit Margin	6.70%	Rel Volume	0.77
LT Debt/Eq	5.94	Earnings	Jan 27 BMO	Payout	8.30%	Avg Volume	6.34M
SMA20	1.21%	SMA50	-1.16%	SMA200	18.55%	Volume	4,930,900

Ilustración 7 - Ejemplo de análisis técnico en la versión gratuita de Finviz [12]

En la modalidad de pago te permite visualizar todo de forma más detallada y gráficamente, así como la posibilidad de hacer backtesting. El coste de la utilidad es de 25\$ mensuales [12].

Esta aplicación web es la que más se asemeja a nuestra aplicación objetivo ya que cuenta con análisis técnico, visualización de las gráficas de los indicadores, recomendaciones de inversión y backtesting.

2.2.2 Investing

Investing es una aplicación web muy similar a Finviz que ofrece la posibilidad de obtener información de análisis técnico. A diferencia de Finviz, esta aplicación muestra cotizaciones de activos de mercados españoles y, por tanto, permite la realización de análisis técnico sobre activos de empresas negociadas en mercados españoles. En lo que al análisis técnico se refiere la aplicación ofrece 11 indicadores técnicos y un conjunto de medias móviles. La aplicación, igual que Finviz, muestra los resultados a día de hoy el indicador y la recomendación de inversión, pero no hay posibilidad alguna de realizar el estudio visual de los indicadores que ofrecen. Esta web es completamente gratuita [13].

Resumen: **COMPRA FUERTE**

Media móvil:	COMPRA	Compra (12)	Venta (0)
Indicadores Técnicos:	COMPRA FUERTE	Compra (6)	Venta (0)

Punto pivote » 01.03.2017 11:00 GMT

Nombre	S3	S2	S1	Puntos pivotes	R1	R2	R3
Clásico	5,222	5,232	5,252	5,262	5,282	5,292	5,312
Fibonacci	5,232	5,243	5,251	5,262	5,273	5,281	5,292
Camarilla	5,264	5,267	5,269	5,262	5,275	5,277	5,280
Woodies	5,226	5,234	5,256	5,264	5,286	5,294	5,316
DeMark's	-	-	5,257	5,264	5,287	-	-

Indicadores técnicos » 01.03.2017 11:00 GMT

Nombre	Valor	Acción
RSI(14)	80,776	Sobrecompra
STOCH(9,6)	75,253	Compra
STOCHRSI(14)	100,000	Sobrecompra
MACD(12,26)	0,039	Compra
ADX(14)	77,313	Sobrecompra
Williams %R	0,000	Sobrecompra
CCI(14)	172,7378	Compra
ATR(14)	0,0281	Mayor volatilidad
Highs/Lows(14)	0,1098	Compra
Ultimate Oscillator	78,435	Sobrecompra
ROC	4,169	Compra
Bull/Bear Power(13)	0,1830	Compra

Media móvil » 01.03.2017 11:00 GMT

Periodo	Simple	Exponencial
MA5	5,234 Compra	5,238 Compra
MA10	5,185 Compra	5,196 Compra
MA20	5,129 Compra	5,157 Compra
MA50	5,118 Compra	5,134 Compra
MA100	5,149 Compra	5,140 Compra
MA200	5,159 Compra	5,140 Compra

Compra: 12 Venta: 0
Resumen: **COMPRA**

Ilustración 8 - Ejemplo de análisis técnico en Investing [13]

2.3 Obtención de los datos

Los datos, en el mundo financiero, son de gran importancia, ya que todo se compone de datos y todos los procesos intervinientes desde el front-office hasta el back-office hacen uso de los datos financieros. En el caso particular de este proyecto se requieren una serie de datos muy concretos cuyos requisitos de obtención no son muy exigentes. Los datos necesarios para realizar el análisis técnico son los siguientes:

- Precio de cierre: Es el precio de cierre de mercado de un activo en un día específico.
- Precio máximo de un activo en el día.
- Precio mínimo de un activo en el día.
- Volumen: cantidad de operaciones realizadas sobre un activo en el día.
- Fecha a los que pertenece cada uno de estos datos.

Como se puede observar, el número de datos requeridos no es muy elevado y, además, la frecuencia de obtención de los mismos es diaria por lo que tampoco es una gran cantidad de información a guardar.

El hecho de utilizar los datos de los días anteriores en el análisis técnico hace que las exigencias tecnológicas a la hora de obtener los datos sean reducidas. A diferencia de otros sistemas como pueda ser un software de trading de alta frecuencia o un terminal de trading, que necesitan obtener en tiempo real todas las cotizaciones de uno varios mercados, en esta área específica de las finanzas no hay necesidad de realizar sistemas muy sofisticados (en lo que a obtención de datos se refiere) que hagan uso de tecnologías como pueda ser los sistemas de baja latencia. Simplemente se requiere obtener y almacenar los datos mencionados anteriormente al cierre de mercado, para que al día siguiente el gestor o inversor que quieran hacer uso del análisis técnico tenga los datos del día anterior listos para ser usados en sus indicadores.

Estos datos, aunque no sean muchos ni una gran cantidad deben ser obtenidos de algún lugar, como cualquier otro dato financiero. En el sector financiero, así como en muchos otros sectores, existe la figura del **data vendor**. Un *data vendor* es una empresa que se dedica a la aglutinación, procesamiento y provisión de datos, en este caso, datos financieros, como puedan ser las cotizaciones intra-día de los mercados, cotizaciones de cierre, características de todos los activos existentes, volumen negociado, etc. Estas empresas aglutinan toda la información y la venden a aquellas empresas o usuarios que quieran hacer uso de la misma. Algunas de las empresas más importantes del sector que se dedican a esto son las siguientes:

2.3.1 Bloomberg

Bloomberg es la alternativa más conocida y utilizada en el sector financiero no solo para la obtención de datos si no para la solución de otros muchos problemas del sector ya que es una compañía que ofrece muchas soluciones. Entre sus principales servicios están [10]:

- Terminal de trading: terminal profesional que cuenta con infinidad de información en tiempo real de los mercados, datos históricos, así como servicios para la realización de análisis técnico y fundamental entre otras cosas.
- Data vendor: precios en tiempo real de mercado, precios de cierre de mercado, definición y características de los activos financieros, valoración de productos complejos como los derivados, etc.
- Reporting regulatorio.
- Agregador de noticias del sector financiero.
- Radio con contenido del sector financiero.

2.3.2 Reuters

Reuters es la segunda compañía líder del sector, tras Bloomberg. Al igual que bloomberg no solo se dedica a la provisión de datos y también ofrece su propio terminal de trading, servicios post-trading, servicios para el manejo de riesgos o servicios regulatorios [11].

2.3.3 Yahoo Finance

Yahoo Finance es el departamento de Yahoo que se dedica al ámbito financiero. Cuentan con un portal web dedicado a las finanzas en el que incluyen noticias, datos de distintos mercados, cotizaciones en tiempo real e incluso algunos indicadores técnicos y fundamentales. Dentro de este departamento, Yahoo también ofrece una API para distintos lenguajes de programación, como pueden ser Java o R, para hacer uso de los datos de mercado que exponen en su portal web mediante programación software. El uso de esta API es gratuito, la obtención de sus datos también, aunque el uso de los datos está bastante restringido. Esta provisión gratuita de datos es el elemento diferencial entre Yahoo Finance y las anteriores compañías mencionadas, ya que nos permite probar nuestras aplicaciones sin tener que invertir una gran cantidad de dinero como ocurriría con las compañías anteriormente mencionadas [27].

2.3.4 Los mercados

Los propios mercados suelen actuar también como data vendors, ofreciendo distintos planes y precios para la obtención de los datos exclusivamente de ese mercado. Por ejemplo, Bolsas y Mercados Españoles ofrece suscripciones de periodicidad anual o mensual para la obtención de las cotizaciones de cierre de todos los mercados españoles. Estas alternativas son muy válidas por varias razones:

- La fiabilidad de los datos es máxima.
- El precio suele ser más económico que el de los aglutinadores anteriormente mencionados ya que no hay intermediarios que revenden.
- Si solo vas a trabajar con datos de los mercados de un país te ofrecen todo lo que está disponible a nivel nacional.

2.4 Backtesting

El backtesting en el área de las finanzas es el proceso de probar una estrategia de inversión haciendo uso de datos históricos, con el objetivo de comprobar si el modelo diseñado es útil y puede utilizarse a futuro en la negociación real. Este proceso es necesario, independientemente del activo con el que se quiera negociar o mercado en el que se quiera negociar ya que cuando hay dinero en juego toda precaución es poca [14].

El precio futuro de los valores seguirá siendo incierto, pero probar la estrategia en situaciones pasadas que fueron una realidad otorga al inversor cierto nivel de confianza al comprobar la robustez de su estrategia. Además, como se comentó anteriormente, el analista técnico confía en que lo que sucedió en el pasado puede suceder de nuevo en el futuro, y el backtesting tiene mucha relación con este concepto por lo que para los analistas técnicos el proceso de backtesting es un elemento importante.

2.5 Arquitectura SOA

SOA establece un marco de diseño para integrar aplicaciones independientes de forma que se puede acceder a sus funcionalidades a través de la red. Estas funcionalidades se ofrecen como servicios, los cuales implementan una funcionalidad concreta que puede ser descubierta en la red y describe lo que hace el servicio y como interactuar con él [15].

Estos servicios se implementan de forma habitual, aunque no siempre, mediante servicios web, una tecnología basada en diferentes estándares que es independiente de la plataforma. Algunos de estos estándares son: XML para representación de datos, SOAP para el intercambio de datos y el lenguaje WSDL para describir la funcionalidad del servicio web. Con los servicios web, SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar la misma funcionalidad, pero modularizada [15].

Algunos de los beneficios más importantes que ofrece la arquitectura SOA son los siguientes [15]:

- Aplicaciones productivas y flexibles: los servicios que se implementan se pueden comunicar con cualquier aplicación independientemente de la plataforma y el lenguaje de programación, es decir, son independientes de la infraestructura subyacente.
- Desarrollo de aplicaciones más rápido y económico: permite la creación de un repositorio de servicios reutilizables que se pueden utilizar e integrar en distintas aplicaciones eliminándose redundancia de código.
- Aplicaciones más seguras y manejables: SOA facilita el hecho de añadir o modificar servicios ya que son independientes de las aplicaciones estas no tienen por qué verse afectadas en las modificaciones. Además, cada servicio es independiente y no interfieren entre ellos.
- Capacidad para responder más rápidamente a los cambios.

2.6 Sistemas de soporte a la decisión

Definición

Existen numerosas definiciones para determinar lo que es un sistema de soporte a la decisión, ya que es un área muy extensa. No existe una definición universal, aunque si un par de definiciones clásicas que sirven para saber la base sobre la que se cimientan los sistemas de soporte a la decisión.

La primera definición de sistema de soporte a la decisión fue definida por Gorry y Scott-Morton en 1971 y dice que son “sistemas informáticos interactivos que ayudan a los tomadores de decisiones, utilizando modelos y datos, a resolver problemas no estructurados”. La segunda, definida por Keen y Scott-Morton en 1978 dice lo siguiente: “Los sistemas de apoyo a la toma de decisiones emparejan los recursos intelectuales de las personas con las capacidades de los ordenadores para mejorar la calidad de las decisiones. Se trata de un sistema de apoyo informático para los tomadores de decisiones gerenciales que tratan con problemas semi-estructurados”.

En la actualidad, se podría decir que un sistema de soporte a la decisión es un sistema informático que ayuda a las personas a tomar decisiones a la hora de afrontar un problema determinado. Estos sistemas, típicamente, se componen de una base de datos en la que se almacenan los datos de los que hará uso el sistema, de un modelo que haga uso de los datos y una interfaz gráfica con la que el usuario pueda interactuar y observar de forma gráfica los datos y elementos del problema a tratar.

Proceso de decisión

El proceso de decisión es un proceso sistemático por el cuál toda persona pasa a la hora de decidir, de forma consciente o inconsciente. Este proceso fue documentado en 1977 por Simon y consta de 3 etapas, posteriormente se añadió una cuarta.

El proceso comienza con la fase de inteligencia, en la que el tomador de decisiones analiza el entorno para identificar y definir el problema. En la segunda fase, la fase de diseño, se construye un modelo del problema y se analizan distintas alternativas de acción. La tercera fase es la fase de selección en la que se trata de seleccionar una solución para el modelo. En esta fase se toma la decisión, se elige el curso a seguir y se resuelve el modelo, no el problema que representa, ya que este se resolverá en la implementación de la solución. La implementación es la cuarta fase del proceso, la fase en la que se lleva a la práctica la decisión tomada para solucionar el problema real.

Este proceso que hacemos las personas de forma consciente o inconsciente lo incorporan los sistemas de soporte a la decisión, sistemas que se crearán para decidir sobre un problema, que implementan un modelo y que ayudan a seleccionar la solución al problema.



2.6.1 Trading algorítmico

El trading algorítmico se puede definir como una forma de hacer trading financiero haciendo uso de algoritmos en los que se prescinde de la intervención humana para mandar las órdenes, es decir, se hace uso de programas que compran y venden en el mercado de forma autónoma. Cada vez es más frecuente ver sistemas de este tipo, sobre los que se ejecutan complejos algoritmos de forma autónoma en computadores, para ello, se hace uso de muchas técnicas que permiten reducir la latencia de actuación y maximizar la velocidad del proceso. Estos sistemas son muy complejos de programar y están fuertemente regulados ya que un mal funcionamiento de los mismos puede tener graves consecuencias en los mercados.

A diferencia de un sistema de soporte a la decisión como el que se plantea en este proyecto, estos sistemas ejecutan de forma autónoma sin intervención humana, mientras que, en nuestro sistema de soporte a la decisión, hay elementos que configura el propio usuario y la decisión final de realizar o no una operación reside exclusivamente en el usuario, que debe realizar de forma manual y voluntaria la acción.

3. Análisis del sistema

3.1 Detalle de la aplicación

Como se comentó anteriormente, el objetivo de esta aplicación es el de proveer a los gestores de fondos de una herramienta de soporte a la decisión basándose en el análisis técnico financiero.

La aplicación formará parte de una aplicación existente que trata todo el flujo de una operación desde el front-office hasta el back-office, es decir se añadirá como un módulo nuevo a una aplicación existente. Esta aplicación existente se encarga de realizar ciertos procesos básicos que hacen que nuestra aplicación no tenga que implementarlos y, por tanto, no se detallarán.

El funcionamiento básico con el que contará la herramienta será el siguiente:

- Cálculo de indicadores técnicos, se incluirán como punto de partida aquellos descritos en el apartado 2 de este documento.
- Creación de gráficos para la representación de los indicadores.
- Obtención y almacenamiento de cotizaciones diarias.
- Creación y configuración de estrategias.
- Backtesting de estrategias.
- Recomendaciones de compraventa en función de una estrategia.
- Mandar las órdenes de una recomendación a mercado a través de un bróker.
- Configuración de las órdenes generadas tras una recomendación.
- Visualización de los resultados obtenidos con las inversiones.
- Interfaz gráfica que permita utilizar la herramienta y ver la información.
- Autoajuste opcional de los pesos de las estrategias.

Tan importante como definir lo que hace la aplicación es definir lo que no hace la aplicación. Como se ha comentado anteriormente esta aplicación es un módulo que se integrará en otra aplicación existente, la cual se ocupa del front-office, y por ello muchas de las tareas que hay implícitas en varios procesos están resueltas.

Nuestra aplicación abarca desde la obtención y almacenamiento de cotizaciones hasta el envío de órdenes, pasando por el proceso de análisis técnico. El envío de órdenes es algo muy complejo y que está implementado en la aplicación de front-office, es decir, nuestra aplicación no implementará nada respecto al proceso de envío de órdenes si no que usará los elementos existentes. Por tanto, no se implementarán elementos como comprobaciones de límites o comprobaciones de riesgo ya que todo eso existe.

3.1.1 Recomendaciones de compraventa

Los indicadores implementados por la herramienta son muy diversos de forma que habrá algunos con capacidad de generar señales de compraventa y otros que no. Aquellos que, si sean capaces de generar una señal, devolverán uno de los siguientes tres posibles resultados:

- Compra: indica sugerencia de compra del activo para el que se ha realizado el cálculo del indicador.
- Venta: indica una sugerencia de venta del activo para el que se ha realizado el cálculo del indicador.
- Neutro: indica una situación de no realizar ninguna acción con el activo para el que se ha realizado el cálculo del indicador.

Cuando se genera un resultado de compraventa para un activo, a su vez se genera una opción para enviar una orden de mercado al bróker en el mismo sentido de la recomendación (si es compra, de compra y si es venta de venta). Esta opción podrá ser ignorada o ejecutada, así como previamente configurable por el usuario.

En el caso de recibir una recomendación de neutro, la aplicación no permitirá enviar órdenes de ese activo ya que no se está siguiendo lo recomendado por la aplicación. Si quisiera realizar órdenes sobre un activo cuya recomendación es neutra o quiere realizar una orden en sentido opuesto a la recomendación, el usuario tiene la aplicación de front-end existente para hacerlo.

3.1.2 Estrategias

Las estrategias son conjuntos de indicadores (capaces de generar señales de compraventa) ponderados de forma que la suma de los pesos asignados a cada indicador que compone la estrategia sea 1. Cada indicador de forma independiente devolverá uno de los tres posibles resultados descritos en el punto anterior:

- 1, que se corresponde con la acción de vender.
- 2, que se corresponde con un estado neutro de no realizar ninguna acción.
- 3, que se corresponde con la acción de comprar.

En base a ese resultado y la ponderación del indicador se obtendrá un resultado final en conjunto para la estrategia siguiendo unas reglas. El resultado final de una estrategia puede tomar los valores de compra, venta o neutro.

A continuación, se muestra un ejemplo de cálculo de una estrategia compuesta por dos indicadores siguiendo una posible regla de entre muchas que se podrían aplicar:

El indicador 1 nos da una señal de venta y tiene un peso de 0,5 y el indicador 2 nos da una señal de compra y tiene un peso de 0,5:

$$ResInd1 = 0,5 * 1 = 0,5$$

$$ResInd2 = 0,5 * 3 = 1,5$$

Posteriormente se sumarán todos los valores obtenidos y se obtendrá un valor comprendido entre 1 y 3. Siguiendo el ejemplo:

$$ResEstrat = 0,5 + 1,5 = 2$$

Este conjunto de valores que van desde 1 hasta 3 se divide en tres partes iguales de forma que se dé a cada posible acción las mismas posibilidades:

Si $ResEstrat \geq 1$ y $ResEstrat < 1,666$: el resultado global de la estrategia será "VENDER".

Si $ResEstrat \geq 1,666$ y $ResEstrat < 2,333$: el resultado global de la estrategia será "NEUTRO".

Si $ResEstrat \geq 2,333$ y $ResEstrat \leq 3$: el resultado global de la estrategia será "COMPRAR".

Estas estrategias serán totalmente configurables por el usuario. El usuario podrá crear la estrategia con el conjunto de indicadores que desee (de entre aquellos que devuelvan un resultado explícito de compraventa) y asignarle los pesos que crea conveniente.

3.1.3 Backtesting

El backtesting es un elemento fundamental en el proceso de las inversiones financieras y siempre que se elabore una estrategia es necesario hacer backtesting para probar el funcionamiento de la estrategia diseñada. Se proporciona en la aplicación, una herramienta para hacer el backtesting de una estrategia creada por el inversor en la que para la recomendación de una fecha inicial se verá si en una fecha final se habrá ganado dinero o no.

3.1.4 Autoajuste

Se pretende añadir al sistema una funcionalidad de autoajuste de estrategias. Cuando un usuario se cree una estrategia tendrá la opción de que esta sea autoajutable, de forma que se crearán dos estrategias, la original del usuario, la cual está totalmente controlada por él y la autoajutable la cual es propiedad del sistema y no es modificable por el usuario, aunque sí usable para recibir recomendaciones.

Estas estrategias autoajustables irán cambiando sus pesos de forma dinámica con el tiempo, en función de las recomendaciones que haya generado y si después realmente acertó o no. Lo que se pretende, de forma resumida, es dar más peso a aquellos indicadores de la estrategia que acertaron en la predicción y quitarle peso, en favor de los que acertaron, a aquellos indicadores que fallaron.

Cada vez que se realice una recomendación con la estrategia original, se realizará una recomendación con la autoajustada y viceversa, de forma que siempre puedan ser comparadas posteriormente. Además, el usuario también puede realizar recomendaciones con las estrategias autoajustadas. Cada vez que se genere una recomendación por cualquiera de las dos vías se realizará un autoajuste en un periodo de tiempo determinado por el usuario a la hora de crear la estrategia. De esta forma se puede dotar a la estrategia autoajustada del horizonte temporal que el gestor desea emplear en sus inversiones.

Esta funcionalidad es algo totalmente experimental que puede o no funcionar y que se puede implementar de múltiples maneras, obteniendo comportamientos diferentes, aunque el objetivo siempre es el descrito anteriormente. Para ver la evolución de estas estrategias, el usuario tendrá un visor de resultados en el que podrá comparar las recomendaciones que generan las estrategias originales con las autoajustadas y comprobar cuál está funcionando mejor.

3.2 Requisitos de Usuario

Los requisitos de usuario son la descripción de lo que el cliente quiere o necesita de un software. Estos requisitos se obtienen, normalmente, mediante diversas reuniones con el cliente. En este caso el cliente es nuestro tutor. La recogida de estos requisitos se ha llevado a cabo mediante diferentes reuniones con el tutor.

Vamos a distinguir tres tipos de requisitos de usuario:

- **Requisitos de capacidad:** Son aquellos que representan lo que los usuarios necesitan para llevar a cabo un objetivo o resolver un problema.
- **Requisitos de restricción:** Son aquellas restricciones impuestas por los usuarios sobre cómo llevar a cabo un objetivo o cómo resolver un problema.
- **Requisitos inversos:** Se encargan de describir lo que el sistema no hará. Útiles especialmente para evitar confusiones.

Por otra parte, para definir cada requisito de usuario incluiremos en cada uno los siguientes atributos:

- **Identificador:** Sirve para identificar cada requisito y así facilitar la trazabilidad.
Identificador seguirá el siguiente formato: RUX-YYY
 - **RU:** Indica que es un requisito de usuario.
 - **X:** Indica el tipo de requisito, y admite los valores:
 - ✓ **C:** Indica que es un requisito de tipo Capacidad.
 - ✓ **R:** Indica que es un requisito de tipo Restricción.
 - ✓ **I:** Indica que es un requisito de tipo Inverso.
 - **YYY:** Número Comprendido entre 001 y 999 que identifica al requisito de usuario dentro de su tipo.
- **Prioridad:** Indica el valor de prioridad para facilitar la planificación del programador.
- **Necesidad:** El nivel de necesidad indicará si el requisito es negociable o no con el cliente.
- **Tipo:** Se distingue entre requisitos de capacidad, de restricción o inversos.
- **Descripción:** Reseña dónde se describe el requisito de usuario.

La tabla que muestra la información de cada requisito es una como la siguiente:

IDENTIFICADOR: RUX-YYY
PRIORIDAD:
NECESIDAD:
TIPO:
DESCRIPCIÓN:

Tabla 1-Ejemplo requisito de usuario

Una vez explicada toda la información necesaria, a continuación, se muestran todos los requisitos de usuario obtenidos.

3.2.1 Requisitos de capacidad

IDENTIFICADOR: RUC-01
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Capacidad
DESCRIPCIÓN: La aplicación calcula los siguientes indicadores de análisis técnico para un activo dado: RSI, oscilador Estocástico, Media Móvil, MACD, EMV y Bandas de Bollinger.

Tabla 2-RUC 01

IDENTIFICADOR: RUC-02
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Capacidad
DESCRIPCIÓN: La aplicación genera el gráfico correspondiente a cada indicador.

Tabla 3-RUC 02

IDENTIFICADOR: RUC-03
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Capacidad
DESCRIPCIÓN: La aplicación permite la creación y configuración de estrategias.

Tabla 4- RUC 03

IDENTIFICADOR: RUC-04
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Capacidad
DESCRIPCIÓN: La aplicación proporciona una recomendación de inversión para un activo o cartera y una estrategia o indicador dado.

Tabla 5- RUC 04

IDENTIFICADOR: RUC-05
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Capacidad
DESCRIPCIÓN: La aplicación permite la visualización de los gráficos de cada indicador.

Tabla 6- RUC 05



IDENTIFICADOR: RUC-06	
PRIORIDAD: Media	
NECESIDAD: Obligatoria	
TIPO: Capacidad	
DESCRIPCIÓN:	La aplicación permite probar la bondad de una estrategia creada mediante backtesting.

Tabla 7-RUC 06

IDENTIFICADOR: RUC-07	
PRIORIDAD: Media	
NECESIDAD: Obligatoria	
TIPO: Capacidad	
DESCRIPCIÓN:	La aplicación optimiza las estrategias del inversor.

Tabla 8-RUC 07

IDENTIFICADOR: RUC-08	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Capacidad	
DESCRIPCIÓN:	La aplicación cuenta con una interfaz gráfica desde la que se manejará la aplicación.

Tabla 9-RUC 08

IDENTIFICADOR: RUC-09	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Capacidad	
DESCRIPCIÓN:	La aplicación permite ver las ganancias o pérdidas de las inversiones realizadas.

Tabla 10-RUC 09

IDENTIFICADOR: RUC-10	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Capacidad	
DESCRIPCIÓN:	La aplicación permite elegir si trabajar con datos diarios, semanales o mensuales.

Tabla 11-RUC 10



IDENTIFICADOR: RUC-11
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Capacidad
DESCRIPCIÓN: La aplicación genera órdenes de compraventa de forma automática.

Tabla 12-RUC 11

IDENTIFICADOR: RUC-12
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Capacidad
DESCRIPCIÓN: La aplicación permite configurar la generación automática de órdenes.

Tabla 13-RUC 12

IDENTIFICADOR: RUC-13
PRIORIDAD: Media
NECESIDAD: Obligatoria
TIPO: Capacidad
DESCRIPCIÓN: La aplicación permite enviar órdenes a mercado a través de un bróker.

Tabla 14-RUC 13

IDENTIFICADOR: RUC-14
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Capacidad
DESCRIPCIÓN: La aplicación requiere de unas credenciales de autenticación para su uso.

Tabla 15-RUC 14

IDENTIFICADOR: RUC-15
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Capacidad
DESCRIPCIÓN: La aplicación obtiene cotizaciones.

Tabla 16-RUC 15

3.2.2 Requisitos inversos

IDENTIFICADOR: RUI-01
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Inverso
DESCRIPCIÓN: La aplicación no envía órdenes sin confirmación del usuario.

Tabla 17-RUI 01

3.2.3 Requisitos de restricción

IDENTIFICADOR: RUR-01
PRIORIDAD: Alta
NECESIDAD: Obligatorio
TIPO: Restricción
DESCRIPCIÓN: El programa se manejará desde navegadores Chrome, Firefox y Safari.

Tabla 18-RUR 01

IDENTIFICADOR: RUR-02
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Restricción
DESCRIPCIÓN: El programa se desarrollará en JAVA 5.

Tabla 19-RUR 02

IDENTIFICADOR: RUR-03
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Restricción
DESCRIPCIÓN: La utilidad de backtesting debe ofrecer, al menos, 3 test diferentes.

Tabla 20-RUR 03

IDENTIFICADOR: RUR-04
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Restricción
DESCRIPCIÓN: La lógica de la aplicación se implementará haciendo uso de web services.

Tabla 21-RUR 04



IDENTIFICADOR: RUR-05	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Restricción	
DESCRIPCIÓN:	La aplicación requiere de conexión a internet para su correcto funcionamiento.

Tabla 22-RUR 05

IDENTIFICADOR: RUR-06	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Restricción	
DESCRIPCIÓN:	La aplicación se desarrollará haciendo uso de PostgreSQL.

Tabla 23-RUR 06

IDENTIFICADOR: RUR-07	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Restricción	
DESCRIPCIÓN:	La aplicación se integrará con la aplicación de front-end existente.

Tabla 24-RUR 07

IDENTIFICADOR: RUR-08	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Restricción	
DESCRIPCIÓN:	Los cálculos de los indicadores se realizará en el lenguaje de programación R.

Tabla 25-RUR 08

IDENTIFICADOR: RUR-09	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Restricción	
DESCRIPCIÓN:	El programa funcionará para activos cotizados en mercados regulados de renta variable.

Tabla 26-RUR 09

3.3 Casos de uso

En este punto se especifican los casos de uso en base a la definición de los Requisitos de Usuario definidos anteriormente. El objetivo de describir los casos de uso es el de reflejar lo que el usuario puede realizar con la herramienta. Cada caso de uso viene definido por los siguientes atributos:

- Identificador: código que identifica unívocamente a cada uno de los casos de uso.
- Nombre: nombre identificativo de cada caso de uso.
- Actores: tipo de usuario para el que va dirigido el caso de uso.
- Objetivo: la finalidad que persigue el caso de uso en cuestión.
- Descripción: explicación concisa de cómo el actor del caso interactúa con el sistema
- Precondiciones: condiciones que debe cumplir para poder realizar la operación.
- Postcondiciones: estado en el que queda el sistema tras realizar la operación.
- Condiciones de fallo: posibles fallos durante la operación y la respuesta ante el fallo.

Los casos de uso se definen a continuación:

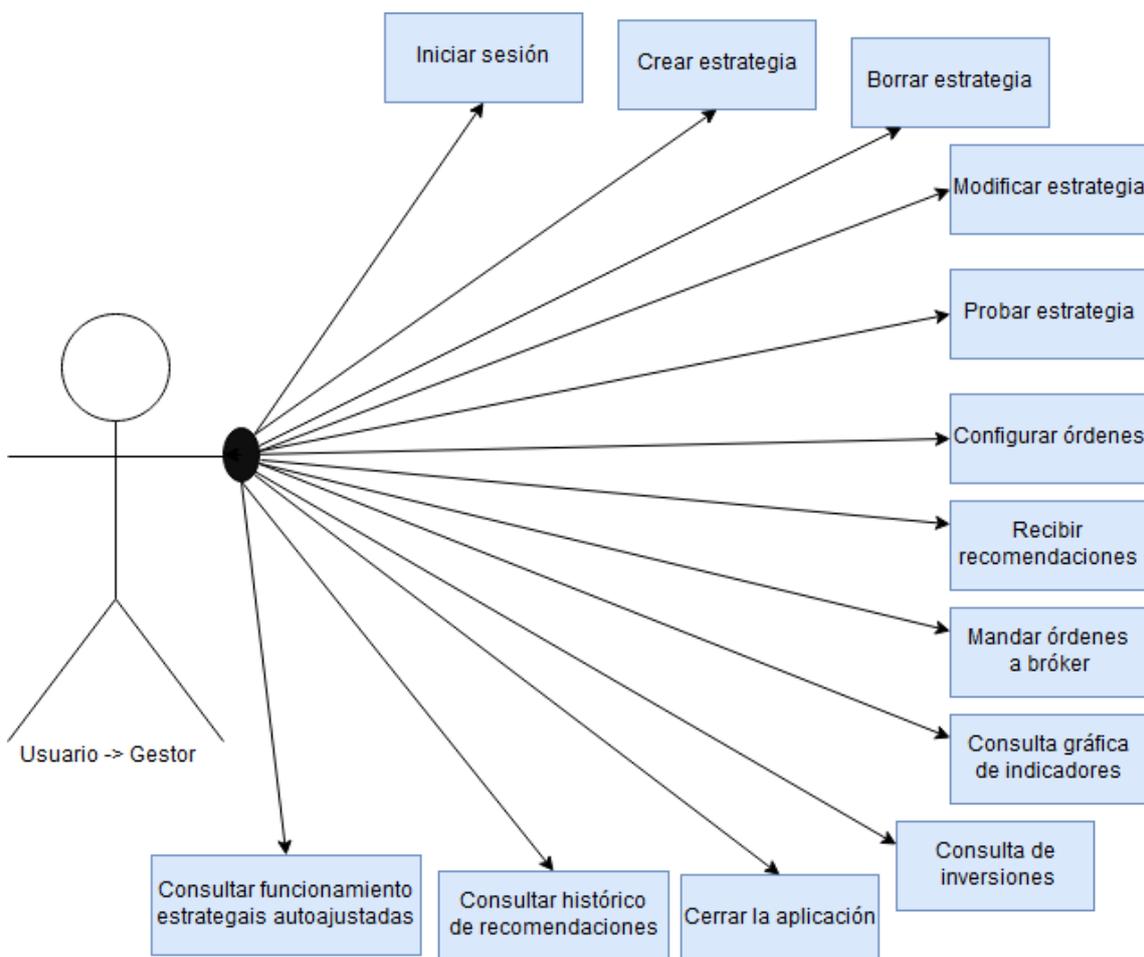


Ilustración 9 - Casos de uso

Identificador	CU-01
Nombre	Crear estrategia
Actores	Usuario
Objetivo	Crear una nueva estrategia compuesta por varios indicadores.
Descripción	El usuario, desde la pantalla de configuración, crea una estrategia añadiendo un nombre, uno o varios indicadores con sus pesos y dándole al botón “guardar”.
Precondiciones	Haber iniciado sesión en la aplicación y acceder a la pantalla de configuración y rellenar los datos de entrada requeridos.
Postcondiciones	La estrategia creada se ha guardado en base de datos.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 27-CU 01

Identificador	CU-02
Nombre	Borrar estrategia
Actores	Usuario
Objetivo	Borrar una estrategia existente.
Descripción	El usuario, desde la pantalla de configuración, borra una estrategia seleccionándola de un desplegable y dándole al botón “borrar”.
Precondiciones	Tener creada una estrategia. Haber iniciado sesión en la aplicación, acceder a la pantalla de configuración y rellenar los datos de entrada requeridos.
Postcondiciones	La estrategia existente se ha borrado de la base de datos.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 28-CU 02

Identificador	CU-03
Nombre	Probar estrategia
Actores	Usuario
Objetivo	Probar una estrategia existente mediante backtesting.
Descripción	El usuario, desde la pantalla de configuración, prueba una estrategia seleccionándola de un desplegable, rellenando los parámetros correspondientes y dándole al botón “probar”.
Precondiciones	Tener creada una estrategia. Haber iniciado sesión en la aplicación y acceder a la pantalla de configuración y rellenar los datos de entrada requeridos.
Postcondiciones	La estrategia existente se ha probado mediante backtesting y se muestra al usuario una ventana emergente con el resultado de la prueba.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 29-CU 03

Identificador	CU-04
Nombre	Modificar estrategia
Actores	Usuario
Objetivo	Modificar una estrategia existente.
Descripción	El usuario, desde la pantalla de configuración, selecciona una estrategia, modifica sus valores y la guarda dándole al botón "GUARDAR".
Precondiciones	Tener creada una estrategia. Haber iniciado sesión en la aplicación y acceder a la pantalla de configuración y rellenar los datos de entrada requeridos.
Postcondiciones	La estrategia modificada se ha actualizado en base de datos.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 30-CU 04

Identificador	CU-05
Nombre	Configurar órdenes
Actores	Usuario
Objetivo	Configurar la generación automática de órdenes.
Descripción	El usuario, desde la pantalla de configuración, rellena los distintos campos de la sección configuración de órdenes y guarda los criterios dándole al botón "GUARDAR".
Precondiciones	Haber iniciado sesión en la aplicación y acceder a la pantalla de configuración y rellenar los datos de entrada requeridos.
Postcondiciones	Los criterios establecidos para la generación automática de órdenes se almacenan en la base de datos.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 31-CU 05

Identificador	CU-06
Nombre	Recibir recomendaciones
Actores	Usuario
Objetivo	Generar recomendaciones de compraventa para una cartera o activo.
Descripción	El usuario, desde la pantalla de Recomendaciones, selecciona un activo o cartera sobre la que recibir recomendaciones, una estrategia a aplicar para generar las recomendaciones y un horizonte temporal de inversión y obtiene una recomendación de compraventa dándole al botón "CALCULAR".
Precondiciones	Tener creada una estrategia. Haber iniciado sesión en la aplicación y acceder a la pantalla de recomendaciones y rellenar los datos de entrada requeridos.
Postcondiciones	Se genera una recomendación de compraventa en función de los datos de entrada recibidos y la configuración de compraventa establecida que el usuario podrá visualizar.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 32-CU 06

Identificador	CU-07
Nombre	Mandar órdenes
Actores	Usuario
Objetivo	Mandar órdenes generadas automáticamente, mediante la recomendación, al mercado a través de un bróker.
Descripción	El usuario, desde la pantalla de Recomendaciones, introduce una cantidad de activos a comprar o vender, o deja la cantidad generada por defecto, y pulsa el botón "Enviar orden" de un activo en concreto o pulsa el botón "Enviar todas las órdenes" para enviar las órdenes de todos los activos.
Precondiciones	Tener configurada la conexión con un bróker en la aplicación de front-office. Haber iniciado sesión en la aplicación, acceder a la pantalla de recomendaciones y realizar una petición de recomendación.
Postcondiciones	La orden se envía al bróker configurado en la aplicación de front-office.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 33-CU 07

Identificador	CU-08
Nombre	Consultar indicadores
Actores	Usuario
Objetivo	Consultar gráficamente el resultado de los indicadores.
Descripción	El usuario, desde la pantalla de Recomendaciones, accede a la información de un activo en concreto de entre todos los que se ha realizado el cálculo de indicadores mediante un menú lateral.
Precondiciones	Tener creada una estrategia. Haber iniciado sesión en la aplicación y acceder a la pantalla de recomendaciones y realizar una petición de recomendación
Postcondiciones	El usuario puede ver el resultado del análisis técnico sobre el activo, incluyendo los gráficos de los indicadores.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 34-CU 08

Identificador	CU-09
Nombre	Consultar inversiones
Actores	Usuario
Objetivo	Consultar las pérdidas o ganancias de las órdenes enviadas mediante la aplicación.
Descripción	El usuario, desde la pantalla de "Inversiones", visualiza una tabla en la que podrá ver si ha ganado o perdido con las inversiones realizadas desde la aplicación. Se permite filtrar los resultados.
Precondiciones	Haber iniciado sesión en la aplicación, acceder a la pantalla de "Inversiones" y rellenar los datos de entrada, en caso de que se desee filtrar.
Postcondiciones	Se muestran los resultados de todas las inversiones realizadas desde la aplicación.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 35-CU 09

Identificador	CU-10
Nombre	Salir de la aplicación
Actores	Usuario
Objetivo	Salir y cerrar la aplicación.
Descripción	El usuario, desde cualquier pantalla de la aplicación puede pulsar el botón de salir.
Precondiciones	Haber iniciado sesión en la aplicación.
Postcondiciones	La sesión del usuario se cierra y se vuelve a la pantalla de login.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 36-CU 10

Identificador	CU-11
Nombre	Iniciar sesión
Actores	Usuario
Objetivo	Entrar a la aplicación y hacer uso de la misma.
Descripción	El usuario debe introducir su nombre y contraseña, así como la cartera que desea gestionar.
Precondiciones	Abrir la aplicación en un navegador compatible.
Postcondiciones	El usuario tiene acceso completo a las funcionalidades de la aplicación.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 37-CU 11

Identificador	CU-12
Nombre	Consultar histórico de recomendaciones
Actores	Usuario
Objetivo	Consultar el histórico de las recomendaciones realizadas.
Descripción	El usuario, desde la pantalla de “Histórico de recomendaciones”, visualiza una tabla en la que podrá ver todas las recomendaciones de compraventa realizadas. Se permite filtrar los resultados.
Precondiciones	Haber iniciado sesión en la aplicación, acceder a la pantalla de “Histórico de recomendaciones” y rellenar los datos de entrada, en caso de que se desee filtrar.
Postcondiciones	Se muestran los resultados de todas las recomendaciones realizadas desde la aplicación.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 38-CU 12

Identificador	CU-13
Nombre	Consultar resultados de las estrategias autoajustadas
Actores	Usuario
Objetivo	Consultar y comparar las recomendaciones y su resultado de las estrategias autoajustadas con las originales.
Descripción	El usuario, desde la pantalla de “Resultados de las estrategias autoajustadas”, visualiza una tabla en la que podrá ver todas las recomendaciones de compraventa realizadas por una estrategia original y su análoga autoajustada. Se permite filtrar los resultados.
Precondiciones	Haber iniciado sesión en la aplicación, acceder a la pantalla de “Resultados de las estrategias autoajustadas” y rellenar los datos de entrada, en caso de que se desee filtrar.
Postcondiciones	Se muestran los resultados de todas las recomendaciones realizadas desde la aplicación.
Condiciones de fallo	Si se detecta algún problema durante la ejecución, una ventana emergente con un mensaje informará al usuario.

Tabla 39-CU 13

3.4 Requisitos Software

En este punto se detallarán las características necesarias para el desarrollo de la herramienta. Cada una de las tablas tendrá una serie de información que se explica a continuación:

- **Identificador:** campo utilizado para identificar de forma unívoca un requisito. El identificador tendrá el siguiente formato: XX-YY, donde XX hace referencia al tipo de requisito e YY el número de requisito de ese tipo. XX tomará el valor RF para los requisitos funcionales, RNF para los requisitos no funcionales y RI para los requisitos de interfaz.
- **Prioridad:** campo que informa de la importancia del requisito. Sus posibles valores son: baja, media o alta.
- **Necesidad:** su implementación puede ser obligatorio u opcional.
- **Tipo:** se distingue entre tres tipos de requisitos.
 - **Funcionales:** especifican que es lo que tiene que hacer el software.
 - **No funcionales:** especifican características del funcionamiento, restricciones de diseño.
 - **De interfaz:** especifican características de la interfaz de usuario que tendrá el software.
- **Descripción:** explicación del requisito.
- **Justificación:** razonamiento de la existencia de ese requisito.

La tabla que muestra la información de cada requisito es una como la siguiente:

IDENTIFICADOR: XXX
PRIORIDAD:
NECESIDAD:
TIPO:
DESCRIPCIÓN:
JUSTIFICACIÓN:

Tabla 40 - Ejemplo tabla requisito software

A continuación, se muestran todos los requisitos software obtenidos:

3.4.1 Requisitos no funcionales

IDENTIFICADOR: RNF-01
PRIORIDAD: Alta
NECESIDAD: Obligatorio
TIPO: No funcional
DESCRIPCIÓN: La aplicación deberá funcionar en cualquier dispositivo que soporte el uso del navegador Chrome.
JUSTIFICACIÓN: Se trata de una aplicación web y este es uno de los navegadores que mayor cuota de mercado tiene en Windows y Android.

Tabla 41 RNF-01



IDENTIFICADOR: RNF-02
PRIORIDAD: Alta
NECESIDAD: Obligatorio
TIPO: No funcional
DESCRIPCIÓN: La aplicación deberá funcionar en cualquier dispositivo que soporte el uso del navegador Firefox.
JUSTIFICACIÓN: Se trata de una aplicación web y este es uno de los navegadores que mayor cuota de mercado tiene en Windows y Android.

Tabla 42 RNF-02

IDENTIFICADOR: RNF-03
PRIORIDAD: Alta
NECESIDAD: Obligatorio
TIPO: No funcional
DESCRIPCIÓN: La aplicación deberá funcionar en cualquier dispositivo que soporte el uso del navegador Safari.
JUSTIFICACIÓN: Se trata de una aplicación web y este es uno de los navegadores que mayor cuota de mercado tiene en dispositivos con macOS.

Tabla 43 RNF-03

IDENTIFICADOR: RNF-04
PRIORIDAD: Alta
NECESIDAD: Obligatorio
TIPO: No funcional
DESCRIPCIÓN: La aplicación se desarrollará en un computador con sistema operativo Windows 7.
JUSTIFICACIÓN: Es un sistema operativo estable y que no consume muchos recursos.

Tabla 44 RNF-04

IDENTIFICADOR: RNF-05
PRIORIDAD: Alta
NECESIDAD: Obligatorio
TIPO: No funcional
DESCRIPCIÓN: La aplicación se desarrollará haciendo uso de JAVA 5.
JUSTIFICACIÓN: Es una restricción establecida por el cliente.

Tabla 45 RNF-05

IDENTIFICADOR: RNF-06
PRIORIDAD: Alta
NECESIDAD: Obligatorio
TIPO: No funcional
DESCRIPCIÓN: La aplicación se desarrollará haciendo uso de PostgreSQL.
JUSTIFICACIÓN: Es una restricción establecida por el cliente.

Tabla 46 RNF-06

IDENTIFICADOR: RNF-07
PRIORIDAD: Alta
NECESIDAD: Obligatorio
TIPO: No funcional
DESCRIPCIÓN: La aplicación se integrará con la aplicación de front-end existente.
JUSTIFICACIÓN: Es una restricción establecida por el cliente.

Tabla 47 RNF-07

IDENTIFICADOR: RNF-08
PRIORIDAD: Alta
NECESIDAD: Obligatorio
TIPO: No funcional
DESCRIPCIÓN: Los cálculos de los indicadores se realizará en el lenguaje de programación R.
JUSTIFICACIÓN: Es una restricción establecida por el cliente.

Tabla 48 RNF-08

IDENTIFICADOR: RNF-09
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: No funcional
DESCRIPCIÓN: La aplicación requiere de conexión a internet para su correcto funcionamiento.
JUSTIFICACIÓN: Es una aplicación web, por lo que se requiere de conexión a internet para su uso siempre que no se esté en una red local.

Tabla 49-RNF 09

IDENTIFICADOR: RNF-10
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: No funcional
DESCRIPCIÓN: La aplicación hará uso de web services para implementar su funcionalidad.
JUSTIFICACIÓN: Es una aplicación web, por lo que se requiere de conexión a internet para su uso siempre que no se esté en una red local.

Tabla 50-RNF 10

3.4.2 Requisitos funcionales

IDENTIFICADOR: RF-01
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Funcional
DESCRIPCIÓN: La aplicación obtiene las cotizaciones diarias de cierre de mercado de los activos de renta variable negociados en los mercados regulados. Estas son: precio de cierre, máximo diario, mínimo diario y volumen.
JUSTIFICACIÓN: Son los datos necesarios para el cálculo de indicadores técnicos.

Tabla 51-RF 01

IDENTIFICADOR: RF-02
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Funcional
DESCRIPCIÓN: La aplicación almacena en base de datos los datos indicados en el RF-01.
JUSTIFICACIÓN: Es necesario persistir esta información para realizar los cálculos de los indicadores técnicos.

Tabla 52-RF 02

IDENTIFICADOR: RF-03
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Funcional
DESCRIPCIÓN: La aplicación permite al usuario configurar la generación automática de órdenes.
JUSTIFICACIÓN: Hay que hacer la herramienta lo más configurable posible y esta es una configuración esencial. El propio usuario debe decidir las cantidades de compra o venta de las órdenes (de ejecución opcional) que se crearán tras calcular el análisis técnico sobre un activo.

Tabla 53-RF 03

IDENTIFICADOR: RF-04
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Funcional
DESCRIPCIÓN: La aplicación incluye y calcula diversos indicadores técnicos.
JUSTIFICACIÓN: El cálculo de indicadores es la base del análisis técnico aplicación.

Tabla 54-RF 04

IDENTIFICADOR: RF-05	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Funcional	
DESCRIPCIÓN:	La aplicación genera una gráfica por indicador con los datos que generan los diversos indicadores técnicos.
JUSTIFICACIÓN:	Los gráficos son necesarios para ofrecerle al usuario la posibilidad de visualizar la evolución de un indicador en el tiempo y justificar el resultado de compra-venta que este puede ofrecer.

Tabla 55-RF 05

IDENTIFICADOR: RF-06	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Funcional	
DESCRIPCIÓN:	La aplicación permite al usuario configurar y crear estrategias compuestas de indicadores que son capaces de generar señales de compraventa.
JUSTIFICACIÓN:	Este elemento es fundamental en la aplicación, darle al usuario la posibilidad de conjuntar indicadores y ponderarlos para obtener un resultado de compraventa en conjunto. Los indicadores que no generan señales de compraventa no se pueden incluir en estrategias y solamente servirán para consultarlos de forma individual.

Tabla 56-RF 06

IDENTIFICADOR: RF-07	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Funcional	
DESCRIPCIÓN:	La aplicación permite probar las estrategias creadas mediante backtesting.
JUSTIFICACIÓN:	Es necesario que el usuario pueda probar sus creaciones sin poner en juego dinero. Por ello es necesario ofrecer unos test basándose en datos pasados para ver si la estrategia creada tendría o no un funcionamiento aceptable.

Tabla 57-RF 07

IDENTIFICADOR: RF-08	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Funcional	
DESCRIPCIÓN:	La aplicación realiza recomendaciones de compraventa mediante el uso de análisis técnico para un activo y un indicador dados.
JUSTIFICACIÓN:	Pieza básica para el funcionamiento de la aplicación. Esta es una de las cuatro posibles opciones que hay para el proceso de recomendación

Tabla 58-RF 08



IDENTIFICADOR: RF-09
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Funcional
DESCRIPCIÓN: La aplicación realiza recomendaciones de compraventa mediante el uso de análisis técnico para un activo y una estrategia dada.
JUSTIFICACIÓN: Pieza básica para el funcionamiento de la aplicación. Esta es una de las cuatro posibles opciones que hay para el proceso de recomendación

Tabla 59-RF 09

IDENTIFICADOR: RF-10
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Funcional
DESCRIPCIÓN: La aplicación realiza recomendaciones de compraventa mediante el uso de análisis técnico para todos los activos de una cartera y un indicador dado.
JUSTIFICACIÓN: Pieza básica para el funcionamiento de la aplicación. Esta es una de las cuatro posibles opciones que hay para el proceso de recomendación

Tabla 60-RF 10

IDENTIFICADOR: RF-11
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Funcional
DESCRIPCIÓN: La aplicación realiza recomendaciones de compraventa mediante el uso de análisis técnico para todos los activos de una cartera y una estrategia dada.
JUSTIFICACIÓN: Pieza básica para el funcionamiento de la aplicación. Esta es una de las cuatro posibles opciones que hay para el proceso de recomendación

Tabla 61-RF 11

IDENTIFICADOR: RF-12
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Funcional
DESCRIPCIÓN: La aplicación muestra la gráfica correspondiente a un indicador para un activo determinado.
JUSTIFICACIÓN: El usuario que hace uso de análisis técnico es probable que quiera consultar los resultados obtenidos en el proceso de recomendación de forma gráfica.

Tabla 62-RF 12



IDENTIFICADOR: RF-13	
PRIORIDAD: Baja	
NECESIDAD: Opcional	
TIPO: Funcional	
DESCRIPCIÓN:	La aplicación muestra una breve descripción de cada indicador existente.
JUSTIFICACIÓN:	Recomendable para la claridad del usuario, aunque normalmente conocen lo que utilizan.

Tabla 63-RF 13

IDENTIFICADOR: RF-14	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Funcional	
DESCRIPCIÓN:	La aplicación genera órdenes de opcional ejecución en base a las recomendaciones generadas según RF-08, RF-09, RF-10 y RF-10 y a la configuración que realizó el usuario según el RF-03.
JUSTIFICACIÓN:	Esta parte de la aplicación automatiza el proceso de comprar o vender un activo en base al resultado de las recomendaciones

Tabla 64-RF 14

IDENTIFICADOR: RF-15	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Funcional	
DESCRIPCIÓN:	La aplicación permite la modificación de los valores de las órdenes opcionales generadas según el RF-14.
JUSTIFICACIÓN:	El usuario configura unos valores por defecto que quiere establecer en las órdenes que se generan de forma automática, pero es probable que no siempre quiera aplicar esos valores y es necesario darla la posibilidad de manipularlos.

Tabla 65-RF 15

IDENTIFICADOR: RF-16	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Funcional	
DESCRIPCIÓN:	La aplicación envía las órdenes generadas según el RF-14 a un bróker.
JUSTIFICACIÓN:	Necesario para ejecutar las órdenes generadas automáticamente, sin esta opción carecería de sentido el proceso de generación y configuración de órdenes automáticas.

Tabla 66-RF 16



IDENTIFICADOR: RF-17
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Funcional
DESCRIPCIÓN: La aplicación permite visualizar las pérdidas o ganancias respecto al día presente de las órdenes que se ejecutaron en el pasado en base a la sugerencia de la aplicación (RF-14).
JUSTIFICACIÓN: Siempre es conveniente informar al usuario del éxito o fracaso de una ejecución y del resultado.

Tabla 67-RF 17

IDENTIFICADOR: RF-18
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Funcional
DESCRIPCIÓN: La aplicación optimiza los pesos de las estrategias generadas por el usuario según el RF-06.
JUSTIFICACIÓN: Se pretende mejorar las estrategias creadas por los usuarios.

Tabla 68-RF 18

IDENTIFICADOR: RF-19
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Funcional
DESCRIPCIÓN: La aplicación permite al usuario comparar los resultados de las recomendaciones realizadas por una estrategia original y su análoga optimizada.
JUSTIFICACIÓN: Esto permitirá al usuario ver si la estrategia autoajustada está funcionando mejor que la estrategia original.

Tabla 69-RF 19

IDENTIFICADOR: RF-20
PRIORIDAD: Media
NECESIDAD: Opcional
TIPO: Funcional
DESCRIPCIÓN: La aplicación dará feedback de las acciones realizadas por el usuario.
JUSTIFICACIÓN: Siempre es conveniente informar al usuario del éxito o fracaso de una acción.

Tabla 70-RF 20

IDENTIFICADOR: RF-21	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Funcional	
DESCRIPCIÓN:	La aplicación permite elegir el horizonte temporal sobre el que trabajar: diario, semanal o mensual o lo que es lo mismo, corto plazo, medio plazo o largo plazo.
JUSTIFICACIÓN:	Esto es necesario para dar la flexibilidad de invertir a largo, corto y medio plazo.

Tabla 71-RF 21

IDENTIFICADOR: RF-22	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Funcional	
DESCRIPCIÓN:	La aplicación permite al usuario ver un histórico de recomendaciones.
JUSTIFICACIÓN:	Permite ver al usuario recomendaciones pasadas.

Tabla 72-RF 22

3.4.3 Requisitos de interfaz

IDENTIFICADOR: RI-01	
PRIORIDAD: Alta	
NECESIDAD: Obligatoria	
TIPO: Interfaz	
DESCRIPCIÓN:	La aplicación tendrá las siguientes pantallas: <ul style="list-style-type: none">• Pantalla de menú de configuraciones.• Pantalla de menú de análisis técnico.• Pantalla de seguimiento de inversiones.• Pantalla de histórico de recomendaciones.• Pantalla de resultado de estrategias autoajustadas.• Pantalla de gestión de estrategias.• Pantalla de configuración de la generación de órdenes automáticas.• Pantalla de otras configuraciones.• Pantalla de recomendaciones o análisis técnico.• Pantalla para visualización de gráficos.
JUSTIFICACIÓN:	Se requiere que la aplicación tenga el menor número de pantallas posible.

Tabla 73-RI 01



IDENTIFICADOR: RI-02
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Interfaz
DESCRIPCIÓN: Los mensajes de error serán mostrados en ventanas emergentes.
JUSTIFICACIÓN: Permite una mejor visualización por parte del usuario.

Tabla 74-RI 02

IDENTIFICADOR: RI-03
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Interfaz
DESCRIPCIÓN: Los textos estarán escritos en español.
JUSTIFICACIÓN: La aplicación tiene como mercado objetivo el sector español.

Tabla 75-RI 03

IDENTIFICADOR: RI-04
PRIORIDAD: Alta
NECESIDAD: Obligatoria
TIPO: Interfaz
DESCRIPCIÓN: Los mensajes informativos serán mostrados en ventanas emergentes.
JUSTIFICACIÓN: Permite una mejor visualización por parte del usuario de las cosas importantes.

Tabla 76-RI 04

IDENTIFICADOR: RI-05
PRIORIDAD: Media
NECESIDAD: Opcional
TIPO: Interfaz
DESCRIPCIÓN: La web debe ser <i>responsive</i> .
JUSTIFICACIÓN: Lo normal es que los gestores operen desde un ordenador, pero siempre es bueno ofrecer soluciones universales.

Tabla 77-RI 05

4. Diseño

En este apartado del documento se presenta el diseño de la aplicación siguiendo las necesidades descritas en la parte de análisis. Se presentará la arquitectura elegida y las decisiones de diseño tomadas antes de la implementación de la herramienta.

4.1 Arquitectura del sistema

Para definir la arquitectura del sistema se ha decidido dividir el sistema en diferentes capas que se comunican con las capas contiguas. La arquitectura se divide en 3 capas de la siguiente manera:

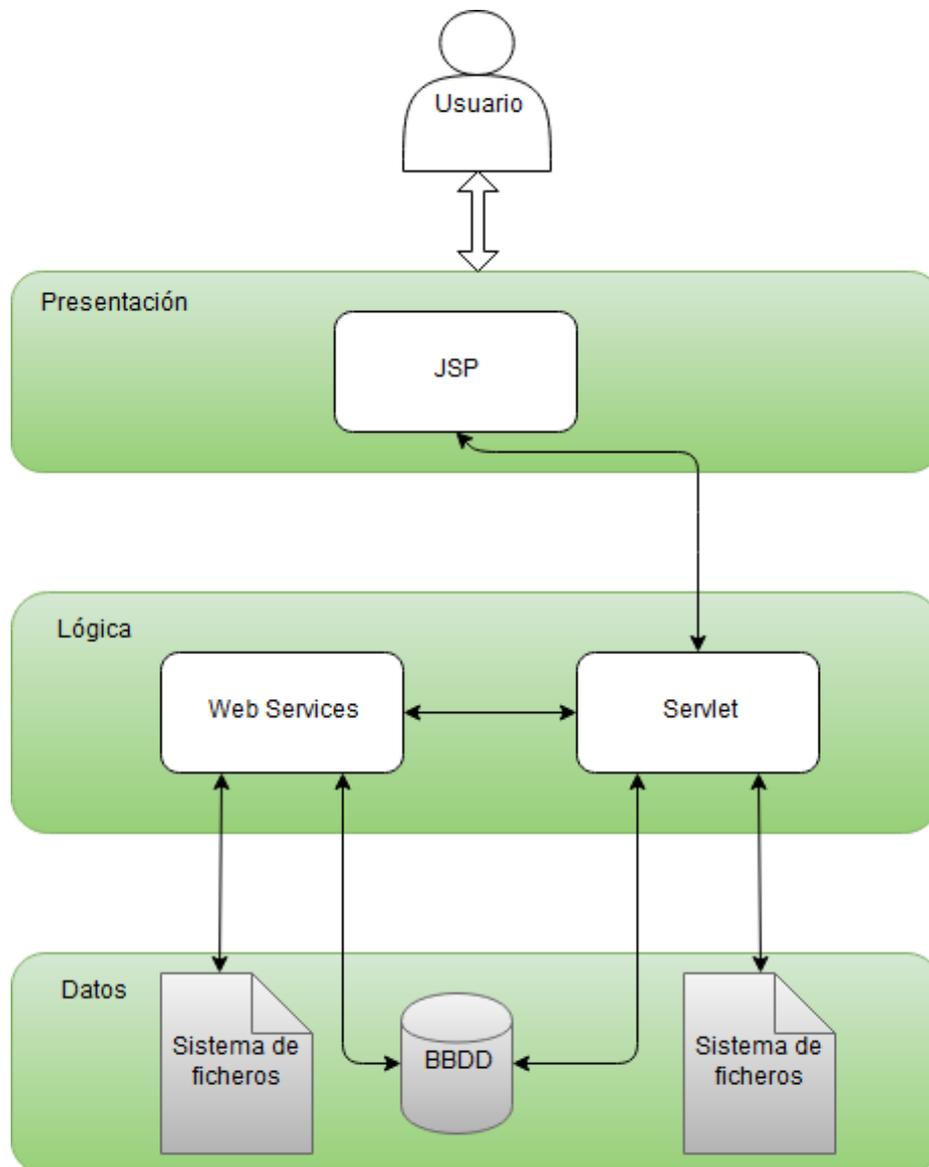


Ilustración 10 - Diagrama de la arquitectura

Capa de presentación

La capa de presentación es la capa con la que el usuario interactúa. Mediante la implementación de diferentes JavaServer Pages (JSP), el usuario podrá interactuar con los distintos menús y formularios existentes en la aplicación, así como visualizar los datos deseados. Para realizar distintos procesos pertenecientes a la capa lógica, se requiere de una serie de datos que el usuario ha de introducir en los distintos JSP existentes. Estos JSP, por tanto, no solo se encargarán de mostrar el apartado visual de la aplicación y los datos generados, sino que también enviará datos a la capa que implementa la lógica.

Por otro lado, la capa lógica también enviará a la capa de usuario información, los resultados de los cálculos exigidos o de las consultas deseadas. Por esta razón la relación entre ambas capas es bidireccional.

Capa Lógica

La capa de lógica es la que contiene la funcionalidad fundamental de la herramienta. Esta capa podemos separarla en dos módulos o tareas bien diferenciadas que se explicarán a continuación.

Uno de los módulos es el formado por el servlet y los handlers. El servlet es el punto de entrada de los datos y operaciones que el cliente quiere realizar y cuya procedencia es la capa de presentación. En función de la operación a realizar, el servlet delegará la petición en handler u otro, al que se le pasarán también los datos de entrada. Cada handler se encarga de gestionar una operación distinta en la aplicación. Los handlers tienen la capacidad de conectarse con la capa de datos con el objetivo de consultar datos y también de insertar o actualizar los mismos. También tienen la capacidad de comunicarse con el otro módulo de la capa que implementa la lógica de la aplicación, los web services. Los handlers usarán los web services para resolver distintas tareas.

El segundo módulo, estará formado por distintos web services ya que se quiere seguir una filosofía SOA. Los web services se encargarán de resolver distintas tareas que debe realizar la aplicación. Estos recibirán llamadas desde los handlers y se comunicarán con ellos de vuelta para devolverles el resultado de la operación pedida. Los web services también pueden acceder a la capa de datos para consultar y escribir datos.

Capa de datos

La capa de datos es la capa que contiene los datos que la aplicación requiere para funcionar. Como se comentó anteriormente tanto los web services como los handlers tienen acceso a los datos y la comunicación entre ambas capas es bidireccional. Dentro de esta capa distinguimos tres unidades de almacenamiento de datos. Por un lado, hay una base de datos que se encargará de almacenar todos los datos estructurados de la aplicación en una base de datos relacional. Por otro lado, tenemos dos sistemas de ficheros que almacenarán las imágenes que contienen los gráficos de los distintos indicadores. Son dos sistemas de ficheros distintos porque los web services y el resto de la aplicación pueden estar ejecutando en servidores diferentes.

4.2 Especificación del entorno tecnológico

Los distintos componentes necesarios para el uso y el funcionamiento de la aplicación requieren de unos requisitos **mínimos** hardware y software. En este apartado definiremos estos requerimientos mínimos para poder utilizar y desarrollar la herramienta.

2.4.1 Hardware

Los requisitos hardware mínimos necesarios son los siguientes:

Ciente:

- El computador debe tener, al menos, 200 Mb libres en el disco duro.
- El computador debe tener, al menos, 2 Gb de memoria RAM.
- El computador debe tener un procesador Intel Core Duo T2400 o superior.
- Conexión a internet vía WiFi y/o Ethernet.

Entorno de desarrollo:

- 4 Gb RAM.
- Intel Core i3 4005U.
- 500 Gb de almacenamiento.
- Conexión WiFi y Ethernet.
- Al menos un puerto USB 2.0.

Servidor:

- El computador debe tener, al menos, 15 Gb libres en el disco duro.
- El computador debe tener, al menos, 16 Gb de memoria RAM.
- Procesador Intel® Xeon® Processor E3-1245 v5 o superior.
- Conexión Ethernet.

2.4.2 Software

Los requerimientos mínimos software necesarios son los siguientes:

Ciente:

- Navegador Chrome o Firefox.

Entorno de desarrollo:

- Sistema operativo Windows 7 o superior.
- Microsoft Office 2010 o superior.
- Java, R, Rserve, PostgreSQL y NetBeans.

Servidor:

- Sistema operativo Windows.
- Java, R, Rserve, PostgreSQL.

4.3 Diseño de la base de datos

El diseño de la base de datos se ha realizado siguiendo un modelo relacional, el esquema diseñado es el siguiente:

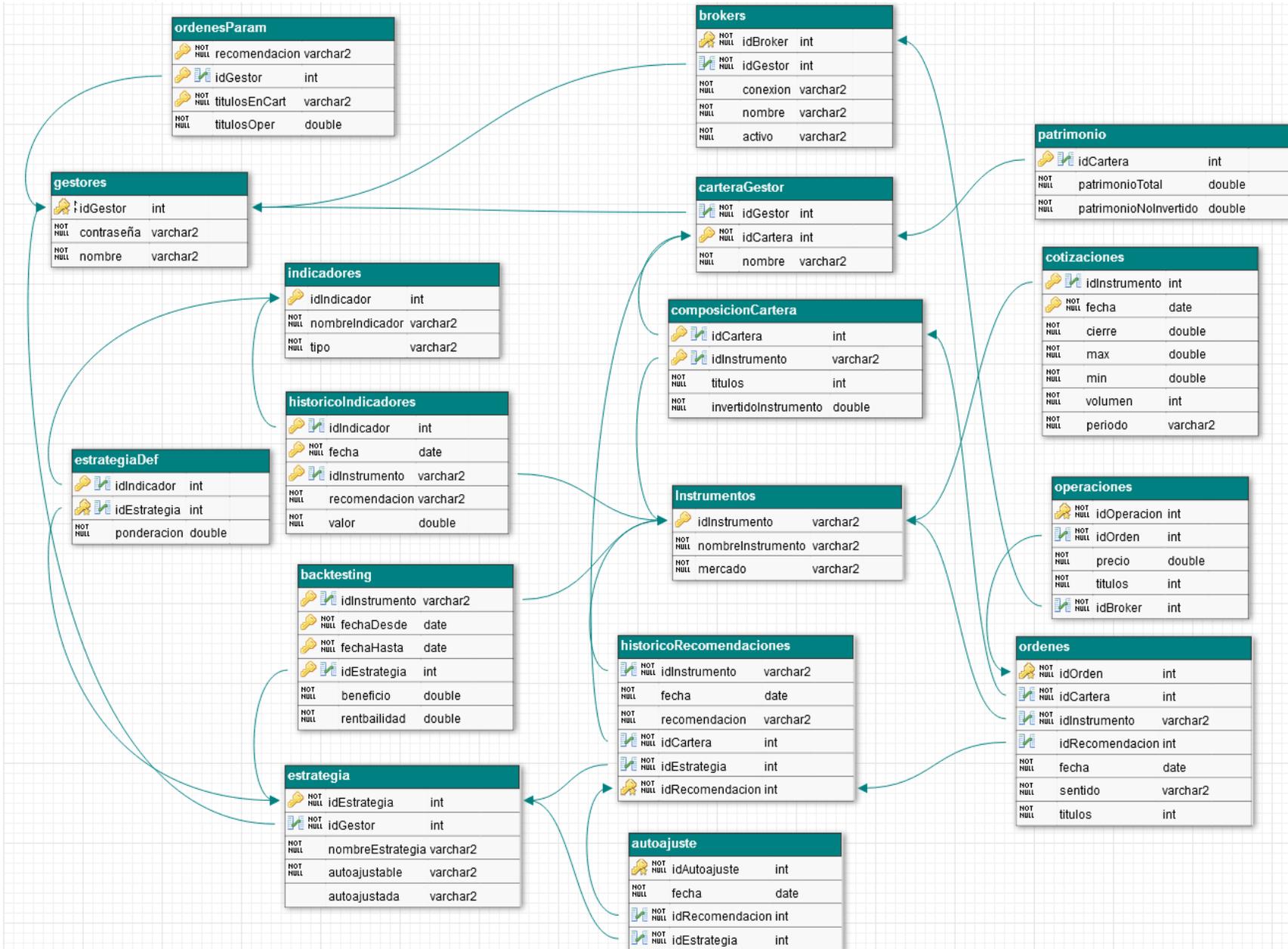


Ilustración 11 - Esquema de base de datos



Tabla instrumentos

Esta tabla se encarga de almacenar todos los instrumentos con los que la aplicación puede trabajar. Esta tabla ya existe en el sistema y su composición es mayor a lo presentado, únicamente se muestra lo imprescindible para nuestra aplicación.

Campo	Tipo	Descripción
idInstrumento	varchar2	Identificador único de un instrumento.
nombreInstrumento	varchar2	Nombre asociado al instrumento.
mercado	varchar2	Mercado en el que se negocia el instrumento.

Tabla 78-Tabla instrumentos

Tabla cotizaciones

Esta tabla se encarga de almacenar toda la información necesaria en relación con las cotizaciones de los activos. Almacena los datos básicos que necesitan los indicadores técnicos para ser calculados. Esta tabla ya existe en el sistema y su composición es mayor a lo presentado, únicamente se muestra lo imprescindible para nuestra aplicación.

Campo	Tipo	Descripción
idInstrumento	int	Identificador único de un instrumento. Se corresponde con el campo idInstrumento de la tabla instrumentos.
fecha	date	Determina la fecha a la que pertenecen los datos del registro.
cierre	double	Precio de cierre del instrumento.
max	double	Precio máximo del día del instrumento.
min	double	Precio mínimo del día del instrumento.
volumen	int	Volumen negociado del instrumento en el día.
periodo	varchar2	Horizonte temporal de los datos. Puede ser "D" diario, "S" semanal o "M" mensual.

Tabla 79-Tabla cotizaciones

Tabla carteraGestor

Esta tabla almacena las carteras existentes y su relación con el gestor que las gestiona.

Campo	Tipo	Descripción
idCartera	int	Identificador único de una cartera.
idGestor	int	Identificador único del gestor. Se corresponde con el campo idGestor de la tabla gestores.
nombre	varchar2	Nombre de la cartera.

Tabla 80 - Tabla carteraGestor

Tabla composicionCartera

Esta tabla se encarga de almacenar la composición actual de la cartera. Esta tabla ya existe en el sistema y su tamaño es mayor a lo presentado, únicamente se muestra lo imprescindible para nuestra aplicación.

Campo	Tipo	Descripción
idCartera	int	Identificador único de una cartera. Se corresponde con el campo idCartera de la tabla carteraGestor.
idInstrumento	varchar2	Identificador único de un instrumento. Se corresponde con el campo idInstrumento de la tabla instrumentos.
titulos	int	Número de títulos que se tienen del instrumento.
invertidoInstrumento	double	Cantidad invertida en el instrumento.

Tabla 81-Tabla carteras

Tabla patrimonio

Esta tabla se encarga de almacenar el patrimonio de una cartera de forma actualizada.

Campo	Tipo	Descripción
idCartera	int	Identificador único de una cartera. Se corresponde con el campo idCartera de la tabla carteraGestor.
patrimonioTotal	double	Patrimonio total del fondo/cartera.
patrimonioNoInvertido	double	Patrimonio no invertido del fondo/cartera.

Tabla 82-Tabla patrimonio

Tabla gestores

Esta tabla se encarga de almacenar los datos referentes a los gestores que usan la aplicación. Esta tabla ya existe en el sistema y su composición es mayor a lo presentado, únicamente se muestra lo imprescindible para nuestra aplicación.

Campo	Tipo	Descripción
idGestor	int	Identificador único de un gestor.
contraseña	varchar2	Contraseña de acceso a las aplicaciones. Válida para el inicio de sesión de nuestra aplicación.
nombre	varchar2	Nombre real del gestor.

Tabla 83-Tabla gestores

Tabla ordenesParam

Esta tabla se encarga de almacenar la configuración de cada gestor para la generación automática de órdenes.

Campo	Tipo	Descripción
idGestor	int	Identificador único de un gestor. Se corresponde con el campo idGestor de la tabla gestores.
recomendacion	varchar2	Valor que se corresponde con uno de los presentes en la tabla recomendaciones.
titulosEnCart	varchar2	Campo que tomará dos posibles valores "Sí" y "No". Diferencia el hecho de invertir en un instrumento del que ya se poseen títulos o no.
titulosOper	double	Campo que almacenará un porcentaje de compra o venta en base a un valor. Este valor será el correspondiente al número de títulos que ya se poseen en caso de poseer títulos de un activo o será un porcentaje del patrimonio sin invertir que tiene el fondo en caso de no poseer títulos de ese activo.

Tabla 84-Tabla Parámetros para órdenes

Tabla indicadores

Esta tabla se encarga de almacenar los distintos indicadores que podrán ser utilizados y calculados en la aplicación.

Campo	Tipo	Descripción
idIndicador	int	Identificador único de un indicador.
nombreIndicador	varchar2	Nombre del indicador. Es único.
tipo	varchar2	Tipo al que pertenece el indicador. Puede tomar uno de los siguientes valores: "Tendencia", "Momentum", "Volatilidad" o "Volumen".

Tabla 85- Tabla indicadores

Tabla historicoIndicadores

Esta tabla se encarga de almacenar las recomendaciones realizadas para un activo, una fecha y un indicador en concreto.

Campo	Tipo	Descripción
idIndicador	int	Identificador único de un indicador. Se corresponde con el campo idIndicador de la tabla indicadores.
idInstrumento	varchar2	Identificador único de un instrumento. Se corresponde con el campo idInstrumento de la tabla instrumentos.
fecha	date	Fecha en la que se realizó la recomendación para el activo dado.
recomendación	varchar2	Valor que se corresponde con uno de los presentes en la tabla recomendaciones.
valor	double	Valor numérico resultante del cálculo del indicador.

Tabla 86-Tabla histórico de indicadores

Tabla estrategia

Esta tabla se encarga de almacenar las estrategias creadas por los usuarios.

Campo	Tipo	Descripción
idEstrategia	int	Identificador único de una estrategia.
nombreEstrategia	varchar2	Nombre que el usuario da a su estrategia.
idGestor	int	Identificador único de un gestor. Se corresponde con el campo idGestor de la tabla gestores.
autoajustable	varchar2	Indica si el gestor quiere que esta estrategia se autoajuste de forma paralela a su estrategia original. Valores posibles: "SI" o "NO"
autoajustada	varchar2	Indica si la estrategia está siendo autoajustada o no. Sirve para diferenciar la estrategia original de la análoga autoajustable.

Tabla 87-Tabla de estrategias

Tabla estrategiaDef

Esta tabla se encarga de almacenar la definición de las estrategias creadas por los usuarios.

Campo	Tipo	Descripción
idEstrategia	int	Identificador único de una estrategia. Se corresponde con el campo idEstrategia de la tabla estrategia.
idIndicador	int	Identificador único de un indicador. Se corresponde con el campo idIndicador de la tabla indicadores.
ponderacion	double	Peso que se le asigna a cada indicador que compone la estrategia.

Tabla 88-Tabla definición de estrategias

Tabla historicoRecomendaciones

Esta tabla se encarga de almacenar las recomendaciones realizadas sobre para un activo, una fecha y una estrategia en concreto.

Campo	Tipo	Descripción
idRecomendacion	int	Identificador único de una recomendación.
idEstrategia	int	Identificador único de una estrategia. Se corresponde con el campo idEstrategia de la tabla estrategia.
idInstrumento	varchar2	Identificador único de un instrumento. Se corresponde con el campo idInstrumento de la tabla instrumentos.
fecha	date	Fecha en la que se realizó la recomendación para la estrategia dada.
idCartera	int	Identificador único de una cartera. Se corresponde con el campo idCartera de la tabla carteraGestor.
recomendación	varchar2	Valor que se corresponde con uno de los presentes en la tabla recomendaciones.

Tabla 89-Tabla histórico de recomendaciones

Tabla órdenes

Esta tabla se encarga de almacenar las órdenes que se envían al bróker. Es una tabla que existía con anterioridad en el sistema a la cual se le ha añadido el campo idRecomendacion que servirá para identificar si esta orden fue generada a través de una recomendación de la aplicación.

Campo	Tipo	Descripción
idOrden	int	Identificador único de una orden.
idRecomendacion	int	Identificador único de una recomendación. Se corresponde con el campo idRecomendacion de la tabla historicoRecomendaciones. Este campo se utilizará para identificar si una orden fue generada y realizada desde el recomendador de la aplicación.
idCartera	int	Identificador único de una cartera. Se corresponde con el campo idCartera de la tabla carteras.
idInstrumento	varchar2	Identificador único de un instrumento. Se corresponde con el campo idInstrumento de la tabla instrumentos.
fecha	date	Fecha en la que se mandó la orden para la estrategia dada.
sentido	varchar2	Campo que indica si la orden enviada es de compra o de venta.
titulos	int	Campo que contiene el volumen de la orden.

Tabla 90-Tabla de órdenes

Tabla backtesting

Esta tabla se encarga de almacenar los datos referentes a las pruebas realizadas sobre una estrategia para comprobar su bondad.

Campo	Tipo	Descripción
idEstrategia	int	Identificador único de una estrategia. Se corresponde con el campo idEstrategia de la tabla estrategia.
idInstrumento	varchar2	Identificador único de un instrumento. Se corresponde con el campo idInstrumento de la tabla instrumentos.
fechaDesde	date	Fecha inicio para la obtención de las cotizaciones que se utilizarán en el test.
fechaHasta	date	Fecha fin para la obtención de las cotizaciones que se utilizarán en el test.
beneficio	double	Pérdida o ganancia que genera el activo en el periodo, estrategia e instrumento especificados.
rentabilidad	double	Rentabilidad que genera el activo en el periodo, estrategia e instrumento especificados.

Tabla 91-Tabla backtesting

Tabla brokers

Esta tabla se encarga de almacenar los datos referentes a los distintos brókers que el usuario tiene dados de alta en la aplicación de front-office. Es una tabla existente de la que se hace uso.

Campo	Tipo	Descripción
idBroker	int	Identificador único de una conexión a un bróker.
idGestor	int	Identificador único de un gestor. Se corresponde con el campo idGestor de la tabla gestores.
conexion	varchar2	Datos de la conexión al bróker.
nombre	varchar2	Nombre del bróker.
activo	varchar2	Si la conexión está activa o no actualmente. Solo habrá una conexión activa.

Tabla 92- Tabla brokers

Tabla operaciones

Esta tabla se encarga de almacenar la ejecución de las órdenes, es decir, las operaciones realizadas.

Campo	Tipo	Descripción
idOperacion	int	Identificador único de una operacion.
idOrden	int	Identificador único de una orden. Se corresponde con el campo idOrden de la tabla órdenes.
precio	double	Precio que se ha pagado por cada título en la operación.
titulos	int	Campo que contiene el volumen de la operación.
idBroker	int	Identificador único de una conexión a un bróker. Se corresponde con el campo idBroker de la tabla brokers.

Tabla 93 - Tabla operaciones

Tabla autoajuste

Esta tabla almacena los autoajustes a realizar por la aplicación sobre las estrategias autoajustables.

Campo	Tipo	Descripción
idAutoajuste	int	Identificador único de un ajuste.
idRecomendacion	int	Identificador único de una recomendación. Se corresponde con el campo idRecomendacion de la tabla historicoRecomendaciones.
fecha	date	Fecha en la que se debe llevar a cabo el autoajuste.
idEstrategia	int	Identificador único de una estrategia. Se corresponde con el campo idEstrategia de la tabla estrategia.

Tabla 94 - Tabla autoajuste

4.4 Diseño de la interfaz

El diseño de la interfaz de la aplicación se ha realizado mediante la realización de distintos prototipos de cada una de las pantallas de la aplicación. El objetivo de este punto es representar las distintas características que ha de tener la aplicación en las diferentes pantallas, independientemente de su distribución o forma de implementación. A continuación, se mostrarán y describirán cada una de las partes de la interfaz que componen la aplicación.

Pantalla de inicio de sesión

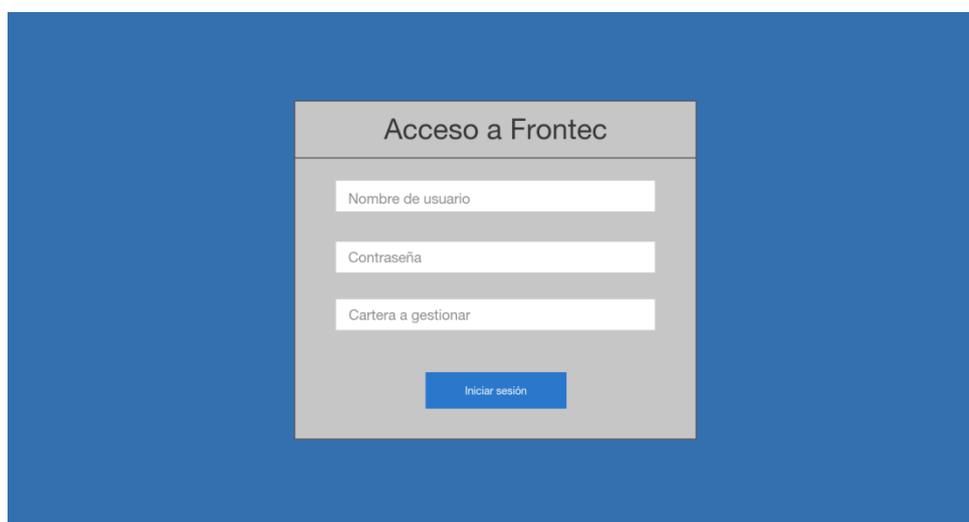


Ilustración 12 - Interfaz: Login

Para hacer uso de la aplicación hace falta estar dado de alta en el sistema, es decir, tener acceso a la aplicación de front-office con la que se integra esta aplicación. En esta pantalla se requieren el nombre de usuario, la contraseña y el nombre o el identificador de la cartera a gestionar. Mediante el botón “iniciar sesión” se realizará la validación y autenticación del usuario.

Menú de navegación



Ilustración 13 - Interfaz: Menú de Navegación

Este menú estará presente en todas las pantallas de la aplicación y permite realizar la navegación entre las tres pantallas principales que componen la aplicación mediante tres pestañas diferentes. Cuenta con un botón para cerrar la sesión en la aplicación en la parte superior derecha y en la parte superior del menú se muestra el Nombre del gestor que está usando la aplicación y la cartera sobre la que se está trabajando.

Menú Análisis Técnico

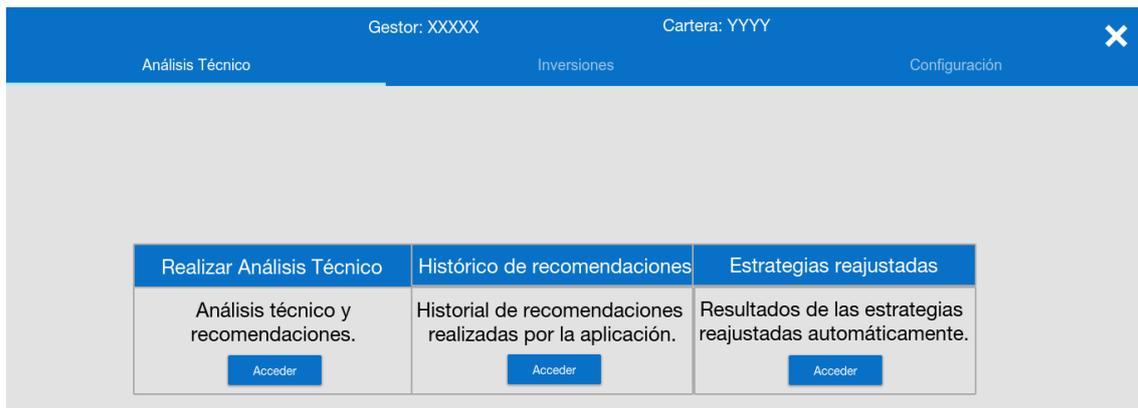


Ilustración 14 - Interfaz: Menú Análisis Técnico

Esta pantalla representa el menú correspondiente a la parte de análisis técnico. Desde aquí se puede acceder a tres funcionalidades del sistema: a la pantalla de análisis técnico, a la pantalla de visualización del histórico de recomendaciones y a la pantalla de visualización del histórico de recomendaciones de estrategias ajustadas automáticamente.

Pantalla de recomendaciones o Análisis técnico

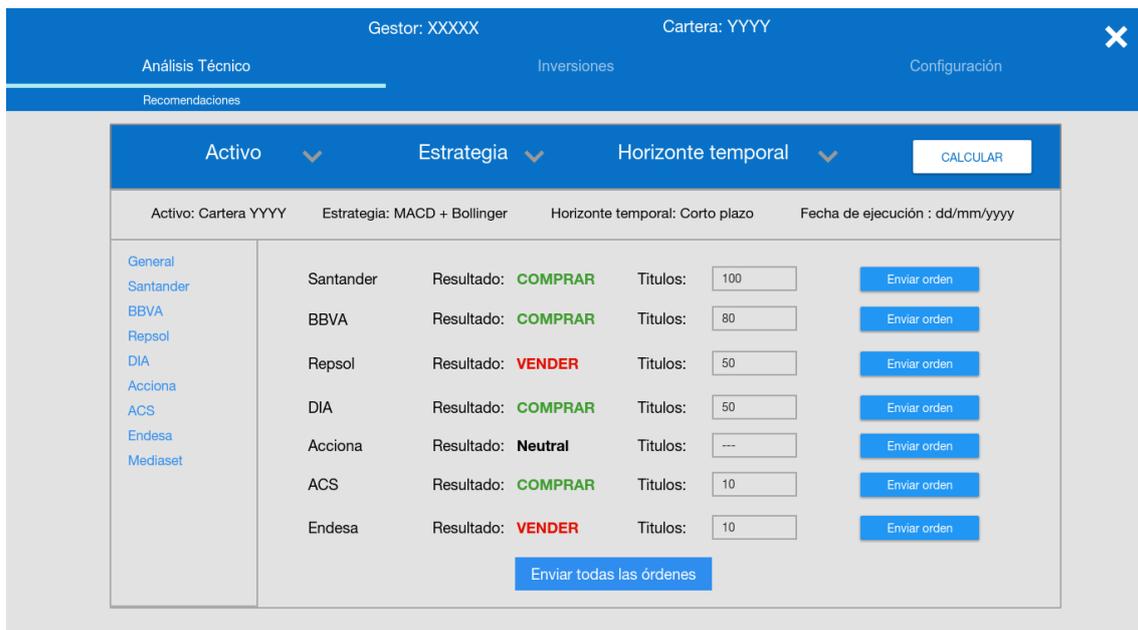


Ilustración 15 - Interfaz: Pantalla de recomendaciones

Desde esta pantalla el usuario podrá realizar las peticiones de recomendación. Para ello tiene que rellenar los campos que hay en el menú situado debajo del menú principal y darle al botón "CALCULAR".

Tras realizar una petición de recomendación, los resultados correspondientes aparecerán justo debajo del menú de petición. Se puede realizar una petición de recomendación para una cartera entera o para un único activo.

Tras calcular las recomendaciones, se habilita un menú en el lateral izquierdo en el que el usuario puede obtener una vista general de las recomendaciones en caso de ser una cartera o puede obtener una vista específica de un activo concreto con información más específica.

En cuanto a la vista general, para cada activo de la cartera se muestra su nombre, el resultado de la recomendación y un input numérico modificable que hace referencia al número de títulos a comprar o vender según la recomendación. Finalmente, se tiene la posibilidad de enviar una **orden de mercado**, en el sentido de la recomendación, al bróker con el número de títulos prefijado por el usuario, mediante el botón habilitado para ello. También se habilita un botón que envía todas las órdenes de compra o de venta que se generan tras la recomendación. No existe posibilidad de manipulación ni de envío de órdenes para activos con recomendaciones neutras.

En cuanto a la vista específica de un activo, su aspecto sería algo similar a lo que se ve en la siguiente imagen:



Ilustración 16 - Interfaz: Pantalla de recomendaciones específica de un activo

Se ofrece la misma posibilidad de enviar la orden generada y se muestra la misma información que para la visión general con el añadido de poder visualizar distintas gráficas referentes a datos del activo en cuestión. Estas gráficas que se pueden visualizar son la cotización del activo en el horizonte temporal que eligió a la hora de pedir la recomendación y por otro lado una gráfica para cada indicador que compone la estrategia o la gráfica del indicador en caso de que se pida recomendación para un indicador solo. Por último, se mostrará un resumen del resultado que proporciona cada indicador de la estrategia.

Pantalla Histórico de Recomendaciones

FECHA	ACTIVO	ESTRATEGIA	CANTIDAD	SENTIDO	PRECIO	PRECIO ACTUAL	P/G
07-03-2017	Santander	MACD + Bollinger	100	Compra	5,47 €	5,57 €	10 €

Ilustración 17 - Interfaz: Pantalla Histórico de Recomendaciones

En esta pantalla se muestra el histórico de recomendaciones realizada por la herramienta. Permitirá al usuario ver el funcionamiento de las estrategias en casos reales y ver lo que se estaría ganando o perdiendo en caso de haber realizado la operación.

Pantalla Histórico de Recomendaciones de estrategias autoajustadas

FECHA	ACTIVO	ESTRATEGIA	RECOMENDACIÓN	PRECIO	PRECIO ACTUAL	P/G
07-03-2017	Santander	MACD + Bollinger	COMPRA	5,47 €	5,37 €	- 0,10 €
07-03-2017	Santander	MACD + Bollinger Reaj	VENTA	5,47 €	5,37 €	0,10 €

Ilustración 18 - Interfaz: Pantalla Histórico de Recomendaciones de estrategias autoajustadas

En esta pantalla se muestran las recomendaciones realizadas por una estrategia en paralelo con la recomendación realizada por su estrategia análoga autoajustada. Permitirá al usuario comparar entre ambas y ver cual tiene un mejor funcionamiento.

Pantalla de inversiones

FECHA	ACTIVO	ESTRATEGIA	RECOMENDACIÓN	PRECIO	PRECIO ACTUAL	P/G
07-03-2017	Santander	MACD + Bollinger	Compra	5,47 €	5,57 €	0,10 €

Ilustración 19 - Interfaz: Pantalla de inversiones

Desde esta pantalla, el usuario podrá ver cuando quiera todas las órdenes realizadas desde la aplicación. Cuenta con un menú para filtrar los resultados mediante la elección de un activo, una estrategia y/o una fecha concreta.

Respecto a la información mostrada, es fundamental mostrar la fecha en la que se realizó la operación, sobre el activo que se realizó, la estrategia empleada para la toma de la decisión, el sentido de la operación, el número de títulos de la operación, el precio de la operación, el precio actual del activo (precio de cierre último) y las pérdidas o ganancias que generan la diferencia entre el precio de la operación y el precio actual.

Pantalla de configuración

Gestión de Estrategias	Configuración de órdenes	Otras configuraciones
Crear, borrar, modificar y probar estrategias.	Configurar la generación automática de órdenes.	Configuración del resto de elementos de la aplicación
GESTIONAR	CONFIGURAR	GESTIONAR

Ilustración 20 - Interfaz: Pantalla de configuración

La pantalla de configuración es el enlace a tres secciones distintas de elementos que el usuario puede configurar: las estrategias, las órdenes y otras configuraciones.

Pantalla de configuración - Configuración de estrategias

The screenshot shows the 'Gestión de Estrategias' interface. At the top, there is a navigation bar with 'Análisis Técnico', 'Inversiones', and 'Configuración' (selected). The main content area is titled 'Gestión de Estrategias' and contains four panels:

- Crear estrategia:** Includes fields for 'INDICADOR' (two dropdowns), 'PESO', 'NOMBRE', and 'PERIODO' (set to '30 días'). There is an 'AUTOAJUSTE' dropdown (set to 'sí/no') and a 'GUARDAR' button.
- Modificar estrategia:** Starts with an 'ESTRATEGIA' dropdown (set to 'Estrategia'). It then has 'INDICADOR' (two dropdowns) and 'PESO' fields. There is a 'GUARDAR' button.
- Borrar estrategia:** Features an 'Estrategia a borrar:' dropdown (set to 'Estrategia') and a 'BORRAR' button.
- Probar estrategia:** Contains five fields: 'Estrategia' (dropdown), 'Activo' (dropdown), 'Periodo' (dropdown), 'Fecha Inicial', and 'Fecha Final'. There is a 'PROBAR' button.

Ilustración 21 - Interfaz: Pantalla de Configuración de Estrategias

La pantalla de configuración o gestión de estrategias permite realizar 4 acciones sobre las estrategias de la aplicación:

- **Crear estrategias:** se permite al usuario añadir tantos indicadores diferentes como existan en la aplicación y a cada uno se le asocia un peso. Una vez elegidos los indicadores, hay que nombrar la estrategia. Por último, el usuario tiene que decidir si quiere que esta estrategia tenga una estrategia paralela autoajustable y el periodo de autoajuste de la misma. Finalmente hay que habilitar un botón para guardar los datos.
- **Modificar estrategias:** se habilita un desplegable en el que el usuario puede elegir una de las estrategias creadas. Al seleccionarla se cargarán todos los indicadores y sus pesos asociados para su modificación. Se habilita un botón para guardar los cambios.
- **Borrar estrategia:** mediante un desplegable el usuario elige la estrategia a borrar entre las existentes y se habilita un botón para hacer efectivo el borrado de la estrategia.
- **Probar estrategia:** el usuario cuenta con cinco parámetros a rellenar para realizar el backtesting de una estrategia, estos son: la estrategia a probar, el activo del cual se quiere obtener las cotizaciones para realizar la prueba, el periodo o horizonte temporal y la fecha inicial y final para la obtención de las cotizaciones.

Pantalla de configuración - Configuración de la generación de órdenes automáticas

Gestor: XXXXX Cartera: YYYY

Análisis Técnico Inversiones Configuración

Configuración de órdenes

Si ya posee títulos de un activo en su cartera

Porcentaje de títulos a comprar: _____

Porcentaje de títulos a vender: _____

Si no posee títulos de un activo en su cartera

Porcentaje del patrimonio a invertir: _____

GUARDAR

Ilustración 22 - Interfaz: Pantalla de Configuración de las Órdenes Automáticas

Desde esta pantalla de configuración el usuario podrá establecer los porcentajes que crea convenientes para cada caso de generación de órdenes automáticas. Para cada caso se provee de un input de datos y se proporciona un botón para guardar los cambios.

Pantalla de configuración - Otras configuraciones

Esta pantalla se dedica para otro tipo de configuraciones de usuario más genéricas como pueden ser el color de la interfaz u otros elementos que no son fundamentales para el funcionamiento de la aplicación.

Ventana emergente

INFORMACIÓN

MENSAJE

ACEPTAR

Ilustración 23 - Interfaz: Ventana emergente

Las ventanas emergentes se componen de un breve título descriptivo y de un mensaje con la información correspondiente. Contarán con dos formas de cerrarse mediante un botón de aceptar el mensaje y un botón de cerrar la ventana.

4.5 Decisiones de diseño

En este apartado se explican y justifican algunas decisiones de diseño tomadas previa implementación de la herramienta, es decir, referentes a las tecnologías que se van a utilizar para desarrollar la herramienta.

Para la implementación de esta herramienta existen muchas posibilidades y combinaciones posibles. En nuestro caso muchas de las posibilidades existentes se han visto limitadas por propia decisión del cliente en sus requisitos. Aun así, se tratará de justificar cada una de las tecnologías escogidas para desarrollar el proyecto.

Desarrollo de la aplicación web

En cuanto a la tecnología para construir la aplicación web se hace uso de Java Enterprise Edition (JEE) [16] por varias razones, es lo que existe en la empresa del cliente, porque es lo que los desarrolladores conocen en mayor medida, porque es una tecnología muy consolidada y porque hay un gran soporte. Estas razones nos hacen desechar alternativas como puede ser spring [17], una solución más novedosa y posiblemente más potente en posibilidades, pero sin tanto potencial en lo anteriormente comentado.

Servidor de aplicaciones

En lo que al servidor de aplicaciones se refiere, existen diferentes alternativas como pueden ser Tomcat, JBoss o Glassfish. JBoss Y Glashfish son servidores que proveen el acceso a toda la funcionalidad ofrecida por JEE mientras que Tomcat es simplemente un servidor web y un contenedor web o contenedor de servlets. La elección es sencilla, puesto que no se requiere de toda la funcionalidad que ofrece JEE, se va a elegir Tomcat como servidor de aplicaciones ya que es más ligero y menos complejo. Además, permite añadir módulos para hacer uso de la funcionalidad que no trae consigo en caso de necesitarlo [18].

Uso de servicios web

La funcionalidad de la aplicación podría implementarse dentro de los distintos servlets o handlers de la aplicación, pero entonces este código no podría ser reutilizable o habría que replicarlo. Por ello se quiere adoptar una arquitectura SOA dentro de la arquitectura planteada anteriormente en el diseño. Se quiere implementar la funcionalidad de la aplicación en servicios web para que estos puedan ser utilizados no solo por esta aplicación si no por cualquier otra que el cliente tiene y que está basada en otros lenguajes. De esta forma se reutiliza código, se modulariza la aplicación y la funcionalidad fundamental se hace independiente de la infraestructura.



Conexión entre java y r

Para resolver esta problemática de ejecutar código en r desde java, como es el caso de los cálculos de los indicadores, existen dos alternativas: Rjava [19] y Rserve [20]. Ambas son dos librerías cuya diferencia fundamental es el modo de ejecutar. Rserve actúa bajo una arquitectura cliente servidor que hace uso de TCP/IP para la comunicación mientras que rjava se integra con el código java y ejecuta el código r directamente. Rserve es más fácil de instalar y usar, y su arquitectura cliente servidor permite multiples peticiones a la vez, sin embargo, puede generar tráfico de red innecesario según se use. Rjava es más complejo de instalar y configurar (variables de entorno y librerías) y da muchos problemas hasta que se consigue un funcionamiento satisfactorio. Por simplicidad, su carácter multi-hilo y la no importancia de generar tráfico de red, se hará uso de Rserve.

Web Responsive

Para hacer la parte web de la aplicación responsive se hará uso de un framework que facilite esta tarea ya que si no puede hacerse muy tedioso y llevaría mucho más tiempo de implementar. Existen multitud de frameworks para conseguir este propósito como pueden ser Foundation o Boilerplate pero para este proyecto se recomienda utilizar Bootstrap por ser más conocido, más usado y con un mayor soporte, además de la potente funcionalidad ofrecida.

5. Implementación

Una vez realizados el análisis y el diseño de la herramienta podemos pasar a la implementación de la misma. Se enumerarán los distintos elementos que componen la aplicación y se detallará que realiza cada uno de ellos. En el punto 10 del documento se detallará lo que se ha conseguido llevar a cabo de lo aquí descrito haciendo referencia a los requisitos de usuario.

5.1 Obtención de cotizaciones

La aplicación objeto de este proyecto basa su funcionamiento en las cotizaciones de los distintos activos por lo que su obtención es un elemento fundamental. La aplicación hará uso de la base de datos de la aplicación para obtener las cotizaciones durante la ejecución. Estos datos deben ser obtenidos previamente de algún lugar para poder almacenarlos en la base de datos de la aplicación.

Como se comentó en el punto 2.3, en el que se trataba la temática de obtención de datos para la realización del análisis técnico, existen numerosos proveedores de datos en el mercado. Cada uno de estos proveedores tiene unos costes y un conjunto de información diferente, por ello, lo mejor es implementar diferentes soluciones para obtener datos de diferentes proveedores y que el cliente final decida de cuál de ellas quiere obtener los datos.

Para conseguir las cotizaciones y almacenarlas en la base de datos se implementará un programa que se ejecutará cada día a una hora determinada para obtener las cotizaciones mediante la llamada a un web service. El funcionamiento del web service se describirá más adelante.

El web service permitirá obtener cotizaciones del día, pero también se requieren las cotizaciones semanales y mensuales ya que la herramienta permite establecer distintos horizontes temporales. Cada viernes después de cierre de mercado o cada lunes antes de la apertura del mismo deberían calcularse las cotizaciones semanales y almacenarlas en la tabla de cotizaciones. Del mismo modo ocurre con las cotizaciones mensuales. El periodo de una cotización debe indicarse rellenando el campo periodo de la tabla cotizaciones.

5.2 Implementación de la lógica de compraventa en R

Para el proyecto se parte de la implementación de distintos indicadores (los tratados en el apartado 2.1.1). De estos indicadores, lo que se tiene es el cálculo matemático puro por lo que hay que implementar las gráficas y la lógica de compraventa correspondiente a cada indicador. Los scripts que implementan la ejecución de un determinado indicador tienen todos la misma estructura y simplemente varían en la implementación de las gráficas y de la lógica de compraventa. La implementación de estos scripts se realiza como una función en R de forma que esta función pueda ser llamada desde java posteriormente. La implementación consta de los siguientes pasos:

- Cargar las librerías necesarias.
- Llamar a la función que realiza los cálculos matemáticos del indicador pasándole por parámetro los datos necesarios.
- Almacenar los resultados en un data frame.
- Pintar el gráfico correspondiente al indicador y generar una imagen en formato PNG que se almacenará de forma local.
- Implementar la lógica de compraventa en la que, en última instancia, se obtendrá una cadena de caracteres con uno de los siguientes valores: “COMPRA”, “VENTA” o “NEUTRO”.
- Generar el resultado de la llamada a la función que será la cadena de caracteres que contiene el resultado de la lógica de compraventa a la que se le concatenará el valor numérico del indicador calculado, separado por “;”. Por ejemplo: *ret = “COMPRA;23.348”*. Esto se hace así porque se va a llamar a estas funciones desde Java y con las cadenas de caracteres no hay problemas de formatos entre lenguajes, además los datos que se obtienen son solo 2 por lo que tampoco se requiere de una estructura compleja para organizarlos y estandarizarlos en la comunicación.
- Salir de la función devolviendo el resultado.

A continuación, se muestra un ejemplo de cómo se implementarían estos scripts:

```
calcularRSI=function(id, data){
  # Carga de librerías necesarias
  library("zoo")
  library("xts")
  library("TTR")
  library("xlsx")
  library("writexLS")

  #####
  #5. RSI (Relative Strength Index)

  source('C:/Users/xxxx/Desktop/TFM/Material/Indicadores en Codigo R/RSI_AFS.R')

  #Llamar a la función
  dataRSI = data.frame(data,RSI_AFS(data[c('CLOSE')],n=14))

  # Plot RSI
  filepdf = paste("C:/Users/xxxx/Desktop/TFM/Graficas",id,".png",sep = "", collapse = NULL)
  pdf(filepdf, width=8,height=6)
  plot(dataRSI$DATE_TIME,dataRSI$RSI, ylim=c(20, 80))
  text(10,75, "Overbought", col="darkgreen")
  text(10,60, "Buy", col="green3")
  text(10,40, "Sell", col="red")
  text(10,25, "oversell", col="red4")
  lines(dataRSI$RSI, col = 'blue')
  abline(h = 70, col = "green4")
  abline(h = 50, col = "darkorange")
  abline(h = 30, col = "red")

  dev.off()

  #Logica de compraventa
  RSIactual = as.double( dataRSI$RSI[nrow(dataRSI)])
  RSIanterior = as.double( dataRSI$RSI[nrow(dataRSI)-1])
  cat(RSIanterior)
  cat(RSIactual)
  res = ""
  if(RSIanterior < 30 && RSIactual >= 30){
    res = "VENTA"
  }
  if(RSIanterior > 70 && RSIactual <= 70){
    res = "COMPRA"
  }
  if(RSIactual > 50 && RSIactual <= 70){
    res = "COMPRA"
  }
  if(RSIactual < 50 && RSIactual >= 30){
    res = "VENTA"
  }

  res = "NEUTRO"
  res = paste(res,;",", dataRSI$RSI[length(dataRSI$RSI)],sep = "", collapse = NULL)
  return res
}
```

Ilustración 24 - Ejemplo Indicador en R

5.3 Implementación de los Web Services

La aplicación cuenta con cuatro web services que resuelven distintas funcionalidades de la aplicación: uno para la obtención de cotizaciones, otro para los procesos de cálculos (cálculo de indicadores, backtesting y autoajuste), otro para la consulta de datos históricos y otro para el tratamiento de las configuraciones.

5.3.1 Web service para la obtención de cotizaciones

Este web service se encargará de obtener los datos financieros de los activos para realizar los cálculos de los diferentes indicadores técnicos. Los datos necesarios para realizar el análisis técnico son los siguientes:

- Precio de cierre: Es el precio de cierre de mercado de un activo en un día específico.
- Precio máximo de un activo en el día.
- Precio mínimo de un activo en el día.
- Volumen: cantidad de operaciones realizadas sobre un activo en el día.
- Fecha a los que pertenece cada uno de estos datos.

Cada uno de estos datos se obtendrá de alguno de los data vendors existentes en el mercado haciendo uso de su respectiva API. Posteriormente, los datos se almacenarán en la base de datos, en la tabla cotizaciones. El web service recibirá como parámetro el identificador del activo o del mercado y una fecha para descargar las cotizaciones y devolverá como resultado un elemento de feedback que informe si el proceso se ha ejecutado correctamente o no.

5.3.2 Web service para los procesos de cálculo

Este web service se encarga de implementar los distintos procesos de cálculo que tiene la aplicación. Cada uno de estos procesos es accesible a través de diferentes URI's/funciones pertenecientes al web service. A continuación, se describe cada una de las funcionalidades de este web service.

Función para el cálculo de indicadores

Esta función del web service se encarga de realizar los cálculos de los indicadores de análisis técnico. En su llamada, recibirá como parámetros la siguiente información:

- Activo o cartera.
- Estrategia.
- Horizonte temporal: corto plazo, medio plazo o largo plazo.

Lo que hará la función es lo siguiente:

- Obtener de base de datos los indicadores de los que se compone la estrategia y también sus pesos.
- Obtener de base de datos los activos que componen la cartera, en caso de tratar una cartera.
- Establecer la conexión con el servidor de Rserve para poder mandarle las peticiones de ejecución de scripts escritos en R.
- Para cada activo:
 - Obtener las cotizaciones para realizar los cálculos.
 - Calcular cada indicador llamando a la función de R correspondiente, pasándole los datos necesarios para el cálculo.
 - Calcular el resultado a nivel de estrategia en función de los pesos de cada indicador y de la recomendación de cada indicador. Existen muchas formas de realizar esto y todas pueden ser válidas. La planteada en este proyecto es la siguiente:

El cálculo de cada indicador va a dar uno de los siguientes resultados:

- 1, que se corresponde con la acción de vender.
- 2, que se corresponde con un estado neutro de no realizar ninguna acción.
- 3, que se corresponde con la acción de comprar.

Este valor numérico de cada indicador se multiplicará por el peso de cada indicador. Por ejemplo, el indicador 1 nos devuelve venta con un peso de 0,5 y el indicador 2 nos devuelve compra con un peso de 0,5:

$$ResInd1 = 0,5 * 1 = 0,5$$

$$ResInd2 = 0,5 * 3 = 1,5$$

Posteriormente se sumarán todos los valores obtenidos y se obtendrá un valor comprendido entre 1 y 3. Siguiendo el ejemplo:

$$ResEstrat = 0,5 + 1,5 = 2$$

Este conjunto de valores que van desde 1 hasta 3 se divide en tres partes iguales de forma que se dé a cada posible acción las mismas posibilidades:

- Si $ResEstrat \geq 1$ y $ResEstrat < 1,666$: el resultado global de la estrategia será "VENDER".
 - Si $ResEstrat \geq 1,666$ y $ResEstrat < 2,333$: el resultado global de la estrategia será "NEUTRO".
 - Si $ResEstrat \geq 2,333$ y $ResEstrat \leq 3$: el resultado global de la estrategia será "COMPRAR".
- En el cálculo de cada indicador para cada activo se genera una imagen con la gráfica correspondiente al indicador. Todas estas imágenes generadas, se enviarán al manejador que realiza la llamada a este web service a través de un socket, en un hilo que ejecutará de forma concurrente.



- Calculada la estrategia para todos los activos, se compone un JSON con los resultados que deberá contener toda la información generada: activos, indicadores, resultados a nivel de indicador, resultado a nivel de estrategia, etc.
- Devolver el JSON.

Función para realizar el backtesting

Esta función del web service se encarga de realizar el backtesting de las distintas estrategias que el usuario se crea. En su llamada, recibirá como parámetros la siguiente información:

- Activo o cartera.
- Estrategia.
- Horizonte temporal: corto plazo, medio plazo o largo plazo.
- Fecha de inicio: fecha para la que se ejecutarán los distintos indicadores de la estrategia.
- Fecha de fin: fecha de la cotización que se usará para comparar con la cotización de la fecha de inicio.

Lo que hará la función es lo siguiente:

- Para cada activo:
 - Calcular el resultado de la estrategia dada para la fecha de inicio dada.
 - Obtener el precio de cierre de la fecha de inicio.
 - Obtener el precio de cierre de la fecha de fin.
 - Comparar los precios obtenidos en función de la recomendación sugerida por la estrategia. Por ejemplo, si el resultado de la estrategia es compra, suponemos que hemos comprado el activo a precio de fecha de inicio y haremos la diferencia de este precio con el precio de la fecha de fin. Si obtenemos un valor positivo quiere decir que el indicador ha actuado de forma correcta en este periodo y lo opuesto para el caso contrario.
- Componer un JSON con toda la información generada que devolverá el web service para su tratamiento.
- Retornar al JSON.

Función para realizar el autoajuste

Esta función del web service se encarga de realizar el autoajuste de los pesos de las estrategias autoajustables. Esto se puede realizar de múltiples maneras las cuales darían resultados distintos. Para el proyecto se ha elegido una la cual se describe a continuación.

Lo que hará la función es lo siguiente:

- Obtener de base de datos todos los autoajustes a realizar en la fecha actual de la tabla autoajuste.
- Para cada autoajuste:
 - Obtener de base de datos los indicadores y pesos de los que se compone la estrategia utilizada en la recomendación. Si durante el proceso hay varios ajustes para la misma estrategia, se hará uso de los datos que hay en memoria y que contienen la última actualización en la distribución de pesos.
 - Obtener de base de datos la recomendación de compraventa de cada indicador para la fecha en la que se realizó la recomendación.
 - Obtener las cotizaciones del día en el que se realizó la recomendación y el día actual.
 - Calcular la diferencia de las cotizaciones y establecer en uno de los tres niveles de compraventa lo que realmente ha sucedido:
 - Si el precio ha caído más de un 2%, la recomendación adecuada hubiera sido “VENTA”.
 - Si el precio ha subido más de un 2%, la recomendación adecuada hubiera sido “COMPRA”.
 - En cualquier otro caso, la recomendación adecuada hubiera sido “NEUTRO”.
 - Para cada indicador, ver si ha acertado, fallado en un nivel o fallado en dos niveles.
 - A los indicadores que han fallado en dos niveles, quitarle un 2% del peso. A los indicadores que han fallado en un nivel, quitarle un 1% del peso.
 - El peso restado se distribuirá de la siguiente forma:
 - Si hay más de un indicador que ha acertado, se repartirá de forma alícuota.
 - Si sólo un indicador ha acertado, se le sumará todo a este.
 - Si no hay indicadores que han acertado, se repartirá de forma alícuota entre aquellos que solo han fallado en un nivel.
 - En su defecto, se repartirá de forma alícuota entre todos los indicadores.
- Componer un JSON con la nueva distribución de pesos de cada estrategia ajustada.
- Retornar el JSON.

5.3.3 Web service para consulta de históricos

Función para resolver el histórico de inversiones

Esta funcionalidad se encargará de obtener los datos históricos de las inversiones realizadas desde la aplicación y monitorizar si se está ganando o perdiendo con una inversión realizada. Para ello, deberá realizar las siguientes acciones:

- Obtener el último precio de cierre de cada activo sobre el que se haya realizado una inversión, aplicando los filtros de búsqueda si corresponde.
- Obtener de base de datos el histórico de operaciones realizadas, aplicando los filtros de búsqueda si corresponde.
- Calcular la diferencia entre el precio actual del activo y el precio o los precios al que se realizó la operación para ver cuánto se ha ganado o perdido.
- Componer un JSON con toda la información generada: activo, estrategia usada en la operación, precio de compra, precio actual, diferencia, cantidad, sentido y fecha.
- Retornar al JSON.

Función para resolver el histórico de recomendaciones

Esta funcionalidad se encargará de obtener los datos históricos de las recomendaciones realizadas desde la aplicación. Para ello, deberá realizar las siguientes acciones:

- Obtener el último precio de cierre de cada activo sobre el que se haya realizado una recomendación, aplicando los filtros de búsqueda si corresponde.
- Obtener de base de datos el histórico de recomendaciones realizadas, aplicando los filtros de búsqueda si corresponde.
- Obtener de base de datos el precio de los activos para cada fecha de recomendación.
- Calcular la diferencia entre el precio actual del activo y el precio al que se realizó la operación para ver de forma resumida cuánto estaría ganando o perdiendo.
- Componer un JSON con toda la información generada: activo, estrategia, precio en la fecha de la recomendación, precio actual, diferencia, recomendación y fecha.
- Retornar al JSON.

Función para resolver la visualización del funcionamiento de las estrategias autoajustadas

Esta funcionalidad se encargará de obtener los datos históricos de las recomendaciones realizadas desde la aplicación de las estrategias originales y las autoajustadas de forma que se puedan comparar visualmente. Para ello, deberá realizar las siguientes acciones:

- Obtener el último precio de cierre de cada activo sobre el que se haya realizado una recomendación, aplicando los filtros de búsqueda si corresponde.
- Obtener de base de datos el histórico de recomendaciones realizadas de aquellas estrategias que sean autoajustables, trayendo los datos de las estrategias originales como de las autoajustadas, aplicando los filtros de búsqueda si corresponde.
- Obtener de base de datos el precio de los activos para cada fecha de recomendación.
- Calcular la diferencia entre el precio actual del activo y el precio al que se realizó la recomendación, para ver de forma resumida cuánto estaría ganando o perdiendo.
- Componer un JSON con toda la información generada: activo, estrategia, precio en la fecha de la recomendación, precio actual, diferencia, recomendación y fecha.
- Retornar al JSON.

5.3.4 Web service para el tratamiento de las configuraciones

Este web service se encarga de tratar todo lo relacionado con las configuraciones que puede realizar el usuario sobre la aplicación. Para ello se implementan distintas funcionalidades haciendo uso de diferentes URI's pertenecientes al web service. A continuación, se describe cada una de las funcionalidades de este web service.

Función para la creación de estrategia

Esta función se encargará de guardar en base de datos las estrategias que el usuario se vaya creando en la aplicación. Para ello, deberá realizar las siguientes acciones:

- Almacenar en las tablas estrategia y estrategiaDef los datos correspondientes.
- Retornar el resultado de la operación.



Función para el borrado de una estrategia

Esta función se encargará de borrar de la base de datos las estrategias que el usuario se haya creado en la aplicación. Para ello, deberá realizar las siguientes acciones:

- Si existe, borrar de la tabla estrategia la estrategia correspondiente.
- Retornar el resultado de la operación.

Función modificación de estrategias

Esta función se encargará de actualizar los datos existentes en base de datos de las estrategias que el usuario se haya creado en la aplicación. Para ello, deberá realizar las siguientes acciones:

- Si existe la estrategia, actualizar en las tablas estrategia y estrategiaDef los datos correspondientes.
- Retornar el resultado de la operación.

Función configuración de órdenes

Esta función se encargará de guardar en base de la configuración establecida por el usuario para para la generación automática de órdenes tras una recomendación. Para ello, deberá realizar las siguientes acciones:

- Almacenar en la tabla ordenesParam los datos de la configuración introducida por el usuario.
- Retornar el resultado de la operación.

5.4 Implementación del servlet

Para comunicar la capa de presentación con la lógica de negocio, todas las acciones que el usuario quiera realizar llegarán a un punto de entrada único el cual será un servlet. Este servlet, a su vez, se encargará de redirigir las peticiones http al handler correspondiente, el cual tratará y resolverá la petición.

El servlet, por tanto, será una clase java que extiende de *HttpServlet* y que implementa los siguientes métodos:

- *init()*: método que se encarga de mapear cada acción que puede realizar el usuario en la interfaz con los distintos handlers que se encargarán de resolver la tarea que el usuario desea realizar.
- *doPost()*: método que se ejecuta cuando llegue una petición POST desde los JSP, normalmente cuando se quiere procesar la petición y los datos del usuario que trae consigo.

Lo que hará este método fundamentalmente es realizar las redirecciones referentes a la navegabilidad dentro de la web y devolver el resultado de la petición al usuario. Si se trata de una redirección en la que simplemente se va a pasar de una vista a otra sin hacer tratamiento de datos, el mismo servlet se encargará de resolverlo. Si la petición conlleva el tratamiento de datos o conlleva la realización de alguna tarea, el servlet delegará en el handler correspondiente redirigiendo la petición.

Finalmente, haciendo uso del *RequestDispatcher* y del método *forward* se devolverá un jsp con la petición y la respuesta que se originó, pero ya tratada.

- *doGet()*: método que redirige al método *doPost*.

5.5 Implementación de los handlers

Los handlers son clases que extienden de *RequestHandler* e implementan de forma obligatoria el método `handleRequest()` para tratar las peticiones que les son redirigidas. Normalmente, dentro de este método se hace la lógica de negocio correspondiente a la acción que desea realizar el usuario. Los distintos manejadores que tiene la aplicación son los siguientes:

Manejador de inicio de sesión:

Este manejador será el que se encarga de gestionar el inicio de sesión en la aplicación. Para ello, deberá realizar las siguientes acciones:

- Comprobar que los datos de entrada son correctos.
- Llamar al servicio web que realiza la autenticación del usuario y comprueba que la cartera a gestionar indicada existe y pertenece a ese usuario. Este servicio web ya existe y se hará uso de él.
- Comprobar el resultado de la llamada al web service.
- Introducir en los datos de sesión el usuario y el identificador de la cartera con la que se está para que puedan ser utilizados posteriormente.
- Retornar al servlet.

Manejador del cálculo de indicadores

Este manejador será el que se encarga de gestionar la petición de cálculo de indicadores. Para ello, deberá realizar las siguientes acciones:

- Comprobar que los datos de entrada son correctos.
- Lanzar un hilo que implementará un socket por el cual se recibirán las imágenes de los indicadores calculados y las almacenará en el sistema de ficheros.
- Llamar al servicio web que realiza el cálculo de indicadores pasándole como parámetros el activo o cartera, la estrategia o indicadores recibidos y el horizonte temporal.
- Se recibirá del web service un JSON que habrá que interpretar.
- Guardar en la base de datos los resultados de la ejecución en las tablas `historicoRecomendaciones` e `historicoIndicadores`.
- Introducir en la request todos los datos a mostrar para que puedan ser leídos desde el JSP.
- Retornar al servlet.

Manejador del envío de órdenes

Este manejador se encargará de enviar las órdenes generadas automáticamente y que finalmente el usuario ha decidido ejecutar al bróker que el usuario tenga configurado. Para ello, deberá realizar las siguientes acciones:

- Comprobar que los datos de entrada son correctos.
- El envío se realiza llamando a un web service que ya existe por lo que se realiza la llamada al servicio web existente, pasándole los parámetros correspondientes. Para ello el usuario debe tener configurado previamente un bróker al que enviar las órdenes. Esta configuración se establece en la aplicación de front-office en la que se integra esta aplicación.
- Se recibirá del web service el resultado de la operación.
- Introducir en la request todos los datos a mostrar para que puedan ser leídos desde el JSP.
- Retornar al servlet.

Manejador de consulta de inversiones

Este manejador se encargará de obtener los datos históricos a mostrar de las inversiones realizadas desde la aplicación y calcular si se está ganando o perdiendo con una inversión realizada. Para ello, deberá realizar las siguientes acciones:

- Comprobar que los datos de entrada son correctos.
- Llamar al servicio web que realiza el proceso de obtención del histórico de inversiones pasándole como parámetros los posibles filtros que haya introducido el usuario.
- Se recibirá del web service un JSON que habrá que interpretar.
- Introducir en la request todos los datos a mostrar para que puedan ser leídos desde el JSP.
- Retornar al servlet.

Manejador de histórico de recomendaciones

Este manejador se encargará de obtener los datos históricos a mostrar de las recomendaciones realizadas desde la aplicación. Para ello, deberá realizar las siguientes acciones:

- Comprobar que los datos de entrada son correctos.
- Llamar al servicio web que realiza el proceso de obtención del histórico de recomendaciones pasándole como parámetros los posibles filtros que haya introducido el usuario.
- Se recibirá del web service un JSON que habrá que interpretar.



- Introducir en la request todos los datos a mostrar para que puedan ser leídos desde el JSP.
- Retornar al servlet.

Manejador de consulta del funcionamiento de las estrategias autoajustadas

Este manejador se encargará de obtener los datos a mostrar de las recomendaciones realizadas por las estrategias originales del inversor y las autoajustadas, para ver si la estrategia autoajustada está dando mejor resultado que la original. Para ello, deberá realizar las siguientes acciones:

- Comprobar que los datos de entrada son correctos.
- Llamar al servicio web que realiza el proceso de obtención de los datos necesarios, pasándole como parámetros los posibles filtros que haya introducido el usuario.
- Se recibirá del web service un JSON que habrá que interpretar.
- Actualizar los datos de las estrategias autoajustadas en base de datos.
- Introducir en la request todos los datos a mostrar para que puedan ser leídos desde el JSP.
- Retornar al servlet.

Manejador creación de estrategia

Este manejador se encargará de validar y guardar en base de datos las estrategias que el usuario se vaya creando en la aplicación. Para ello, deberá realizar las siguientes acciones:

- Comprobar que los datos de entrada son correctos.
- Llamar al servicio web creado para el tratamiento de las configuraciones, pasándole como parámetros los datos provenientes del formulario de creación de estrategias.
- Tratar el resultado proporcionado por el servicio web.
- Retornar al servlet.

Manejador borrado de estrategia

Este manejador se encargará de validar y borrar de la base de datos las estrategias que el usuario se haya creado en la aplicación. Para ello, deberá realizar las siguientes acciones:

- Comprobar que los datos de entrada son correctos.
- Llamar al servicio web creado para el tratamiento de las configuraciones, pasándole como parámetros los datos provenientes del formulario de borrado de estrategias.

- Tratar el resultado proporcionado por el servicio web.
- Retornar al servlet.

Manejador modificación de estrategias

Este manejador se encargará de validar y actualizar los datos existentes en base de datos de las estrategias que el usuario se haya creado en la aplicación. Para ello, deberá realizar las siguientes acciones:

- Comprobar que los datos de entrada son correctos.
- Llamar al servicio web creado para el tratamiento de las configuraciones, pasándole como parámetros los datos provenientes del formulario de modificación de estrategias.
- Tratar el resultado proporcionado por el servicio web.
- Retornar al servlet.

Manejador backtesting

Este manejador será el que se encarga de gestionar las pruebas que el usuario quiera realizar sobre una estrategia para comprobar su comportamiento en situaciones del pasado. Para ello, deberá realizar las siguientes acciones:

- Comprobar que los datos de entrada son correctos.
- Llamar al servicio web que realiza el proceso de backtesting pasándole como parámetros el activo o cartera, la estrategia o indicadores recibidos y las fechas de inicio y de fin.
- Se recibirá del web service un JSON que contiene los distintos datos del resultado generado por el servicio web.
- Interpretar el resultado obtenido
- Guardar en la base de datos los resultados de la ejecución en la tabla de backtesting.
- Introducir en la request los resultados a mostrar para que puedan ser leídos desde el JSP.
- Retornar al servlet.

Manejador configuración de órdenes

Este manejador se encargará de validar y guardar en base de la configuración establecida por el usuario para para la generación automática de órdenes tras una recomendación. Para ello, deberá realizar las siguientes acciones:

- Comprobar que los datos de entrada son correctos.



- Llamar al servicio web creado para el tratamiento de las configuraciones, pasándole como parámetros los datos provenientes del formulario de configuración de las órdenes automáticas.
- Tratar el resultado proporcionado por el servicio web.
- Retornar al servlet.



5.6 Implementación de la interfaz

Para implementar la interfaz se hará uso de Bootstrap, un framework de HTML, CSS y Javascript que permite crear webs *responsives* o adaptables. Este framework facilitará bastante la programación web la cual se realizará implementando distintos ficheros JSP.

En el punto 4.4 se trató el diseño de la interfaz en el cual se tiene una referencia de las pantallas que deben ser creadas y el contenido de las mismas.

6. Pruebas

Una vez finalizada la implementación de la herramienta es imprescindible verificar el correcto funcionamiento de la misma y ver que satisface las necesidades del cliente. Para asegurarse de esto es necesario realizar un conjunto de pruebas suficientes como para afirmar que la herramienta funciona correctamente. Para conseguir esto haremos uso de pruebas unitarias y de pruebas de integración.

6.1 Pruebas unitarias

Las pruebas unitarias son aquellas que permiten probar el correcto funcionamiento de una parte específica del software. Para describir cada una de las pruebas realizadas se utilizará una tabla como la siguiente:

Identificador: PR - XX	
Nombre:	Nombre de la prueba
Objetivo:	Objetivo de la prueba
Procedimiento:	Pasos para realizar la prueba.
Resultado esperado:	Resultado esperado

A continuación, se muestran todas las pruebas realizadas:

Identificador: PR - 01	
Nombre:	Obtención de cotizaciones
Objetivo:	Verificar que la descarga de cotizaciones se realiza correctamente y que estas se almacenan en base de datos.
Procedimiento:	Escribir un programa que llame al web service de obtención de cotizaciones pasándole como parámetro un activo cualquiera que cotice en el Ibex-35 y una fecha determinada. Imprimir en un log las trazas correspondientes.
Resultado esperado:	El web service descarga y almacena en base de datos los datos correspondientes al activo y fecha dados.



Identificador: PR - 02	
Nombre:	Obtención de cotizaciones programada
Objetivo:	Verificar que el proceso encargado de las descargas de cotizaciones de forma automática realiza la descarga los días indicados y a la hora indicada. Verificar también que se calculan las cotizaciones semanales y mensuales en el día y hora programados.
Procedimiento:	Ejecutar el proceso que gestiona la descarga automática de cotizaciones y mantenerlo en ejecución durante una semana. Imprimir en un log las trazas correspondientes.
Resultado esperado:	<ul style="list-style-type: none">▪ Se ha llamado al web service una vez por día a la hora establecida.▪ Se ha calculado la cotización semanal en el día y fecha establecido.

Identificador: PR - 03	
Nombre:	Inicio de sesión
Objetivo:	Verificar que el proceso de inicio de sesión funciona correctamente.
Procedimiento:	Escribir un programa que haga uso del manejador de inicio de sesión. Imprimir en un log las trazas correspondientes.
Resultado esperado:	La verificación del usuario y contraseña ha sido satisfactoria. La cartera a gestionar existe.

Identificador: PR - 04	
Nombre:	Cálculo de indicadores
Objetivo:	Verificar que cada uno de los indicadores técnicos realizan los cálculos correspondientes de forma correcta, obteniendo los resultados correctos.
Procedimiento:	Ejecutar cada uno de los distintos scripts que calculan indicadores. Comparar los resultados con otras fuentes. Imprimir en un log las trazas correspondientes.
Resultado esperado:	Los cálculos se han realizado de forma correcta.



Identificador: PR - 05	
Nombre:	Gráficos de los indicadores
Objetivo:	Verificar que los gráficos que genera cada indicador se pintan correctamente y se obtiene un fichero PNG con el contenido del mismo.
Procedimiento:	Ejecutar cada uno de los scripts programados en R que generan una gráfica para cada indicador. Imprimir en un log las trazas correspondientes.
Resultado esperado:	Se generan las gráficas de forma correcta y en formato PNG.

Identificador: PR - 06	
Nombre:	Recomendaciones de compraventa a nivel de indicador
Objetivo:	Verificar que cada indicador tiene implementada la lógica de compraventa correctamente.
Procedimiento:	Ejecutar cada uno de los scripts programados en R que realizan la recomendación de compraventa para un conjunto de datos determinado. Imprimir en un log las trazas correspondientes.
Resultado esperado:	Se obtiene una recomendación de compraventa correcta, es decir, que la lógica de compraventa del indicador se corresponde con la que sugiere el valor numérico obtenido, atendiendo a la teoría matemática del indicador.

Identificador: PR - 07	
Nombre:	Recomendaciones de compraventa a nivel de estrategia
Objetivo:	Verificar que la recomendación final de compraventa, es decir la recomendación a nivel de estrategia, se hace correctamente.
Procedimiento:	Escribir un programa que contenga el código que realiza la recomendación de compraventa a nivel de estrategia predefiniendo un vector de pesos y un vector de recomendaciones.
Resultado esperado:	Los cálculos devuelven un valor dentro del conjunto [1,3] y se obtiene la recomendación de compraventa asociada a ese valor.



Identificador: PR - 08	
Nombre:	Backtesting de estrategias
Objetivo:	Verificar que la funcionalidad de backtesting funciona correctamente.
Procedimiento:	Escribir un programa que llame a la función de backtesting pasándole unos parámetros determinados.
Resultado esperado:	Se ha obtenido una recomendación de compraventa y un valor en euros de lo que habría ganado o perdido para las fechas dadas y para cada activo.

Identificador: PR - 09	
Nombre:	Histórico de inversiones
Objetivo:	Verificar que la funcionalidad que implementa la obtención y cálculo de los datos correspondientes a las inversiones realizadas se realiza correctamente.
Procedimiento:	Escribir un programa que llame a la función que resuelve el histórico de inversiones.
Resultado esperado:	Se ha recibido un JSON que contiene para cada operación la siguiente información: activo, estrategia usada en la operación, precio de compra, precio actual, diferencia, cantidad, sentido y fecha.

Identificador: PR - 10	
Nombre:	Creación de estrategias
Objetivo:	Verificar que la funcionalidad de creación de estrategias está correctamente implementada.
Procedimiento:	Escribir un programa que llame a la función de creación de estrategias pasándole los parámetros necesarios.
Resultado esperado:	Se ha almacenado la información en base de datos de forma correcta.

Identificador: PR - 11	
Nombre:	Borrado de estrategias
Objetivo:	Verificar que la funcionalidad de borrado de estrategias está correctamente implementada.
Procedimiento:	Escribir un programa que llame a la función de borrado de estrategias pasándole los parámetros necesarios.
Resultado esperado:	Se ha borrado la estrategia de base de datos.



Identificador: PR - 12	
Nombre:	Modificación de estrategias
Objetivo:	Verificar que la funcionalidad de modificación de estrategias está correctamente implementada.
Procedimiento:	Escribir un programa que llame a la función de modificación de estrategias pasándole los parámetros necesarios.
Resultado esperado:	Se ha modificado la información en base de datos de forma correcta.

Identificador: PR - 13	
Nombre:	Configuración de órdenes
Objetivo:	Verificar que la funcionalidad configuración de órdenes está correctamente implementada.
Procedimiento:	Escribir un programa que llame a la función de configuración de órdenes pasándole los parámetros necesarios.
Resultado esperado:	Se ha almacenado la información en base de datos de forma correcta.

Identificador: PR - 14	
Nombre:	Histórico de recomendaciones
Objetivo:	Verificar que la funcionalidad que implementa la obtención de los datos correspondientes a las recomendaciones realizadas se realiza correctamente.
Procedimiento:	Escribir un programa que llame a la función que resuelve el histórico de recomendaciones.
Resultado esperado:	Se ha recibido un JSON que contiene para cada recomendación la siguiente información: activo, estrategia, precio en fecha de recomendación, precio actual, diferencia, recomendación y fecha.

Identificador: PR - 15	
Nombre:	Autoajuste
Objetivo:	Verificar que el proceso de autoajuste se realiza correctamente.
Procedimiento:	Escribir un programa que llame al web service que contiene la función que implementa el autoajuste. Imprimir en un log las trazas correspondientes.
Resultado esperado:	El web service genera un JSON con la nueva distribución de pesos de cada estrategia autoajustable.



Identificador: PR - 16	
Nombre:	Funcionamiento de estrategias autoajustadas
Objetivo:	Verificar que la funcionalidad que implementa esta funcionalidad se realiza correctamente.
Procedimiento:	Escribir un programa que llame a la función que resuelve la obtención de los datos para ver el funcionamiento de estrategias autoajustadas
Resultado esperado:	Se ha recibido un JSON que contiene para cada recomendación (original y su análoga autoajustada) la siguiente información: activo, estrategia, precio en fecha de recomendación, precio actual, diferencia, recomendación y fecha.

6.2 Pruebas de integración

Una vez se superan las pruebas unitarias, es necesario probar el software en el conjunto de todas sus partes, como una sola unidad. Con este fin se realizan las pruebas de integración. Para describir cada una de las pruebas realizadas se utilizará una tabla como la siguiente:

Identificador: PRI - XX	
Nombre:	Nombre de la prueba.
Procedimiento:	Pasos para realizar la prueba.
Precondiciones:	Requisitos previos a la realización de la prueba.
Resultado esperado:	Resultado esperado.

A continuación, se describen las pruebas de integración:

Identificador: PRI - 01	
Nombre:	Iniciar sesión en la aplicación
Precondiciones:	Ninguna.
Procedimiento:	Para la realización de la prueba hay que llevar a cabo los siguientes pasos: <ul style="list-style-type: none">• Entrar a la aplicación desde un navegador compatible.• En la pantalla de inicio de sesión introducir los datos de acceso.• Darle al botón iniciar sesión.
Resultado esperado:	La petición llegará al servlet, este delegará en el handler de inicio de sesión que ejecutará el código que contiene. Se retornará al servlet y se genera la respuesta para el usuario. Si todo ha ido bien y los datos del usuario eran correctos se le redirigirá a la pantalla de recomendación.



Identificador: PRI - 02	
Nombre:	Realizar una recomendación de compraventa
Precondiciones:	<ul style="list-style-type: none">• Haber iniciado sesión en la aplicación.• Opcionalmente, tener creada una estrategia.
Procedimiento:	Para la realización de la prueba hay que llevar a cabo los siguientes pasos: <ul style="list-style-type: none">• Ir a la pantalla de “Análisis Técnico”.• Darle al botón “Acceder” correspondiente al apartado de análisis técnico.• Rellenar los datos de entrada: “Activo”, “Estrategia” y “Horizonte temporal”.• Darle al botón “CALCULAR”.
Resultado esperado:	La petición llegará al servlet, este delegará en el handler de cálculo de indicadores que ejecutará el código que contiene. Se retornará al servlet y se genera la respuesta para el usuario. Si todo ha ido bien, se le mostrará al usuario el resumen general de las recomendaciones generadas por la aplicación.

Identificador: PRI - 03	
Nombre:	Enviar un orden a un bróker
Precondiciones:	<ul style="list-style-type: none">• Tener un bróker configurado y activo en la aplicación de front-office.• Haber iniciado sesión en la aplicación.
Procedimiento:	Para la realización de la prueba hay que llevar a cabo los siguientes pasos: <ul style="list-style-type: none">• Ir a la pantalla de “Análisis Técnico”.• Darle al botón “Acceder” correspondiente al apartado de análisis técnico.• Hacer una petición de recomendación.• Darle al botón “Enviar orden” de una de las recomendaciones recibidas.
Resultado esperado:	La petición llegará al servlet, este delegará en el handler de envío de órdenes que ejecutará el código que contiene. Se retornará al servlet y se genera la respuesta para el usuario. Si todo ha ido bien, se le mostrará una ventana emergente con información.

Identificador: PRI - 04	
Nombre:	Consultar el histórico de inversiones
Precondiciones:	<ul style="list-style-type: none">• Haber iniciado sesión en la aplicación.• Haber ejecutado una orden desde la aplicación con anterioridad.• Opcionalmente, tener creada una estrategia.
Procedimiento:	Para la realización de la prueba hay que llevar a cabo los siguientes pasos: <ul style="list-style-type: none">• Ir a la pantalla de "Inversiones".• Rellenar, opcionalmente, los datos de entrada: "Activo", "Estrategia" y "Fecha"• Darle al botón "CONSULTAR".
Resultado esperado:	La petición llegará al servlet, este delegará en el handler de histórico de inversiones, que ejecutará el código que contiene. Se retornará al servlet y se genera la respuesta para el usuario. Si todo ha ido bien, se le mostrará al usuario el histórico de inversiones correspondiente, en función de los filtros que haya puesto.

Identificador: PRI - 05	
Nombre:	Crear estrategias
Precondiciones:	<ul style="list-style-type: none">• Haber iniciado sesión en la aplicación.
Procedimiento:	Para la realización de la prueba hay que llevar a cabo los siguientes pasos: <ul style="list-style-type: none">• Ir a la pantalla de "Configuración".• Darle al botón "Gestionar" correspondiente al apartado Gestión de estrategias.• En el apartado "Crear estrategia", rellenar los campos existentes.• Darle al botón "Guardar".
Resultado esperado:	La petición llegará al servlet, este delegará en el handler de creación de estrategias, que ejecutará el código que contiene. Se retornará al servlet y se genera la respuesta para el usuario. Si todo ha ido bien, la información se encontrará en base de datos y el usuario podrá seleccionar esta estrategia en las distintas funcionalidades de la aplicación.

Identificador: PRI - 06	
Nombre:	Borrar estrategias
Precondiciones:	<ul style="list-style-type: none">• Haber iniciado sesión en la aplicación.• Tener una estrategia creada.
Procedimiento:	Para la realización de la prueba hay que llevar a cabo los siguientes pasos: <ul style="list-style-type: none">• Ir a la pantalla de “Configuración”.• Darle al botón “Gestionar” correspondiente al apartado Gestión de estrategias.• En el apartado “Borrar estrategia”, rellenar los campos existentes.• Darle al botón “Guardar”.
Resultado esperado:	La petición llegará al servlet, este delegará en el handler de borrado de estrategias, que ejecutará el código que contiene. Se retornará al servlet y se genera la respuesta para el usuario. Si todo ha ido bien, la información se habrá eliminado de base de datos y el usuario ya no podrá seleccionar la estrategia borrada en las distintas funcionalidades de la aplicación.

Identificador: PRI - 07	
Nombre:	Modificar estrategias
Precondiciones:	<ul style="list-style-type: none">• Haber iniciado sesión en la aplicación.• Tener una estrategia creada.
Procedimiento:	Para la realización de la prueba hay que llevar a cabo los siguientes pasos: <ul style="list-style-type: none">• Ir a la pantalla de “Configuración”.• Darle al botón “Gestionar” correspondiente al apartado Gestión de estrategias.• En el apartado “Modificar estrategia”, rellenar los campos existentes.• Darle al botón “Guardar”.
Resultado esperado:	La petición llegará al servlet, este delegará en el handler de modificación de estrategias, que ejecutará el código que contiene. Se retornará al servlet y se genera la respuesta para el usuario. Si todo ha ido bien, la información que había en la base de datos se habrá actualizado.

Identificador: PRI - 08	
Nombre:	Backtesting de estrategias
Precondiciones:	<ul style="list-style-type: none">• Haber iniciado sesión en la aplicación.• Tener creada una estrategia.
Procedimiento:	Para la realización de la prueba hay que llevar a cabo los siguientes pasos: <ul style="list-style-type: none">• Ir a la pantalla de “Configuración”.• Darle al botón “Gestionar” correspondiente al apartado Gestión de estrategias.• En el apartado “Probar estrategia”, rellenar los campos existentes.• Darle al botón “Probar”.
Resultado esperado:	La petición llegará al servlet, este delegará en el handler de backtesting, que ejecutará el código que contiene. Se retornará al servlet y se genera la respuesta para el usuario. Si todo ha ido bien, se le mostrará una ventana emergente al usuario con el resultado del proceso.

Identificador: PRI - 09	
Nombre:	Configuración de órdenes
Precondiciones:	<ul style="list-style-type: none">• Haber iniciado sesión en la aplicación.
Procedimiento:	Para la realización de la prueba hay que llevar a cabo los siguientes pasos: <ul style="list-style-type: none">• Ir a la pantalla de “Configuración”.• Darle al botón “Gestionar” correspondiente al apartado “Configuración de órdenes”.• Rellenar los campos existentes.• Darle al botón “Guardar”.
Resultado esperado:	La petición llegará al servlet, este delegará en el handler de configuración de órdenes, que ejecutará el código que contiene. Se retornará al servlet y se genera la respuesta para el usuario. Si todo ha ido bien, la información se encontrará en base de datos y el usuario comenzará a recibir recomendaciones acordes con los parámetros que ha establecido.

Identificador: PRI - 10	
Nombre:	Consultar el histórico de recomendaciones
Precondiciones:	<ul style="list-style-type: none">• Haber iniciado sesión en la aplicación.• Haber ejecutado una recomendación desde la aplicación con anterioridad.• Opcionalmente, tener creada una estrategia.
Procedimiento:	Para la realización de la prueba hay que llevar a cabo los siguientes pasos: <ul style="list-style-type: none">• Ir a la pantalla de “Análisis Técnico”.• Darle al botón “Acceder” correspondiente al apartado de histórico de recomendaciones.• Rellenar, opcionalmente, los datos de entrada: “Activo”, “Estrategia” y “Fecha”• Darle al botón “CONSULTAR”.
Resultado esperado:	La petición llegará al servlet, este delegará en el handler de histórico de recomendaciones, que ejecutará el código que contiene. Se retornará al servlet y se genera la respuesta para el usuario. Si todo ha ido bien, se le mostrará al usuario el histórico de inversiones correspondiente, en función de los filtros que haya puesto.

Identificador: PRI - 11	
Nombre:	Consultar el funcionamiento de las estrategias autoajustables
Precondiciones:	<ul style="list-style-type: none">• Haber iniciado sesión en la aplicación.• Tener creada una estrategia con la opción de autoajustable activada.• Haber ejecutado una orden desde la aplicación con anterioridad.
Procedimiento:	Para la realización de la prueba hay que llevar a cabo los siguientes pasos: <ul style="list-style-type: none">• Ir a la pantalla de “Análisis Técnico”.• Darle al botón “Acceder” correspondiente al apartado estrategias reajustadas.• Rellenar, opcionalmente, los datos de entrada: “Activo”, “Estrategia” y “Fecha”• Darle al botón “CONSULTAR”.
Resultado esperado:	La petición llegará al servlet, este delegará en el handler de consulta del funcionamiento de las estrategias autoajustadas, que ejecutará el código que contiene. Se retornará al servlet y se genera la respuesta para el usuario. Si todo ha ido bien, se le mostrará al usuario el histórico de inversiones correspondiente, en función de los filtros que haya puesto.

7. Planificación

En este apartado se incluyen la planificación inicial del trabajo fin de máster, la planificación real (final) del trabajo fin de máster y una posible planificación del proyecto real con un límite de duración de 7 meses. También se incluye una breve descripción de la metodología utilizada para realizar el trabajo fin de máster.

En lo referente a las planificaciones realizadas sobre el TFM, comentar que la planificación real dista un poco de la inicial, el motivo principal es la inexperiencia a la hora de planificar y establecer periodos de tiempo en este tipo de proyectos.

Metodología de trabajo aplicada en el TFM

Para la realización del presente trabajo fin de máster se ha seguido una filosofía similar a la metodología de trabajo en cascada. Esta metodología es una de las metodologías más básicas y clásicas existentes. Esta metodología define y ordena las distintas etapas del proceso de desarrollo de software de forma que, para poder iniciar una etapa, la anterior debe haberse completado en su totalidad. Fundamentalmente se ha utilizado esta metodología por ser un trabajo individual y por ser la más conocida.

En lo que al desarrollo verdadero del proyecto se refiere, es posible que una metodología de desarrollo ágil como pueda ser SCRUM fuera más factible puesto que es un proyecto de una duración media el cual es bastante susceptible a cambios por parte del cliente u otros tipos de cambios como puedan ser cambios regulatorios.

Planificaciones

A continuación, se muestran la planificación inicial y final del TFM y la planificación real del proyecto con sus correspondientes diagramas de Gantt:

Planificación Inicial Del Trabajo Fin de Máster

Actividad	Fecha de inicio	Duración	Fecha de finalización	Justificación
Propuesta del proyecto	23/01/2017	4	26/01/2017	Debate y discusión del proyecto.
Estado del arte	01/02/2017	37	09/03/2017	Análisis e investigación de la situación actual. Uno de los núcleos del proyecto por eso se le dedica algo más de un mes de tiempo.
Análisis del sistema	13/03/2017	12	24/03/2017	Análisis del sistema, definición de las características básicas que tendrá el sistema. Incluyen conversaciones con el cliente. Con la experiencia existente para la elaboración de esta parte con dos semanas debería ser suficiente.
Diseño del sistema	27/03/2017	19	14/04/2017	Diseño del sistema. Incluye investigación sobre diversas alternativas posibles y el diseño completo del sistema. Existe experiencia previa en el diseño de base de datos y prototipado de interfaces por lo que tres semanas es un plazo razonable.
Implementación	17/04/2017	44	30/05/2017	Consiste en realizar un pequeño prototipo y en base a ello tratar de documentar la implementación del proyecto completo. Es la parte más que lleva más tiempo en el proyecto puesto que conlleva implementar.
Base de datos	17/04/2017	3	19/04/2017	Creación de tablas y carga de datos.
Indicadores en código R	20/04/2017	19	08/05/2017	Incluye estudio y aprendizaje de R, estudio de los indicadores proporcionados, implementación de la lógica de compraventa y creación de gráficas.
Integrar R y JAVA	09/05/2017	2	10/05/2017	Proceso de integrar Java y R mediante una librería intermediaria.
Web Services	11/05/2017	20	30/05/2017	Desarrollo de la parte básica de la aplicación como son la funcionalidad de la recomendación de compraventa o el backtesting.
Pruebas	17/04/2017	44	30/05/2017	Las pruebas se irán realizando a la par que la implementación por eso de esta duración y mismo periodo que la implementación.

Tabla 95 - Planificación inicial TFM

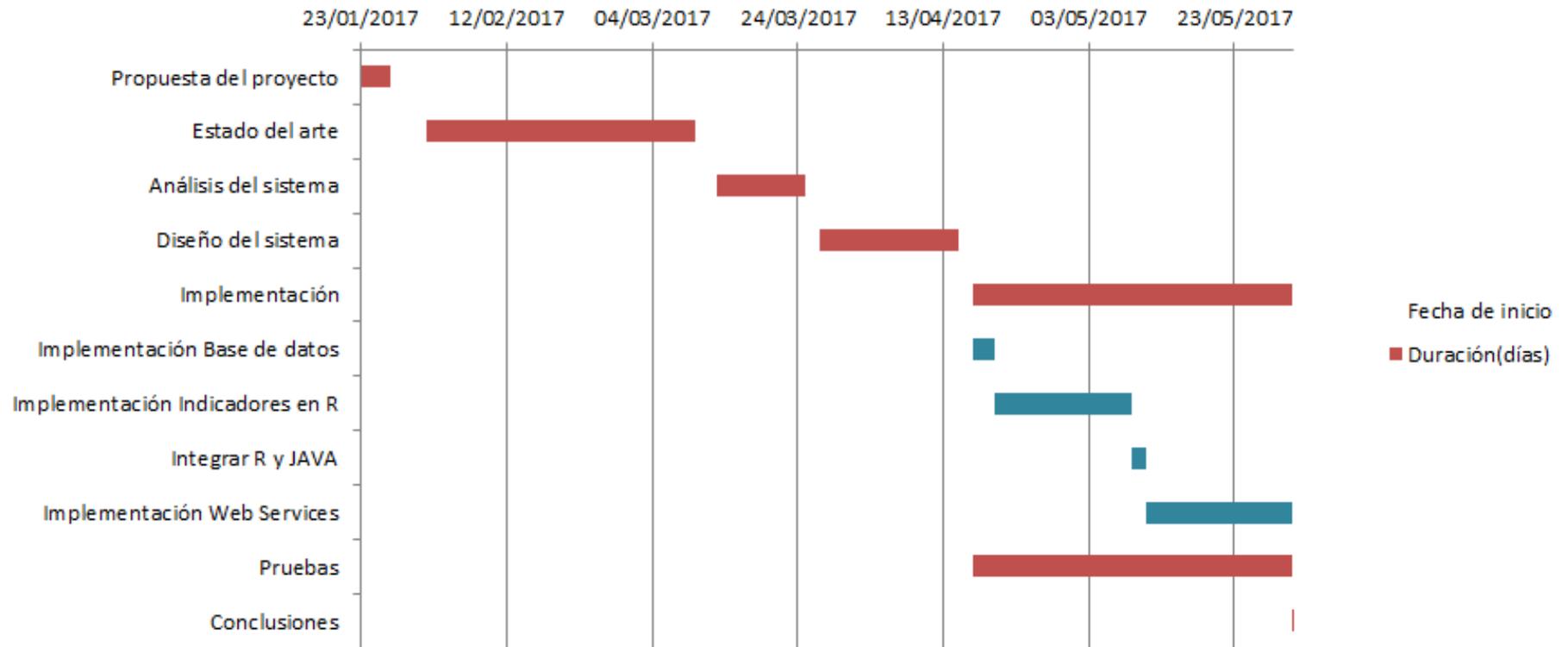


Ilustración 25 - Planificación Inicial del TFM

Planificación Real Del Trabajo Fin de Máster

Actividad	Fecha de inicio	Duración	Fecha de finalización
Propuesta del proyecto	23/01/2017	4	26/01/2017
Estado del arte	01/02/2017	28	28/02/2017
Análisis del sistema	01/03/2017	9	09/03/2017
Diseño del sistema	13/03/2017	18	31/03/2017
Implementación	03/04/2017	32	05/05/2017
Base de datos	03/04/2017	2	04/04/2017
Indicadores en código R	05/04/2017	10	14/04/2017
Integrar R y JAVA	17/04/2017	2	18/04/2017
Web Services	19/04/2017	17	05/05/2017
Pruebas	03/04/2017	32	05/05/2017

Tabla 96 - Planificación real TFM

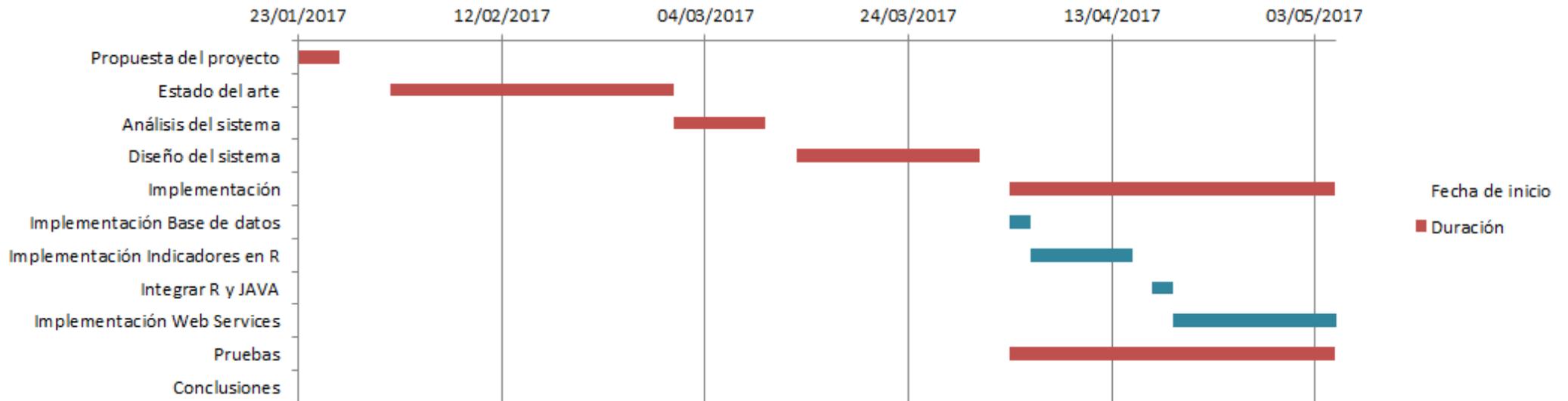


Ilustración 26 - Planificación Real del TFM

Planificación Del Proyecto

Actividad	Fecha de inicio	Duración	Fecha de finalización	Justificación
Propuesta del proyecto	23/01/2017	4	26/01/2017	Debate y discusión del proyecto.
Documento desarrollo del proyecto software	01/02/2017	76	17/04/2017	Elaboración del documento que contempla todo el proceso de desarrollo del proyecto software. Son las bases de todo por lo que requiere de tiempo para su elaboración.
Aprendizaje y documentación	18/04/2017	3	20/04/2017	Aprendizaje y documentación básica de las tecnologías usadas para el desarrollo del proyecto.
Implementación base de datos	20/04/2017	3	24/04/2017	Creación de tablas y carga de datos.
Implementación lógica de negocio	24/04/2017	103	04/08/2017	Desarrollo del código R con cálculos de indicadores, recomendaciones y gráficas. Desarrollo de los web services, servlet y handlers.
Implementación Interfaz	24/04/2017	28	19/05/2017	Implementación de la interfaz de usuario de la aplicación.
Pruebas Unitarias	04/08/2017	7	11/08/2017	Implementación y realización de las pruebas unitarias.
Pruebas de Integración	11/08/2017	7	18/08/2017	Implementación y realización de las pruebas de integración.

Tabla 97 - Planificación del proyecto

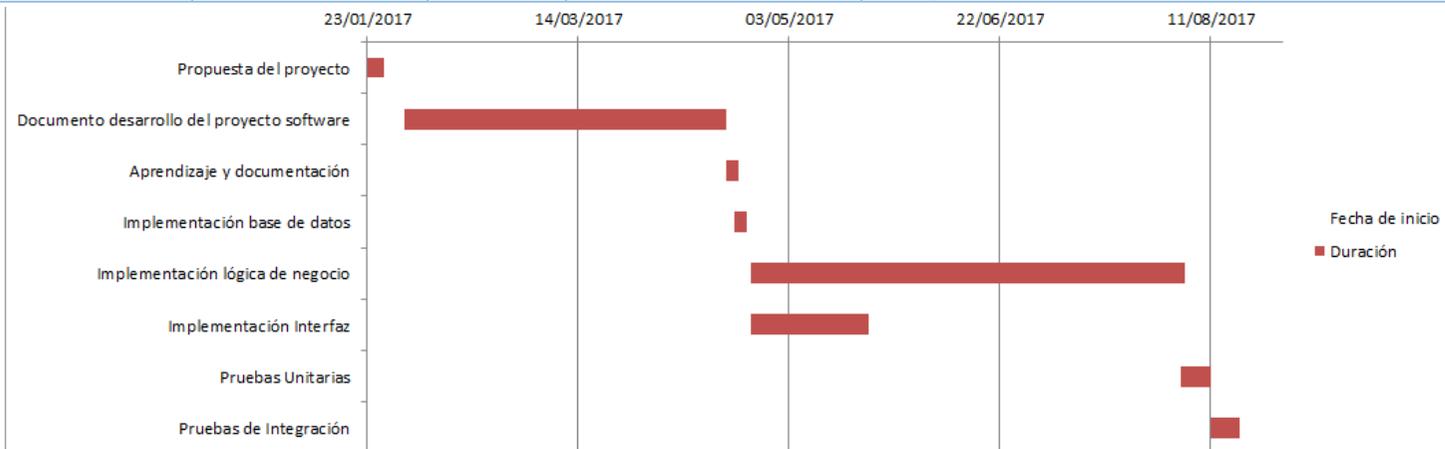


Ilustración 27 - Planificación del Proyecto

8. Presupuesto

El presupuesto sin IVA para la realización del **proyecto real** será de **cuarenta y ocho mil ciento ochenta y seis euros con diecinueve céntimos** (48.186,19 €), a lo que hay que añadir el IVA, que actualmente es de un 21%. El desglose de los costes se detalla a continuación:

Personal

Puesto	Horas	Coste(€/hora)	Coste total(€)
Jefe de proyecto	80	50	4.000
Analista programador	700	27	18.900
Analista programador	700	27	18.900
Programador web	150	20	3.000
Total (€)			44.800

Tabla 98 - Presupuesto: Personal.

Equipos

Descripción	Coste de cada equipo	Coste total de los equipos	Meses de uso	Coste aplicable al proyecto
4 Ordenadores portátiles	499 €	1996 €	5 meses	415,83 €
Total (€)				415,83

Tabla 99 - Presupuesto: Equipos.

Software

Descripción	Coste	Meses de uso	Coste aplicable al proyecto
4 x Microsoft Office 365	196 €	5 meses	40,83 €
Descarga de cotizaciones	3.000 €	5 meses	625 €
Total (€)			665,83

Tabla 100 - Presupuesto: Software.

Material fungible

Descripción	Coste de cada elemento	Coste total de los elementos
1 Paquete Papel Din A-4	2,95 €	2,95 €
4 bolígrafos	0,5 €	2 €
1 Paquete 1000 Post - it	5 €	5 €
Total (€)		9,95

Tabla 101 - Presupuesto: Material Fungible.

Otros gastos

Costes indirectos	5% de los costes directos
Total (€)	2.294,58 €

Tabla 102 - Presupuesto: Otros Gastos.



Resumen

Concepto	Coste (€)
Personal	44.800
Equipos	415,83
Software	665,83
Material fungible	9,95
Otros gastos	2.294,58
Total sin IVA (€)	48.186,19 €
Total con IVA (€)	58.305,29 €

Tabla 103 - Presupuesto: Resumen.

9. Marco regulador

En este apartado se describen las distintas restricciones legales de uso de los distintos productos software utilizados para implementar el prototipo. En cuanto a los aspectos legales de las SGIIC no se documenta nada, ya que como se comentó con anterioridad, ya existe software de front-office financiero que se encarga de eso y del que hace uso la aplicación. Por ejemplo, el cálculo de límites oficiales o la verificación de si una orden cumple con lo establecido en el folleto del fondo. Además, el sistema no funciona de forma autónoma por lo que tampoco se ve afectado por temas legales que afectan a sistemas de trading automático.

9.1 Java

Java se distribuye bajo una licencia de software libre GNU GPL, una de las más utilizadas en el mundo del software libre y que garantiza la libertad de compartir usar, editar, compartir y estudiar el software [22].

9.2 R

R se distribuye bajo una licencia de software libre GNU GPL. Tiene código fuente bajo la versión GPL-2 y GPL-3 [22]. Además, tiene código fuente bajo la licencia GNU LGPL-2.1, la cual también es una licencia de software libre compatible con las enumeradas anteriormente [23].

9.3 Rserve

La librería Rserve se distribuye también bajo la licencia GNU GPL que se puede consultar en [22].

9.4 Yahoo finance

Cada data vendor tiene sus propias condiciones de uso y licencias. Para el TFM se ha hecho uso de los datos que proporciona Yahoo Finance y también se ha hecho uso de una API de terceros que hace uso directo de la API de Yahoo Finance.

El API utilizado tiene Copyright y su autor permite su libre uso, copia, distribución y modificación bajo las siguientes condiciones:

“THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY



CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

Mientras que el uso del API o los datos proporcionados por Yahoo Finance se distribuyen bajo una serie de términos de uso descritos en [24], los cuales son bastante restrictivos. Los datos proporcionados no podrían usarse para una versión comercial de la aplicación, pero si para hacer las pruebas y el prototipo del proyecto ya que es para uso personal y no comercial.

9.5 PostgreSQL

PostgreSQL se distribuye bajo una licencia opensource propia denominada PostgreSQL license que permite compartir usar, editar, compartir y estudiar el software [25].

9.6 Apache Tomcat

Apache Tomcat se distribuye bajo una licencia denominada Apache License que define los términos de uso, reproducción y distribución [26].

10. Evaluación

Una vez presentado todo el proyecto de desarrollo software de la aplicación, en este apartado se documentará todo lo conseguido en cuanto a implementación. Abarcar una aplicación así en tan poco tiempo y solo por una sola persona es algo impensable, por ello, lo que se ha realizado es un pequeño prototipo que cumple con algunos de los requisitos de usuario formulados anteriormente. A continuación, se hará una verificación de requisitos cumplimentados por este prototipo y se mostrarán algunos ejemplos de los resultados obtenidos.

10.1 Verificación de requisitos de usuario

Identificador	Cumplido	Resultado del prototipo
RUC-01	Si	La aplicación calcula los siguientes indicadores: Bandas de Bollinger, EMV, MACD, RSI y Oscilador estocástico.
RUC-02	Si	La aplicación genera los gráficos de los siguientes indicadores: Bandas de Bollinger, EMV, MACD, RSI y Oscilador estocástico. Estos gráficos se exportan y almacenan en formato PNG.
RUC-03	Si	La base de datos está implementada por lo que, a pesar de carecer de interfaz visual, se pueden insertar los datos correspondientes a la creación y configuración de estrategias.
RUC-04	Si	La aplicación proporciona recomendaciones de compraventa.
RUC-05	No	La aplicación carece de interfaz gráfica por lo que esta funcionalidad no está integrada. Como se comentó anteriormente sí que se generan los gráficos y se pueden visualizar con un visor de imágenes, pero no con la propia aplicación.
RUC-06	Si	La funcionalidad de backtesting se ha implementado.
RUC-07	No	No se ha implementado este requisito.
RUC-08	No	No se ha implementado este requisito. Sí que se ha diseñado un prototipo de la interfaz gráfica.
RUC-09	Si	Esta funcionalidad se ha implementado, pero no de forma gráfica.
RUC-10	No	El prototipo se ha realizado con un horizonte temporal único.
RUC-11	No	No se ha implementado este requisito.
RUC-12	No	No se ha implementado este requisito.
RUC-13	No	No se ha implementado este requisito.
RUC-14	No	No se ha implementado este requisito.
RUC-15	Si	La aplicación obtiene cotizaciones de la base de datos para

		los cálculos. Se he hecho uso de una API que conecta con Yahoo Finance [21].
RUI-01	NA	No aplica.
RUR-01	No	El prototipo no tiene desarrollada la parte web.
RUR-02	Si	El desarrollo se ha realizado teniendo en cuenta el API de JAVA 5.
RUR-03	Si	El backtesting permite la realización de test personalizados mediante el paso de una serie de parámetros.
RUR-04	Si	Las funcionalidades de la aplicación se han incorporado en distintos web services según lo documentado anteriormente.
RUR-05	NA	No aplica.
RUR-06	Si	La base de datos implementada se ha realizado haciendo uso de PostgreSQL.
RUR-07	No	No se ha cumplimentado al ser un prototipo.
RUR-08	Si	El cálculo de los indicadores implementados se ha realizado en el lenguaje de programación R.
RUR-09	Si	El programa funciona para activos de renta variable cotizados en mercados regulados.

Tabla 104 - Verificación de requisitos

10.2 Ejemplos

Recomendación de compraventa

En este ejemplo se muestra el resultado de realizar una recomendación de compraventa. Para ello se han utilizado un conjunto de cotizaciones pertenecientes al Santander entre los años 2013 y 2014. La estrategia utilizada para la recomendación es la estrategia "1", cuya composición se puede apreciar en la siguiente imagen:

Codigo	Estrategia	Indicador	Ponderacion	Codigo	Nombre
9	1	15	0,34	15	BBollinger
8	1	14	0,33	14	MACD
7	1	12	0,33	12	RSI
16	2	13	0,2	13	oscEstoc
18	2	15	0,2	15	BBollinger
17	2	14	0,2	14	MACD
19	2	16	0,2	16	EMV
15	2	12	0,2	12	RSI

Ilustración 28 - Tabla de base de datos

El resultado de ejecutar el proceso de recomendación es el siguiente:

```
NUM REG: 3
Resultado RSI 3.0
NOMBRE: RSI PESO: 0.33 APORTACION: 3.0 APORTACION PONDERADA: 0.99
Resultado MACD 2.0
NOMBRE: MACD PESO: 0.33 APORTACION: 2.0 APORTACION PONDERADA: 0.66
Resultado BBollinger 1.0
NOMBRE: BBollinger PESO: 0.34 APORTACION: 1.0 APORTACION PONDERADA: 0.34
NEUTRO
Resultado estrategia-----> 1.99
```

Ilustración 29 - Resultado ejecución recomendación de compraventa.

La codificación de los resultados es: 3 = “COMPRA”, 2 = “NEUTRO” y 1 = “VENDER”. Sumando lo que aporta cada estrategia de forma ponderada con su peso, obtenemos un valor de 1.99 que atendiendo a la codificación descrita en el punto 5.1.3.2 del documento, tenemos una recomendación de “NEUTRO”, es decir, de no vender ni comprar. Como justificación de este resultado tenemos las tres gráficas que han generado los 3 indicadores:

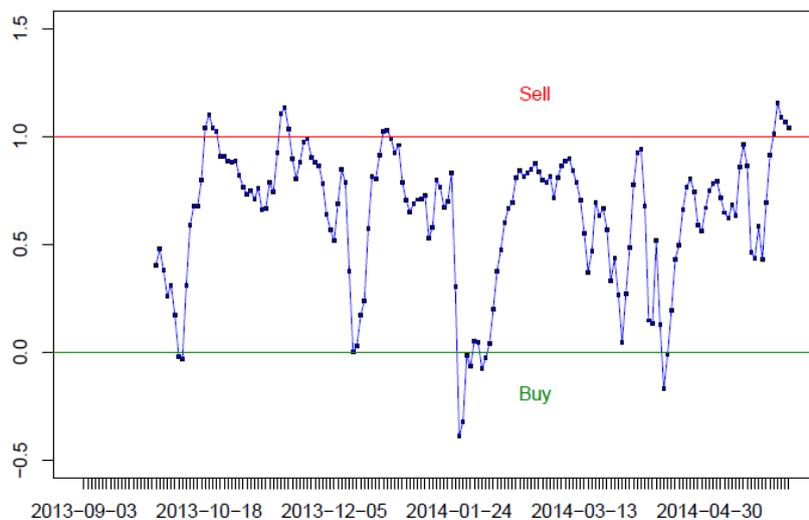


Ilustración 30 - Bandas de Bollinger

Tal y como se observa en la traza de ejecución, en la que las Bandas de Bollinger dan señal de venta, la gráfica perteneciente al indicador también nos muestra la señal de venta.

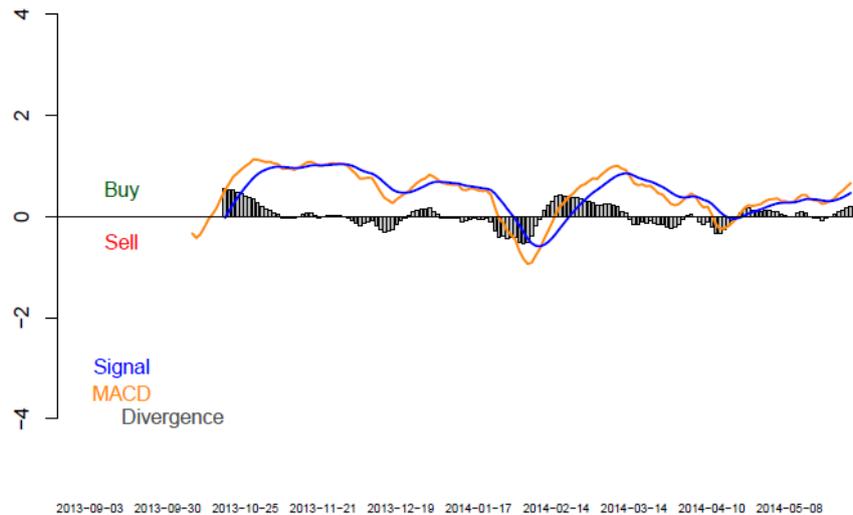


Ilustración 31 – MACD

En cuanto al indicador MACD, se observa que en la última fecha no se ha producido el cruce de la línea MACD con la línea señal ni hacia arriba ni hacia abajo por lo que no hay señal de compra ni de venta y devuelve “NEUTRO”, igual que lo que se puede observar en la traza anterior. Si bien es verdad que la línea MACD se mantiene en la parte de compra, esta señal es más débil que la señal proporcionada por el cruce de las líneas por eso se ha decidido codificar esta situación como “Neutro”. En una aplicación con 5 niveles de recomendación en vez de 3, el cruce de líneas se podría codificar como “COMPRA FUERTE” o “VENTA FUERTE” y que MACD esté en la zona de compra o de venta se podría codificar simplemente con un “COMPRA” o “VENTA”.

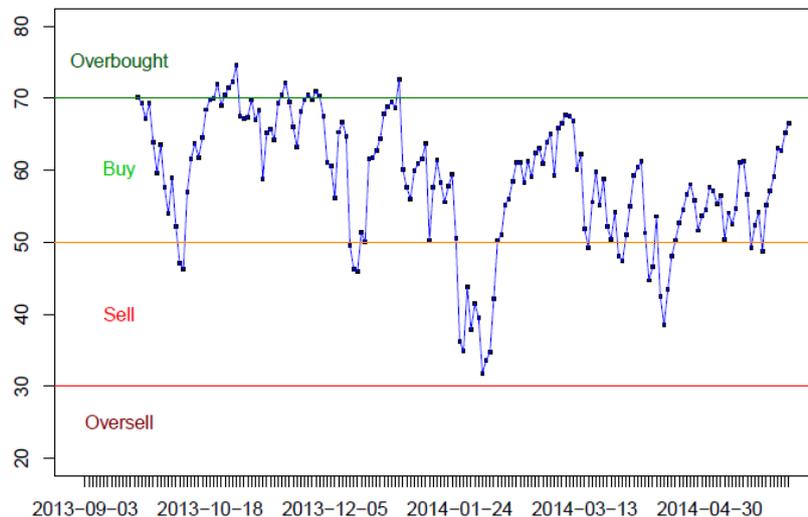


Ilustración 32 – RSI

Por último, podemos ver que el indicador RSI se mantiene entre los valores 50 y 70 que indican “COMPRAR”, igual que el resultado obtenido en la traza de ejecución.

Backtesting

En este ejemplo de funcionamiento, se muestra una ejecución de la funcionalidad de backtesting. Para ello, se ha elegido un subconjunto de datos del conjunto descrito anteriormente, que representaría una situación pasada en la que se quiere probar una estrategia. La estrategia elegida para realizar la prueba es la número “2” que se compone de todos los indicadores implementados en el prototipo, que son cinco, los cuales están equiponderados.

El subconjunto de datos utilizado para realizar el backtesting contiene las cotizaciones del Santander desde el 03-09-2013 hasta el 2014-02-14. La fecha en la que se va a realizar la recomendación es para el 2014-01-17. Se hará la recomendación de compraventa, por tanto, con los datos que van desde el 03-09-2013 al 2014-01-17. Después se hará la diferencia de las cotizaciones que hubo en la fecha de la recomendación, 2014-01-17, y en la fecha futura para la que se quiere ver si he ganado o perdido 2014-02-14. La diferencia temporal es de un mes que, para un horizonte temporal diario (como es el caso), entra dentro de la franja de inversión a corto plazo. El resultado obtenido es el siguiente:

```
NUM REG: 5
Resultado RSI 3.0
  NOMBRE: RSI PESO: 0.2 APORTACION: 3.0 APORTACION PONDERADA: 0.6000000000000001
Resultado OscEstoc 2.0
  NOMBRE: OscEstoc PESO: 0.2 APORTACION: 2.0 APORTACION PONDERADA: 0.4
Resultado MACD 2.0
  NOMBRE: MACD PESO: 0.2 APORTACION: 2.0 APORTACION PONDERADA: 0.4
Resultado BBollinger 2.0
  NOMBRE: BBollinger PESO: 0.2 APORTACION: 2.0 APORTACION PONDERADA: 0.4
Resultado EMV 3.0
  NOMBRE: EMV PESO: 0.2 APORTACION: 3.0 APORTACION PONDERADA: 0.6000000000000001
COMPRAR
Resultado estrategia-----> 2.4
PRECIO INI: 183.6 PRECIO FIN: 184.01 DIFERENCIA: 0.40999999999999966
```

Ilustración 33 - Ejecución backtesting

Como se puede observar, la recomendación de la estrategia para el conjunto de datos que va desde el 03-09-2013 al 2014-01-17 es de “COMPRAR”. Posteriormente, restamos la cotización de la fecha 2014-02-14 menos la cotización de la fecha en la que se hizo la recomendación obteniendo un incremento en el valor del activo de 0.41 céntimos por acción. Esto significa que para este caso pasado la estrategia funcionó correctamente ya que se nos recomendó comprar y el valor de la acción ha subido.

11. Conclusiones

En este apartado se presentan las conclusiones tanto personales como referentes al proyecto. Asimismo, se plantean distintas opciones para trabajos futuros. Por último, se comentará la relación de este proyecto con las distintas materias cursadas en el máster.

11.1 Conclusiones

Conclusiones del proyecto

El desarrollo del proyecto y del prototipo implementado ha servido como un primer paso a la hora de llevar a cabo el desarrollo real de la aplicación. Se han estudiado y explorado las distintas problemáticas y posibilidades que hay para su desarrollo y se han asentado las bases del mismo haciendo un estudio teórico e implementando funcionalidades básicas de la aplicación. También se ha analizado y diseñado todo lo necesario para realizar el proyecto, incluyendo también un plan de pruebas para la aplicación. Era imposible abordar mucho más del proyecto en lo realizado en este trabajo por su magnitud y esta primera toma de contacto será muy útil para una implementación futura.

Conclusiones personales

En lo personal, ha sido un trabajo que me ha gustado ya que trata un área que desconocía y por estar centrado en el ámbito financiero. He aprendido bastante sobre el análisis bursátil, algo de lo que en el máster se ha tratado muy por encima y de lo que conocía muy poco. Informatizar una de las herramientas básicas que los gestores e inversores en general utilizan a diario para tomar sus decisiones de inversión era algo que me llamaba la atención.

El trabajo ha sido muy útil también ya que me ha permitido aplicar conocimientos adquiridos en el máster y aplicar conocimientos adquiridos en el grado y que hacía tiempo no trataba. También me ha permitido coger experiencia a la hora de elaborar proyectos de desarrollo software.

Por otro lado, creo que el proyecto se nos ha ido de las manos en lo que a dimensiones se refiere. Esto, bajo mi punto de vista, es por tratar de abordar lo que sería una aplicación comercial con muchas funcionalidades de forma casi completa. Esto era algo que sabíamos desde el principio pero que quería probar y enfrentarme a ello porque la idea me llamaba la atención. A veces era complejo pensar en todo y cada pocos días de hacer algo surgía un cabo suelto que había que resolver. Para un trabajo académico individual es mejor hacer algo más específico.

Si tuviera la opción de volver al pasado y volver a decidir, haría el mismo trabajo sin dudar, aunque seguramente lo afrontaría de otra manera.

11.2 Líneas futuras

Todo proyecto software es mejorable y este, no es una excepción. Es posible plantear una serie de mejoras o alternativas para trabajos futuros sobre esta aplicación.

- Introducir gráficos dinámicos: En la actualidad la aplicación genera gráficos estáticos que se exportan en una imagen y se plasman en la interfaz de la aplicación. Estos gráficos son estáticos y no es posible interactuar con ellos. La generación de gráficos interactivos sería una mejora para la aplicación.
- Mejorar el rendimiento de la aplicación: Realizar un estudio de las posibilidades existentes de paralelización o concurrencia que tiene la aplicación, de forma que se consigan reducir los tiempos de ejecución de la misma. Estudiar la posibilidad de crear un grid de datos en memoria que contenga toda la tabla de cotizaciones. Esta tabla se utiliza en diferentes funcionalidades del sistema, alguna de ellas muy pesada, y permitiría reducir tiempos de ejecución ya que el acceso a memoria es mucho más rápido que el acceso a disco.
- Ampliar los niveles de recomendación: pasar de los tres niveles propuestos en el proyecto de COMPRA, VENTA y NEUTRO a COMPRA, VENTA, NEUTRO, VENTA FUERTE y COMPRA FUERTE.
- Ampliar el conjunto de activos financieros objeto de esta aplicación.
- Ampliar las posibilidades de la aplicación permitiendo la realización de análisis fundamental.
- Añadir a la aplicación un mayor número de indicadores técnicos. Existen infinidad de indicadores y siempre es bueno añadir cuantos más mejor de forma que los gestores tengan donde elegir.
- Añadir la obtención de cotizaciones procedentes de otros proveedores de datos y permitirle elegir al usuario.
- Hacer un estudio de diferentes algoritmos para realizar el autoajuste, de forma que se obtenga aquel con un funcionamiento mejor entre distintas posibilidades.
- Añadir la posibilidad de mandar distintos tipos de órdenes a mercado.

11.3 Relación del trabajo con el máster cursado

Durante el desarrollo del proyecto fin de máster se ha tratado con multitud de elementos de carácter financiero (código, documentación, programas de carácter financiero), los cuales se pueden relacionar directamente con algunos de los contenidos que se han visto en el máster en las distintas asignaturas impartidas.

A continuación, se realiza una descripción que asocia algunos de los elementos del TFM con las áreas de conocimiento vista en el máster:

- Sistemas de Soporte a la Decisión en el Sector Financiero: la aplicación propuesta en este proyecto es puramente un sistema de soporte a la decisión.
- Mercados financieros: tratamiento de activos financieros de renta variable cotizada en mercados secundarios regulados, teoría de mercados financieros, cálculo de rentabilidades y estudio de fondos de inversión.
- Tecnologías Aplicadas a los Mercados Financieros: teoría de mercados financieros y renta variable, tipos de órdenes, proceso de invertir: desde la decisión de invertir hasta el back-office financiero. Uso servicios web y arquitectura orientada a servicios.
- Desarrollo de software financiero: desarrollo de software con elementos de carácter financiero como por ejemplo el cálculo de rentabilidades, cálculo de indicadores de análisis técnico o implementación de la lógica de inversión de análisis técnico. El diseño/implementación se ha realizado aplicando conocimientos de altas prestaciones como por ejemplo el uso de hilos para realizar tareas de forma concurrente. Uso del API de Yahoo finance para la descarga de cotizaciones.

12.Referencias

- [1]"Administrador de un Fondo de Inversión - Economipedia", *Economipedia*, 2017. [Online]. Available: <http://economipedia.com/definiciones/gestor-administrador-fondo-inversion.html>. [Accessed: 07- Feb- 2017].
- [2]I. URRÁ MAGALLÓN and G. ZOTES CIANCAS, *ESTRATEGIAS STOCK PICKING*, 1st ed. 2017.
- [3]J. Murphy, *Análisis técnico de los mercados financieros*, 1st ed. Barcelona: Gestión 2000, 2000.
- [4]J. Codina Castro, *Manual de análisis técnico*, 1st ed. Madrid: Inversor Ediciones, 2011.
- [5]U. S.L., "Momentum", *Expansion.com*, 2017. [Online]. Available: <http://www.expansion.com/diccionario-economico/momentum.html>. [Accessed: 08- Feb- 2017].
- [6]*Proyecto Indicadores técnicos*, 5th ed. Madrid: Arfima financial solutions, 2015.
- [7]A. Elder, *Vivir del trading*, 1st ed. [La Coruña]: Netbiblo, 2004.
- [8]"ATR", *Pullback.es*, 2017. [Online]. Available: <http://www.pullback.es/wp-content/uploads/2015/01/ejemplo-de-ATR-y-precios.jpg>. [Accessed: 13- Feb- 2017].
- [9]U. S.L., "Términos económicos de Análisis bursátil", *Expansion.com*, 2017. [Online]. Available: <http://www.expansion.com/diccionario-economico/bolsa/analisis-bursatil.html>. [Accessed: 15- Feb- 2017].
- [10]"Bloomberg for Enterprise | Bloomberg Finance L.P.", *Bloomberg Enterprise*, 2017. [Online]. Available: <https://www.bloomberg.com/enterprise/>. [Accessed: 01- Mar- 2017].
- [11]"English Home", *Thomson Reuters*, 2017. [Online]. Available: <https://www.thomsonreuters.com/en.html>. [Accessed: 01- Mar- 2017].
- [12] "FINVIZ.com - Stock Screener", *Finviz.com*, 2017. [Online]. Available: <https://finviz.com/>. [Accessed: 01- Mar- 2017].
- [13] B. (SAN), "Banco Santander Análisis técnico de acciones (SAN)", *investing.com Español*, 2017. [Online]. Available: <https://es.investing.com/equities/banco-santander-technical>. [Accessed: 01- Mar- 2017].
- [14] I. Staff, "Backtesting", *Investopedia*, 2017. [Online]. Available: <http://www.investopedia.com/terms/b/backtesting.asp>. [Accessed: 08- Mar- 2017].
- [15] *La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real*, 1st ed. Microsoft Corporation, 2006.
- [16] "Java Platform, Enterprise Edition (Java EE) | Oracle Technology Network | Oracle", *Oracle.com*, 2017. [Online]. Available: <http://www.oracle.com/technetwork/java/javae/overview/index.html>. [Accessed: 21- Mar- 2017].



- [17] "spring.io", *Spring.io*, 2017. [Online]. Available: <https://spring.io/>. [Accessed: 21- Mar- 2017].
- [18] M. Davis, "JBoss vs. Tomcat: Choosing A Java Application Server", *Future Hosting*, 2017. [Online]. Available: <https://www.futurehosting.com/blog/jboss-vs-tomcat-choosing-a-java-application-server/>. [Accessed: 21- Mar- 2017].
- [19] "CRAN - Package rJava", *Cran.r-project.org*, 2017. [Online]. Available: <https://cran.r-project.org/web/packages/rJava/index.html>. [Accessed: 21- Mar- 2017].
- [20] "CRAN - Package Rserve", *Cran.r-project.org*, 2017. [Online]. Available: <https://cran.r-project.org/web/packages/Rserve/index.html>. [Accessed: 21- Mar- 2017].
- [21] S. Strickx, "Java Finance Quotes API for Yahoo Finance", *Financequotes-api.com*, 2017. [Online]. Available: <http://financequotes-api.com/>. [Accessed: 03- May- 2017].
- [22]"The GNU General Public License v3.0- GNU Project - Free Software Foundation", *Gnu.org*, 2017. [Online]. Available: <http://www.gnu.org/licenses/gpl.html>. [Accessed: 03- May- 2017].
- [23]"GNU Lesser General Public License v2.1 - GNU Project - Free Software Foundation", *Gnu.org*, 2017. [Online]. Available: <https://www.gnu.org/licenses/lgpl-2.1.en.html>. [Accessed: 03- May- 2017].
- [24]"Yahoo APIs Terms of Use (Yahoo Developer Network Terms of Use)", *Policies.yahoo.com*, 2017. [Online]. Available: <https://policies.yahoo.com/us/en/yahoo/terms/product-atos/apiforydn/index.htm>. [Accessed: 03- May- 2017].
- [25]"PostgreSQL: License", *Postgresql.org*, 2017. [Online]. Available: <https://www.postgresql.org/about/licence/>. [Accessed: 03- May- 2017].
- [26]"Apache License, Version 2.0", *Apache.org*, 2017. [Online]. Available: <https://www.apache.org/licenses/LICENSE-2.0>. [Accessed: 03- May- 2017].
- [27]"Yahoo Finanzas - Financiación empresarial, bolsa de valores, cotizaciones, noticias", *Es.finance.yahoo.com*, 2017. [Online]. Available: <https://es.finance.yahoo.com/>. [Accessed: 09- May- 2017].