

Sistema de Informação para Negociação Automática em Investimentos Financeiros

Luís Miguel Falcão Oliveira

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática
(2^o ciclo de estudos)

Orientador: Prof. Doutor Pedro Domingues de Almeida

Covilhã, Julho de 2021

Agradecimentos

A conclusão deste trabalho, bem como da grande parte da minha vida académica não seria possível sem a ajuda da minha família, namorada e amigos, que sempre me apoiaram e me incentivaram a nunca desistir, mesmo quando eu era a primeira pessoa a fazer-lo. Sem o apoio deles seria tudo mil vezes mais difícil. Quero deixar também os meus agradecimentos aos professores desta instituição, pela sorte que tive em conhecê-los e de aprender com. Em especial, ao Professor Doutor Pedro Almeida pela oportunidade que me deu para desenvolver esta dissertação. Para finalizar, deixo um especial agradecimento à Universidade da Beira Interior que me deu a oportunidade de me formar no curso que sempre sonhei.

Resumo

Este projeto visa a construção de um sistema de informação para negociação automática em mercados financeiros. O sistema implementa um algoritmo de decisão simples e usa este algoritmo para gerar indicações de compra e de venda de um activo financeiro. O sistema interage em tempo real com a *Application Programming Interface* (API) de uma corretora para lançar ordens de compra e venda, ou para as modificar ou retirar, e recolhe o *feedback* da corretora sobre as execuções de ordens e sobre a evolução e a situação da conta de negociação. O sistema ainda recolhe da Internet informação contextual adicional, proporciona ao utilizador uma interface informativa e de comando eficaz, e mantém um registo de todas as operações e dos restantes dados relevantes.

Palavras-chave

Mercados financeiros, Ativos financeiros, Sistema Automatizado, *Backtesting*, Renda variável, Ações, Dinheiro, Obrigações, Fundos de investimento e *Commodities*.

Índice

1	Introdução	1
1.1	Enquadramento e Motivação	1
1.2	Objetivos	1
1.3	Organização do Documento	2
2	Estado da Arte	3
2.1	Ativos Financeiros	3
2.2	Mercado Financeiro	4
2.3	Software de Negociação Automatizado	5
2.4	Proposta	8
2.5	Conclusões	8
3	Problema Proposto a Resolver	9
3.1	<i>Interactive Brokers</i> (IBKR)	9
3.2	Abordagem Proposta	10
3.3	Conclusões	10
4	Engenharia de Software	11
4.1	Análise dos Requisitos e Casos de Uso	11
4.1.1	Requisitos Funcionais	11
4.1.2	Requisitos Não-Funcionais	12
4.1.3	Casos de Uso	12
4.2	Base de Dados	13
4.3	Arquitetura	14
4.4	Conclusões	14
5	Metodologias Usadas e Resultados	15
5.1	<i>Trader Workstation</i> (TWS)	15
5.2	Tecnologias e Ferramentas Usadas	16
5.2.1	Linguagens de programação	16
5.2.2	Base de Dados	17
5.2.3	<i>Frameworks</i>	18
5.3	Metodologias	19
5.3.1	Obtenção dos dados do Portefólio	19
5.3.2	Obtenção das Posições	20
5.3.3	Obtenção das Ordens em Aberto	20
5.3.4	Obtenção das Ordens Enviadas	21
5.3.5	Obtenção dos dados recolhidos pelo <i>Trading Bot</i>	21
5.3.6	Obtenção de Dados Históricos	22
5.3.7	Envio de Ordens	22

5.3.8	Cancelamento de Ordens	23
5.3.9	Funcionamento do <i>Trading Bot</i>	23
5.4	Funcionalidades da Aplicação	24
5.4.1	Página Inicial	24
5.4.2	Portefólio	25
5.4.3	Preços	27
5.4.4	<i>Trading Bot</i>	30
5.5	Resultados	31
5.6	Conclusões	31
6	Reflexão Crítica e Problemas Encontrados	33
6.1	Objetivos Propostos vs. Alcançados	33
6.2	Problemas Encontrados e Reflexão Crítica	33
6.3	Conclusão	34
7	Conclusão	35
7.1	Conclusões Principais	35
7.2	Trabalho Futuro	35
	Bibliografia	37

Lista de Figuras

2.1	Crescimento por região do uso de softwares de negociação automatizados baseados em algoritmos complexos [5]	5
2.2	Lista do 10 maiores <i>Robo-advisors</i> do mercado com base nos ativos sob gestão em Março de 2020.	6
2.3	<i>Dashboard</i> da plataforma SmarttBot.	7
4.1	Casos de uso da plataforma.	13
4.2	Estrutura da Base de Dados.	13
4.3	Diagrama de representação da arquitectura do sistema.	14
5.1	<i>Dashboard</i> da plataforma <i>Trader Workstation</i> (TWS) [1]	15
5.2	Diagrama de fluxo para obter dados do portefólio.	19
5.3	Diagrama de fluxo para obter as posições da conta.	20
5.4	Diagrama de fluxo para obter as ordens em aberto.	20
5.5	Diagrama de fluxo para obter as ordens submetidas.	21
5.6	Diagrama de fluxo para obter os dados recolhidos pelo algoritmo de negociação automatizada.	21
5.7	Diagrama de fluxo para obter os dados históricos de um determinado ativo.	22
5.8	Diagrama de fluxo para submeter uma ordem.	22
5.9	Diagrama de fluxo para cancelar uma ordem.	23
5.10	Diagrama de fluxo para inicializar o algoritmo de negociação automatizada.	23
5.11	Página inicial da aplicação.	24
5.12	Página do portefólio.	25
5.13	Tabela de posições.	25
5.14	Opções de venda.	26
5.15	Tabela de ordens em aberto.	26
5.16	Tabela de ordens enviadas.	27
5.17	Página de preços.	27
5.18	Gráfico de consulta de preços históricos.	28
5.19	Opções de compra.	29
5.20	Página do <i>trading bot</i>	30
5.21	Tabela de dados recolhidos pelo algoritmo de negociação automatizado.	30

Lista de Tabelas

2.1	Prós e contras no que diz respeito à utilização de softwares de negociação automatizados.	6
3.1	Lista de bolsas de valores para investir disponibilizadas pela corretora IBKR.	9
5.1	Listagem de ordens submetidas pelo algoritmo de negociação automatizado.	31
6.1	Lista de objetivos alcançados.	33

Lista de Acrónimos

IDE *Integrated Development Environment*

IBKR *Interactive Brokers*

ETF *Exchange Traded Fund*

NYSE *New York Stock Exchange*

TSX *Toronto Stock Exchange*

Cboe *Chicago Board Options Exchange*

BMV *Bolsa Mexicana de Valores*

VSCode *Visual Studio Code*

JS *JavaScript*

TS *TypeScript*

HTML *Hyper Text Markup Language*

CSS *Cascading Style Sheets*

XMAD *Bolsa de Madrid*

SGBD *Sistema de Gestão de Base de Dados*

SIX *Swiss Exchange*

LSE *London Stock Exchange*

XML *Extensible Markup Language*

XHTML *eXtensible Hypertext Markup Language*

HTTP *Hypertext Transfer Protocol*

API *Application Programming Interface*

FRA *Frankfurt Stock Exchange*

BSE *Budapest Stock Exchange*

MOEX *Moscow Stock Exchange*

ASX *Australia Stock Exchange*

TASE *Tel Aviv Stock Exchange*

NSE *National Stock Exchange of India*

SGX *Singapore Stock Exchange*

HKSE *Hong Kong Stock Exchange*

KSC *Korea Stock Exchange*

TSE *Tokyo Stock Exchange*

TWS *Trader Workstation*

SMA *Simple Medium Average*

NoSQL *No Structured Query Language*

JSON *JavaScript Object Notation*

Capítulo 1

Introdução

Este documento foi elaborado no âmbito da Unidade Curricular de Dissertação ou Estágio em Engenharia Informática, inserida no Mestrado em Engenharia Informática da Universidade da Beira Interior e apresenta todo o trabalho desenvolvido durante o desenrolar do mesmo.

Inicialmente será contextualizado o projeto, a área em que se enquadra, seguido da motivação, objetivos e, por fim, a organização do documento.

1.1 Enquadramento e Motivação

O investimento em ativos financeiros é um conceito que está presente na história da humanidade há várias centenas de anos que envolve empresas, dinheiro e pessoas. Desde a viragem para o século XXI, as máquinas têm ganho uma maior importância no mundo dos investimentos, pode-se dizer até que estas mudaram a maneira de se investir em ativos financeiros por completo e representam um factor chave e indispensável para o sucesso dos investidores. Ao contrário do século passado, onde as ordens de compra e venda em ativos financeiros eram realizadas em grande percentagem por humanos, as ordens de investimento nestes ativos são realizadas na maioria através de máquinas, sendo agora a concorrência entre máquinas. Tal mudança não é surpreendente dada a capacidade da computação atual das máquinas para analisar as flutuações do mercado e para executar ordens de compra e venda de ativos financeiros na bolsa de valores.

Um dos grandes desafios da atualidade nesta área é a complexidade presente no desenho de sistemas para a finalidade de automatizar as ordens de compra e venda. A tomada de decisão de uma ordem de compra ou venda de um ativo deriva de vários fatores que devem ser tomados em conta, principalmente o estudo das flutuações do mercado. De modo a que um sistema automatizado seja bem sucedido, é fundamental desenhar bons métodos e estratégias de investimento nos mercados financeiros.

1.2 Objetivos

De modo a destacar os objetivos principais para este projeto, estes foram repartidos nos seguintes pontos:

- Realizar um sistema completamente automatizado;
- Realizar um algoritmo de decisão simples para gerar indicações de compra e venda de ativos financeiros;

- Interagir em tempo real com a API da IBKR para lançar ordens de compra e venda, ou para modificar ou retirar;
- Registrar e apresentar todas as ordens realizadas de compra e venda dos ativos financeiros;
- Recolher informação contextual adicional de uma fonte externa;
- Proporcionar ao utilizador uma interface informativa e eficaz.

1.3 Organização do Documento

De modo a refletir o trabalho que foi realizado, foi entendido estruturar o relatório da seguinte forma:

- o primeiro capítulo – **Introdução** – apresenta o projeto realizado, indicando as razões que levaram à escolha do mesmo, o objetivo pretendido, assim como a forma de organização do documento;
- o segundo capítulo – **Estado de Arte** – apresenta sistemas já existentes relacionados com a proposta neste documento, as técnicas usadas pelos mesmos para a resolução dos problemas derivados da implementação de um sistema do género e apresenta a proposta do autor para o desenvolvimento de um novo sistema com base nas técnicas existentes;
- o terceiro capítulo – **Problema Proposto a Resolver** – apresenta um estudo sobre a utilização da IBKR como a corretora escolhida para a realização deste projeto e a descrição da abordagem ao problema;
- o quarto capítulo - **Engenharia de Software** - exibe a análise de requisitos e os casos de uso do sistema, bem como outros diagramas essenciais para a compreensão do sistema desenvolvido;
- o quinto capítulo – **Metodologias Usadas e Resultados** – descreve ao pormenor as metodologias e técnicas a usar para o desenvolvimento do *frontend*, *backend* e base de dados do sistema bem como todas as ferramentas a utilizar no desenvolvimento do projeto, incluindo também a descrição das funcionalidades da aplicação e os resultados obtidos;
- o sexto capítulo - **Reflexão Crítica e Problemas Encontrados** - discute os objetivos propostos e os alcançados na elaboração deste trabalho, assim como problemas encontrados no decorrer do mesmo. Este capítulo inclui ainda uma reflexão crítica sobre o trabalho realizado;
- o sétimo capítulo – **Conclusões e Trabalho Futuro** – descreve as tarefas a realizar no próximo semestre para alcançar os objetivos propostos neste documento e para a conclusão do projeto.

Capítulo 2

Estado da Arte

Neste capítulo são inicialmente apresentados os ativos financeiros, o mercado financeiro, o software de negociação automatizado e as plataformas já existentes para investir em ativos financeiros, a proposta baseada no conteúdo abordado e, por fim, uma conclusão sobre este capítulo.

2.1 Ativos Financeiros

Os ativos constituem bens ou direitos que uma empresa ou pessoa possui e que podem gerar rendimentos. Um ativo é financeiro quando existe apenas como um direito económico e o seu valor é obtido através de um direito contratual. Um exemplo de um ativo financeiro é o dinheiro, pois este representa um valor na posse de um agente que pode trocar por outros bens que desejar comprar ou depositar num banco. Os ativos financeiros são intangíveis, o que significa que estes não existem de forma física, como por exemplo um terreno ou uma casa. Estes ativos apenas recebem um valor que vem da oferta e demanda do mercado em que participam ou pelo grau de risco que possuem. No entanto, os rendimentos gerados por ativos financeiros são gerados por meio de ativos reais ou tangíveis. As ações de uma empresa, por exemplo, são ativos financeiros onde o rendimento é gerado pelos lucros da empresa a que este ativo pertence. Uma outra forma de gerar os rendimentos dos ativos financeiros é por meio dos rendimentos do governo, quando são emitidos títulos públicos. O que acontece muitas vezes na economia é que, quem possui uma poupança ou um excedente, empresta a outro agente que investe em ativos reais. Este investimento gera rendimentos onde parte destes são transferidos para quem o emprestou [3].

No mercado atual existe vários tipos de ativos financeiros, entre os quais se destacam os seguintes:

- **Ações** – parte do capital de uma empresa que pode ser adquirido e gerar rendimentos por meio dos seus dividendos ou pela sua venda.
- **Dinheiro** – carrega em si um valor de troca. A troca do mesmo com outros países pode gerar um rendimento através do câmbio.
- **Obrigações** – são emitidos por governos e empresas como uma forma de arrecadar capital para investimentos. O seu rendimento provém dos juros que são pagos.
- **Fundos de investimento** – aplicação financeira que junta o dinheiro de diversos participantes e é administrado no mercado de capitais com o objetivo de gerar rendimentos.

- **Commodities** – são produtos que funcionam como matéria-prima, produzidos em escala e que podem ser armazenados sem perda de qualidade, como petróleo, sumo de laranja congelado, café, soja, ouro, entre outros.

Dentre os ativos financeiros, existe uma classificação quanto à remuneração que estes podem gerar, podendo ser de dois tipos, de renda fixa ou de renda variável, sendo este último o tipo de ativos mais relevante para investir.

- **Ativos de renda variável** – quando na aplicação não se pode prever com certeza se haverá um rendimento ou o quanto poderá ser, diferente dos de renda fixa. Exemplos de investimentos de renda variável são as ações, fundos de ações, fundos multimercado, fundos imobiliários, entre outros.

2.2 Mercado Financeiro

O mercado financeiro é, por definição, um ambiente de compra e venda de valores mobiliários (ações, opções, obrigações), câmbio (moedas estrangeiras) e mercadorias (ouro, produtos agrícolas). Nestas negociações, estão envolvidas diversas instituições, que facilitam o encontro entre agentes e regulam e fiscalizam as transações. No mercado financeiro, o investidor é aquele que dispõe de dinheiro e que deseja rentabilizar-lo. Se por um lado há investidores, na outra ponta do mercado financeiro temos os tomadores de recursos. Eles são as empresas, instituições ou pessoas que querem captar dinheiro para diversos fins, como pagamento de dívidas, financiamento de recursos, entre outros. Nestes mercados existem também negócios empresa a empresa, como compra de matérias primas, *hedging* de compras ou vendas futuras, entre outros. De um modo geral podemos dizer que o mercado financeiro permite o devido fluxo da economia.

Na atualidade é possível, através do uso de corretoras, negociar *online* vários tipos de ativos financeiros como ações, taxas de câmbio, preços de *commodities*, obrigações, entre outros. Historicamente, as ordens de negociação em bolsas (nos mercados financeiros) eram decididas e introduzidas por humanos. As próprias bolsas também funcionavam através da interação de operadores humanos. Com o surgimento dos computadores e da Internet, esta negociação passou a ser de forma eletrônica, bem como a introdução das ordens pelos investidores. Atualmente, a maioria das ordens e decisões de compra e venda de um ativo financeiro realizam-se através de computadores, embora seguindo regras ou algoritmos complexos definidos e programados por humanos.

Com esta explicação, é possível ter uma visão mais ampla sobre como os mercados financeiros funcionam e da sua importância não só para os investidores como para a economia. De destacar que a chave aqui é entender que o poder do mercado está com quem detém o dinheiro: o investidor pode tirar proveito das necessidades de recursos de terceiros para aumentar o seu capital. Outro ponto chave é a importância do recurso ao poder de computação das máquinas atuais. Hoje em dia nos mercados financeiros, a concorrência ao investimento deixou de ser entre humanos e passou a ser entre máquinas pois estas superam as limitações do ser humano para analisar e executar ações no mercado.

2.3 Software de Negociação Automatizado

O conceito de sistema de negociação automatizado foi introduzido pela primeira vez por Richard Donchian em 1949, quando ele usou um conjunto de regras para comprar e vender fundos. Então, na década de 1980, o conceito de negociação baseada em regras tornou-se mais popular quando investidores famosos como John Henry começaram a usar tais estratégias.

Um software de negociação automatizado [2] é uma plataforma de negociação sofisticada que usa algoritmos para monitorizar os mercados em certas condições. Um corretor do mercado de ações que usa uma plataforma automatizada pode definir algumas diretrizes iniciais para ações, tais como ações voláteis de pequena capitalização com preços que recentemente ultrapassaram a sua média móvel. Os investidores também podem definir os pontos de entrada e saída para as suas posições potenciais e, em seguida, permitir que os computadores assumam. O software automatizado pode selecionar ações que atendam aos critérios e executar negociações com base nos parâmetros pré-estabelecidos. A Figura 2.1 ilustra o crescimento por região do uso de softwares deste tipo que se baseia em algoritmos complexos nos últimos anos e a previsão deste crescimento até ao ano de 2024.



Fig. 2.1: Crescimento por região do uso de softwares de negociação automatizados baseados em algoritmos complexos [5]

Alguns dos benefícios da negociação automatizada são óbvios. Os seres humanos são limitados no número de ações que podem monitorizar num determinado momento. Por outro lado, os computadores podem examinar diferentes mercados e títulos com uma velocidade incompreensível para os investidores de carne e osso. Os algoritmos podem detectar uma reversão de tendência e executar uma nova negociação numa fração de segundo. Na verdade, aproximadamente 80% das ações negociadas nas bolsas de valores dos EUA vêm de sistemas de negociação automática [6].

As estratégias automatizadas só funcionam em condições de mercado específicas: não são universais e o investidor precisa de entender quando aplicar-las. Além disso, muitas estratégias automatizadas se tornam excessivamente otimizadas e deixam de ter em conta

as condições do mercado do mundo real. De lembrar que por mais boa e bem desenhada que seja uma estratégia para a tomada de decisão, esta não é infalível. Como podemos ver na Tabela 2.1, um software de negociação automatizado possui algumas contra-partidas para além dos seus benefícios.

Prós	Contras
+ Execução rápida de negociações	- A falta de controlo é difícil para alguns investidores aceitarem
+ Segura uma estratégia predeterminada e não se desvia	- Estratégias de sucesso em <i>backtests</i> podem falhar na vida real
+ Remove o stress e a emoção	- Fácil de otimizar demais a estratégia usada
+ Custos de transação mais baixos	- Muitos robôs ou consultores especializados têm taxas de sucesso questionáveis

Table 2.1: Prós e contras no que diz respeito à utilização de softwares de negociação automatizados.

Os ativos de um cliente são alocados com base nas preferências de risco e no retorno desejado. Geralmente, a lista de produtos de investimento incluem ações, títulos, futuros, *commodities*, imóveis e, na maioria das vezes, carteiras *Exchange Traded Fund* (ETF). Os clientes escolhem se desejam técnicas de alocação passiva de ativos que reduzam os seus riscos ou adotam o estilo de gestão de ativos mais ativo, gerando retornos mais elevados mas aumentam o risco.

Existem mais de 100 destes serviços disponíveis ao público. Estes serviços, também conhecidos como *Robo-advisors*, detinham \$224 bilhões de dólares americanos em ativos sob gestão em Outubro de 2017 [8]. Na Figura 2.3 são apresentados os maiores *Robo-advisors* com base nos ativos sob gestão.

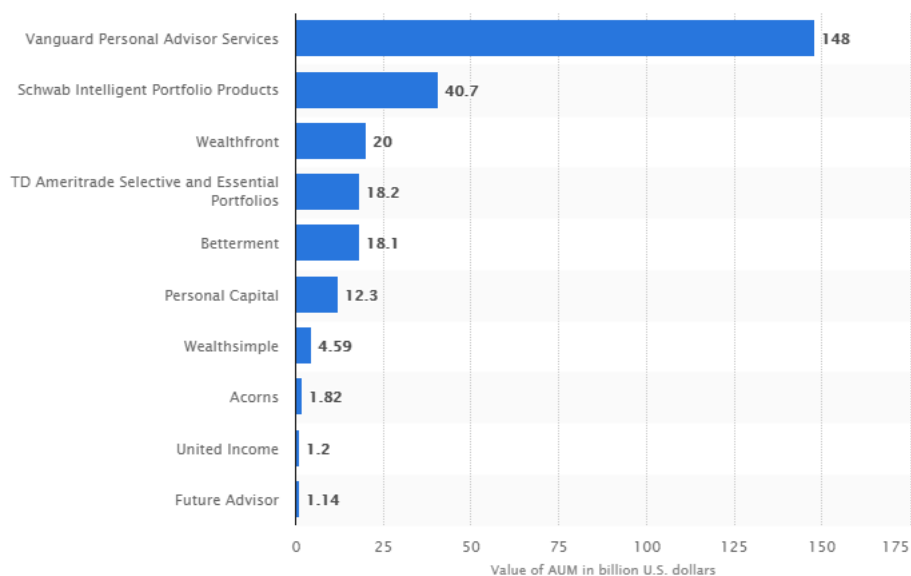


Fig. 2.2: Lista do 10 maiores *Robo-advisors* do mercado com base nos ativos sob gestão em Março de 2020.

Os serviços oferecidos pelas empresas apresentadas na Figura 2.3 representam a vanguarda no que diz respeito a sistemas de negociação automatizados. Estes serviços podem ser adquiridos por um preço elevado comparado com outros serviços menos eficazes. Vários bancos e fundos de cobertura (*hedge funds*) utilizam estes serviços para investir os

seus fundos, o que realça ainda mais o facto destas empresas serem de topo e possuírem os melhores algoritmos e técnicas de investimento que existe no mercado. O estudo destes algoritmos e técnicas não é possível de realizar, pois estes são totalmente confidenciais uma vez que a chave do sucesso desta empresas depende dos mesmos.

De modo a poder explorar algumas das técnicas mais comuns para o investimento em ativos financeiros, um exemplo de uma plataforma comercial do mesmo género, mas muito menos sofisticada, é apresentada a seguir.

SmarttBot



Fig. 2.3: *Dashboard* da plataforma SmarttBot.

O SmarttBot [7] é uma plataforma virada para os investidores que querem automatizar as suas aplicações na bolsa de valores. Para isso, o investidor precisa entrar no site, registar uma estratégia nos robôs (a plataforma é totalmente online) e acompanhar o desempenho das regras escolhidas. Este pode desenhar as próprias estratégias ou comprar estratégias disponíveis numa Loja de Estratégias no site. Tudo é mostrado na interface gráfica do utilizador com as informações importantes, como:

- **Ordens enviadas** – para cada ordem é apresentado o tipo de operação (compra ou venda), ativo e o valor médio negociado, valor da negociação total, resultado (quanto ganho ou perdido do capital) e horário. Inclui as ordens executadas, alteradas, canceladas, rejeitadas e os detalhes.
- **Carteira de ativos negociados** – permite consultar o preço, o volume e o resultado parcial das posições abertas do robô, bem como Zerar Carteira, que encerra a posição aberta por aquele robô. Caso não esteja posicionado, a carteira fica zerada.
- **Gráficos de acompanhamento** – permite visualizar num gráfico onde o robô comprou e vendeu para verificar se ele está a agir conforme os sinais do indicador e dos stops. Para isso, os indicadores são definidos na parametrização do robô e configurados da mesma forma para a estratégia.
- **Verificação de saldo e custos** – apresenta informações referentes à análise de desempenho do robô. Permite visualizar o saldo, custos operacionais, o retorno, o risco e a análise dos investimentos realizados até o momento pelo mesmo.

- **Acompanhamento das ordens realizadas** – permite visualizar ao detalhe todas as ordens que foram executadas.
- **Configuração e otimização de estratégias** – permite especificar como o robô irá operar a partir de um indicador previamente escolhido. Permite ajustar stops, regras de compra e venda e outros itens quando quiser melhores resultados com o robô.

De notar que as melhores plataformas de negociação automatizadas compartilham algumas características comuns, tais como: **acesso aos mercados preferidos do investidor, interface funcional, linguagem de programação padrão da indústria e históricos de preços detalhados para *backtesting***.

2.4 Proposta

Com base no referido na secção anterior, a proposta que este documento apresenta é a realização de uma plataforma automatizada para a compra e venda de ativos financeiros com recurso a um algoritmo de tomada de decisão simples. De destacar que o **foco principal deste trabalho é a automatização do sistema** e este deve conter as seguintes funcionalidades:

- Proporcionar uma interface gráfica informativa, simples e eficaz ao utilizador;
- Manter um registo de todas as transacções feitas através da plataforma;
- Permitir ao utilizador executar ordens de compra e venda em tempo real, incluindo a alteração e a remoção da mesma;
- Proporcionar um acesso á carteira do utilizador, para consultar saldo, ativos negociados, entre outros;
- Recolher dados relevantes sobre o valor histórico de ativos negociáveis para a realização de *backtesting* de modo a influenciar de forma positiva a tomada de decisão.

Os tópicos destacados vão de encontro e satisfazem os objetivos principais propostos no Capítulo 1. Destaca-se também a possibilidade de acrescentar outras funcionalidades no desenrolar deste projeto de modo a dar suporte às funcionalidades principais. A proposta será abordada com maior detalhe mais a frente no Capítulo 3 deste documento.

2.5 Conclusões

Neste capítulo foi contextualizado o que são os ativos financeiros e como estes podem ser classificados, o que são os mercados financeiro e a evolução deste mercado para os dias de hoje, o que são softwares de negociação automatizados e a sua importância para o investimento em ativos financeiros, um exemplo de um sistema de negociação automatizado e a suas componentes principais, e por fim, a proposta do autor baseada no conteúdo abordado neste capítulo para realizar o sistema automatizado proposto neste documento.

Capítulo 3

Problema Proposto a Resolver

Neste capítulo será inicialmente contextualizado a corretora a usar, a IBKR, seguido pela abordagem proposta pelo autor e, por fim, uma conclusão sobre este capítulo.

3.1 *Interactive Brokers (IBKR)*

A IBKR [1] é uma corretora norte-americana que permite investir em ativos financeiros em várias bolsas de valores. Esta corretora dispõe de uma plataforma *online* que oferece aos investidores em todo o mundo um vasto catálogo de opções de investimento. Como esta corretora tem sucursais registadas em vários países, é possível investir através desta sem a necessidade de residir nos Estados Unidos. Através desta corretora temos acesso a varias bolsas de valores não só nos Estados Unidos como no resto do mundo. A Tabela 3.1 apresenta a lista de bolsa de valores que a corretora IBKR abrange.

Bolsa de Valores	País
<i>New York Stock Exchange (NYSE)</i>	EUA
<i>Toronto Stock Exchange (TSX)</i>	Canada
<i>Chicago Board Options Exchange (Cboe)</i>	EUA
<i>Bolsa Mexicana de Valores (BMV)</i>	México
Bolsa de Madrid (XMAD)	Espanha
<i>Swiss Exchange (SIX)</i>	Suíça
<i>London Stock Exchange (LSE)</i>	Grã-Bretanha
<i>Frankfurt Stock Exchange (FRA)</i>	Alemanha
<i>Budapest Stock Exchange (BSE)</i>	Hungria
<i>Moscow Stock Exchange (MOEX)</i>	Rússia
<i>Australia Stock Exchange (ASX)</i>	Austrália
<i>Tel Aviv Stock Exchange (TASE)</i>	Israel
<i>National Stock Exchange of India (NSE)</i>	Índia
<i>Singapore Stock Exchange (SGX)</i>	Singapura
<i>Hong Kong Stock Exchange (HKSE)</i>	Hong Kong
<i>Korea Stock Exchange (KSC)</i>	Coreia do Sul
<i>Tokyo Stock Exchange (TSE)</i>	Japão

Table 3.1: Lista de bolsas de valores para investir disponibilizadas pela corretora IBKR.

Como podemos ver a lista é enorme. No total a IBKR abrange 135 mercados em 33 países e em 23 moedas distintas. O acesso a estes mercados é possível através da plataforma TWS da IBKR. Esta plataforma é a porta principal para que o sistema proposto possa interagir com este mundo. A interação com esta plataforma é explicada com mais detalhe mais a frente no Capítulo 5.

3.2 Abordagem Proposta

Com base no conteúdo abordado nos Objetivos no Capítulo 1 e na Proposta no Capítulo 2, é então agora apresentado com detalhe a abordagem do autor ao problema. A abordagem proposta para realizar o sistema em questão passa por duas partes.

A primeira é criar um método para interagir com a corretora escolhida, a IBKR, para: recolher e trabalhar os dados relacionados com o valor dos ativos, executar ordens de compra e venda destes ativos e recolher dados relacionados com o utilizador da conta IBKR. Também será necessário recolher dados de fontes externas para trabalhar o algoritmo de tomada de decisão. Estes dados são os valores históricos de um determinado ativo. Desenvolver o algoritmo de tomada de decisão, testar-lo com estes valores históricos e analisar o seu comportamento. Este algoritmo de tomada de decisão será baseado num *Simple Medium Average* (SMA) [9]. Estes mecanismos vão ser inseridos no *backend* do sistema.

A segunda parte é criar a plataforma de acesso ao sistema, uma interface gráfica para o utilizador. Nesta plataforma será trabalhada a forma como os dados são apresentados ao utilizador, as diversas opções que este pode seleccionar como ver o valor de um determinado ativo, consultar a sua carteira, consultar registo de ordens executadas, entre outros. Estes mecanismos serão inseridos no *frontend* do sistema.

Até aqui é possível entender que para a realização bem sucedida do sistema são necessárias, na óptica do autor, ferramentas de desenvolvimento de *frontend* e *backend*, e também é necessário escolher uma linguagem de programação adequada para a interação com a plataforma da corretora IBKR. As opções de tecnologias a usar e a plataforma da corretora IBKR são abordadas com maior detalhe no Capítulo 5.

3.3 Conclusões

Neste capítulo foi contextualizado o que é a corretora IBKR e as diversas bolsas valores que esta oferece aos investidores, e por fim, a abordagem ao problema proposta pelo autor de modo a que a realização deste projeto seja bem sucedida.

Capítulo 4

Engenharia de Software

Neste capítulo é efectuado o levantamento dos requisitos funcionais e não funcionais, bem como dos casos de uso, a estrutura da base de dados e ainda a arquitectura do sistema.

4.1 Análise dos Requisitos e Casos de Uso

4.1.1 Requisitos Funcionais

Os requisitos funcionais da plataforma são os seguintes:

RF01 Permitir ao utilizador consultar o nome da conta sincronizada;

RF02 Permitir ao utilizar consultar a data e hora atual;

RF03 Permitir ao utilizador consultar se o mercado esta aberto ou fechado;

RF04 Permitir ao utilizador consultar o saldo da conta sincronizada;

RF05 Permitir ao utilizador consultar as posições da conta sincronizada;

RF06 Permitir ao utilizador consultar as ordens em aberto da conta sincronizada;

RF07 Permitir ao utilizador consultar as ordens enviadas através do sistema;

RF08 Permitir ao utilizador vender as posições da conta sincronizada ao preço do mercado ou a um preço limite;

RF09 Permitir ao utilizador consultar o valor atual do ativo selecionado;

RF10 Permitir ao utilizador consultar o valor histórico de um ativo selecionado;

RF11 Permitir ao utilizador consultar os dados recolhidos pelo algoritmo de negociação automatizado;

RF12 Permitir ao utilizador consultar as ordens enviadas pelo algoritmo de negociação automatizado;

RF13 Permitir ao utilizador consultar o comportamento do algoritmo de negociação automatizado;

RF14 Permitir ao utilizador ativar e desativar o algoritmo de negociação automatizado;

RF15 Permitir ao utilizador consultar a lista de ativos disponíveis para negociar;

RF16 Permitir ao utilizador cancelar ordens em aberto.

4.1.2 Requisitos Não-Funcionais

Por outro lado, os requisitos não funcionais do sistema são os seguintes:

RNFO1 A comunicação entre o sistema e a API da TWS deve estar sempre disponível;

RNFO2 O preço dos ativos deve ser sempre o preço atual;

RNFO3 A recolha dos preços históricos de um ativo deve ser feita tendo em conta o ultimo ano (incluindo o dia atual);

RNFO4 O sistema deve guardar, em tempo real, numa base de dados os dados recolhidos pelo algoritmo de negociação automatizado;

RNFO5 O sistema deve guardar, em tempo real, numa base de dados todas as ordens enviadas pelo algoritmo de negociação automatizado;

RNFO6 O sistema deve guardar numa base de dados todas as ordens enviadas manualmente pelo utilizador;

4.1.3 Casos de Uso

Nesta subsecção são apresentados os casos de uso do sistema. Tal como é apresentado na Figura 4.1 o utilizador pode: consultar a conta; consultar a data e hora; consultar abertura e fecho do mercado; ativar e desativar o *trading bot*; consultar o *trading bot*, onde o utilizador poderá consultar o histórico de preços e o registo de ordens enviadas; consultar preços, onde o utilizador poderá consultar o histórico de preços, os preços atuais e comprar posições; consultar o portefólio, onde o utilizador poderá consultar o seu saldo, a lista de ordens enviadas, as suas posições onde pode optar por vender-las e, por fim, consultar as ordens em aberto onde pode optar por cancelar ordens.

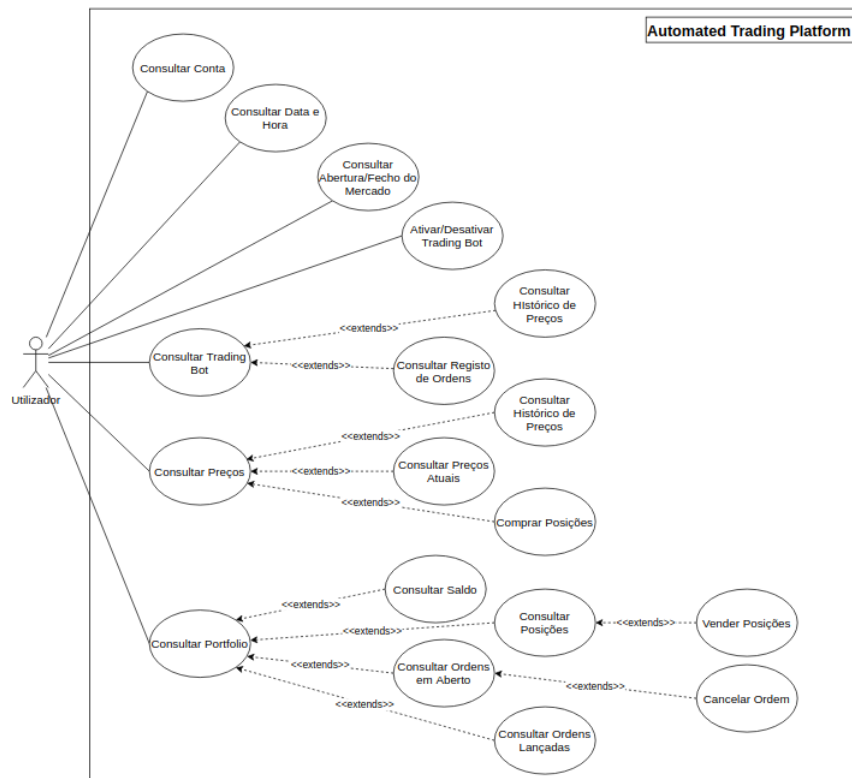


Fig. 4.1: Casos de uso da plataforma.

4.2 Base de Dados

Nesta subsecção é apresentado o esquema físico da base de dados do sistema. A base de dados é composta por duas tabelas não relacionadas. Tal como podemos observar na Figura 4.2, a primeira tabela (*orders*) que corresponde aos dados relativos a uma ordem enviada através do sistema é formada por 9 elementos. A segunda tabela (*records*) que corresponde a cada dado recolhido pelo algoritmo de negociação automatizado é composto por 7 elementos.

Orders	
PK	<u>id</u>
reqId	INT
time	DATETIME
symbol	STRING
action	STRING
quantity	INT
price	FLOAT
option	STRING
submitType	STRING

Records	
PK	<u>id</u>
time	DATETIME
symbol	STRING
price	FLOAT
sma30	FLOAT
sma100	FLOAT
action	STRING

Fig. 4.2: Estrutura da Base de Dados.

4.3 Arquitectura

Nesta subsecção é apresentado o diagrama de representação da arquitectura do sistema. Tal como podemos ver na Figura 4.3, o servidor do sistema satisfaz os pedidos vindos do cliente e gere a comunicação com o Sistema de Gestão de Base de Dados (SGBD), a API da TWS e a API do *Yahoo Finance*. Podemos também observar que a TWS se encarrega de submeter todas as ordens enviadas pelo sistema para rede da IBKR.

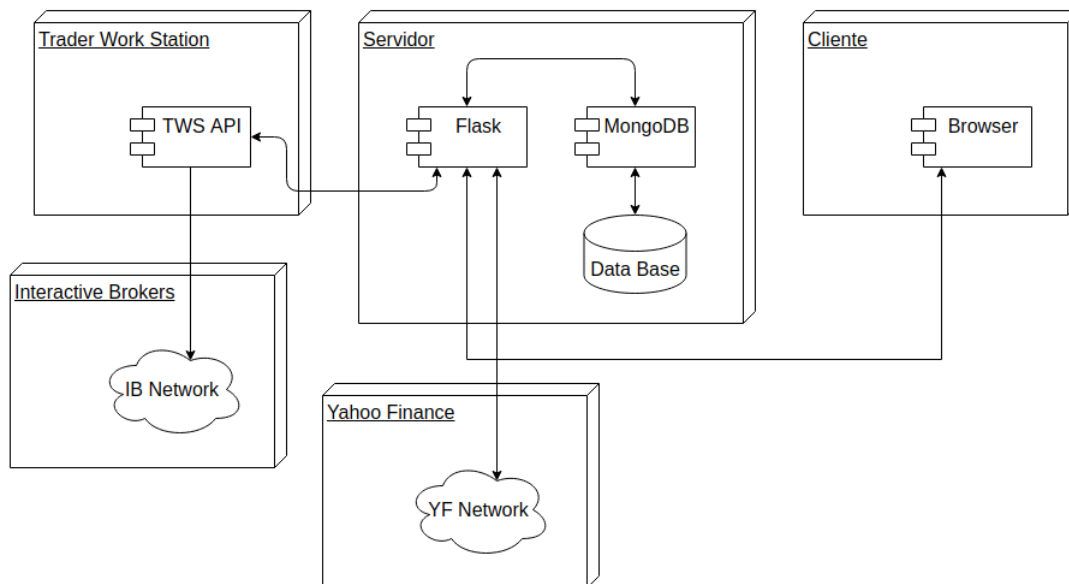


Fig. 4.3: Diagrama de representação da arquitectura do sistema.

4.4 Conclusões

Em suma, com o auxílio desta secção adquire-se uma ideia geral do sistema desenvolvido incluindo os seus requisitos, os casos de uso, a estrutura da base de dados e a arquitectura proporcionando um passo importante que serve como base para definir e entender o sistema em causa.

Capítulo 5

Metodologias Usadas e Resultados

Neste capítulo será inicialmente contextualizado a plataforma TWS corretora IBKR, seguido pela opção das tecnologias e ferramentas a usar para a realização deste projeto, as metodologias, os resultados e por fim, as conclusões principais sobre este capítulo.

5.1 Trader Workstation (TWS)

De modo a que os investidores possam aceder aos mercados disponibilizados pela corretora IBKR, esta oferece para este efeito a plataforma TWS disponível para Windows e Mac OS. A TWS é uma plataforma de *trading* completa, através da qual é possível comprar e vender ações, mas também negociar opções, negociar forex, futuros e outros produtos complexos. Como a IBKR é uma corretora muito usada por investidores, mas também por *day traders*, é normal que esta plataforma inclua funcionalidades para satisfazer as necessidades destes dois tipos de perfis. Quando falamos em **investidores** referimos a alguém que compra ações para vender algumas semanas, meses ou anos depois. Os **day traders** normalmente abrem e fecham posições no próprio dia ou em alguns casos, podem abrir uma posição e deixar até ao dia seguinte, também conhecido como **swing trade**. Na Figura 5.1 podemos observar o *layout* da *dashboard* da plataforma TWS.



Fig. 5.1: Dashboard da plataforma Trader Workstation (TWS) [1]

Esta plataforma também permite um acesso a sua API para que os *developers* possam desenvolver o software tirando proveito das opções que a TWS tem para oferecer. A interação com esta API está disponível em várias linguagens de programação, tais como Java,

C# e Python. Existe bastante documentação de suporte ao desenvolvimento de aplicações que interagem com a TWS para estas 3 linguagens. Para a finalidade deste projeto, o autor opta por usar a linguagem Python V3.9. Para além do suporte a esta linguagem através de documentação, o Python dispõe de várias ferramentas úteis para to tratamento de dados como o Numpy e o Pandas.

5.2 Tecnologias e Ferramentas Usadas

De modo a descrever esta secção, é introduzida primeiramente as línguas de programação, seguido pela base de dados, e por fim, as *frameworks* e *Integrated Development Environment* (IDE) consideradas a usar para o desenvolvimento deste projeto.

5.2.1 Linguagens de programação

Para a realização deste projeto foram escolhidas várias linguagens de programação para o desenvolvimento do *backend* e *frontend* do sistema. Para o *backend* vai ser usada a linguagem Python como referido na secção anterior e para o *frontend* vão ser usadas as linguagens *JavaScript* (JS), *TypeScript* (TS), *Hyper Text Markup Language* (HTML) e *Cascading Style Sheets* (CSS). Para a base de dados, o autor considera o uso do MongoDB como uma linguagem *No Structured Query Language* (NoSQL).

De modo a entender as linguagens escolhidas e qual a sua finalidade, segue-se um pouco de contexto relacionado com cada uma delas.

Python

O *Python*, criado por Guido van Rossum, é uma linguagem de programação criada para produzir um bom código e fácil de manter de maneira rápida. Enumeram-se as seguintes características desta linguagem:

- baixo uso de caracteres especiais, tornando assim o código muito mais limpo e de leitura mais fácil;
- uso de indentação para marcar blocos;
- fácil compilação;

Além disso, o *Python* suporta técnicas de orientação a objetos, tem uma biblioteca padrão imensa, que contém classes, métodos e funções para realizar essencialmente qualquer tarefa, dispõe de ferramentas para lidar com dados científicos e também possui inúmeras capacidades de meta-programação. A versão de *Python* usada neste projeto é a versão 3.9.

TypeScript (TS)

O TS, criado pela *Microsoft*, dispõe de recursos que melhor suportam o uso da Programação Orientada a Objetos, que tem como base quatro princípios fundamentais: encapsulamento, herança, abstração e polimorfismo. Esta linguagem possui uma sintaxe mais simplificada e mais clara, em comparação com o JS, e é amplamente suportada por IDE de código modernos.

JavaScript (JS)

O JS é uma linguagem de programação do lado cliente, ou seja, é processada pelo próprio navegador. Com o JS podemos criar efeitos especiais nas páginas Web, além de podermos proporcionar uma maior interatividade com os utilizadores. O JS é uma linguagem orientada a objetos, tratando todos os elementos da página como objetos distintos, facilitando a tarefa da programação.

Resumindo, o JS é uma poderosa linguagem que deve ser dominada por quem deseja criar páginas Web dinâmicas e interativas.

Hyper Text Markup Language (HTML)

O HTML, criado por Tim Barners Lee, consiste numa linguagem de marcação utilizada para produção de páginas na *web*, que permite a criação de documentos que podem ser acedidos por qualquer dispositivo informático via Internet.

As *tags* presentes nesta linguagem servem para indicar a função de cada elemento da página *web*, e funcionam como comandos de formatação de textos, formulários, *links*, tabelas, etc. Estas são interpretadas pelos *browsers* que são responsáveis por apresentar a página *web*.

Cascading Style Sheets (CSS)

O CSS é uma "folha de estilo" composta por "camadas" e utilizada para definir a apresentação (aparência) em páginas da *internet* que adotam para o seu desenvolvimento linguagens de marcação (como *Extensible Markup Language (XML)*, HTML e *eXtensible Hypertext Markup Language (XHTML)*). O CSS define como serão exibidos os elementos contidos no código de uma página *web*, e a sua maior vantagem é efetuar a separação entre o formato e o conteúdo de um documento.

5.2.2 Base de Dados

Para a criação da base de dados deste projeto, considera-se usar o SGBD NoSQL do MongoDB.

MongoDB

O MongoDB é um software de base de dados orientado a documentos livre, de código aberto e multiplataforma, escrito na linguagem C++. Classificado como um programa de base de dados NoSQL, o MongoDB usa documentos semelhantes a *JavaScript Object*

Notation (JSON) com esquemas. É desenvolvido pela MongoDB Inc. e publicado sob uma combinação da GNU Affero General Public License e Licença Apache.

As suas características permitem com que as aplicações modelem informações de modo muito mais natural, pois os dados podem ser agrupados em hierarquias complexas e continuar a ser indexáveis e fáceis de buscar.

5.2.3 Frameworks

Para a realização deste projeto é necessária a utilização de algumas *frameworks* para garantir o sucesso do mesmo. As seguintes *frameworks* foram implementadas em diferentes componentes do projeto.

Para o *frontend* vai ser utilizado o *Angular* para criar e compilar todos os componentes da aplicação *web* e o *Bootstrap* para utilizar elementos específicos HTML e CSS.

Para o *backend* vai ser utilizado o *Flask* do Python onde vão ser criados os *endpoints* para satisfazer os pedidos *Hypertext Transfer Protocol* (HTTP) que vem do *frontend*, e onde vai ser incorporado os métodos para a interação com a base de dados e com a TWS.

Angular

O *Angular* é uma plataforma e *framework* para construção da interface de aplicações usando HTML, CSS e, principalmente, JS, criada pela Google.

Ele possui alguns elementos básicos que tornam essa construção interessante. Dentre os principais, podemos destacar os componentes, *templates*, diretivas, roteamento, módulos, serviços, injeção de dependências e ferramentas de infraestrutura que automatizam tarefas, como a de executar os testes unitários de uma aplicação.

Com Angular podemos criar *Single-Page Applications* com uma grande qualidade e produtividade, resultando numa experiência mais fluida para o utilizar quando interage com a página *web*.

PrimeNG

O *PrimeNG* é um *framework* utilizado para criar *layouts* e telas de sistemas *web*, *dashboards*, sites responsivos e sites comuns também. Esta *framework* facilita muito o trabalho de *frontend* com a sua vasta biblioteca para o desenho de elementos HTML e CSS.

Flask

Flask é um pequeno *framework web* escrito em Python. É classificado como um *microframework* porque não requer ferramentas ou bibliotecas particulares, mantendo um núcleo simples, porém, extensível. Este não possui camada de abstração de base de dados, validação de formulário ou quaisquer outros componentes onde bibliotecas de terceiros pré-existentes fornecem funções comuns. No entanto, o Flask oferece suporte a extensões que podem adicionar recursos da aplicação como se fossem implementados no próprio Flask.

Existem extensões para mapear objetos-relacionais, validação de formulário, manipulação de *upload*, várias tecnologias de autenticação aberta e várias ferramentas comuns relacionadas ao *framework*.

Visual Studio Code (VSCode)

O VSCode é um IDE que inclui suporte para depuração, controle *Git* incorporado, realce de sintaxe, terminal integrado, complementação inteligente de código e fatores de código. Este editor possui todas as funcionalidades necessárias para a escrita do código necessário.

5.3 Metodologias

Nesta secção são explicadas ao detalhe as metodologias usadas para a realização do sistema. São abordados os processos de todas as funcionalidades do sistema e a forma como foram implementadas. Dentro destas funcionalidades destacam-se a obtenção do portefólio, das posições, das ordens em aberto, das ordens enviadas e dos dados históricos relativos aos ativos e ao dados recolhidos pelo algoritmo de negociação automatizado. É descrito o processo para o cancelamento de ordens, envio de ordens, a inicialização do algoritmo de negociação automatizado e como este está implementado.

5.3.1 Obtenção dos dados do Portefólio

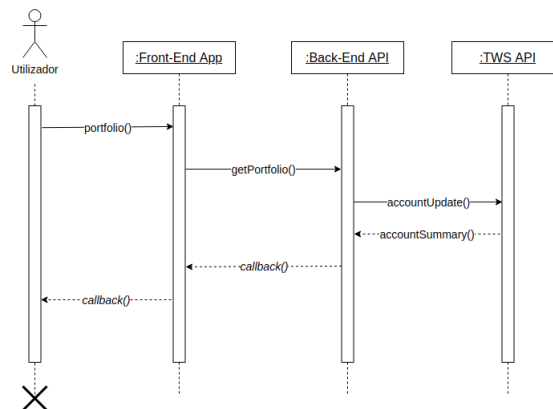


Fig. 5.2: Diagrama de fluxo para obter dados do portefólio.

Como podemos ver na Figura 5.2, a obtenção dos dados é realizada em 6 passos. No primeiro passo, o utilizador solicita os dados através da aplicação que envia um pedido à API do servidor no segundo passo. Em seguida, no terceiro passo, o servidor envia um pedido à API da TWS para recolher os dados. No quarto passo, a TWS responde ao pedido do servidor enviando os dados relativos ao portefólio da conta. No quinto passo, o servidor envia os dados obtidos para a aplicação onde por fim, no sexto passo, os dados do portefólio são apresentados ao utilizador.

5.3.2 Obtenção das Posições

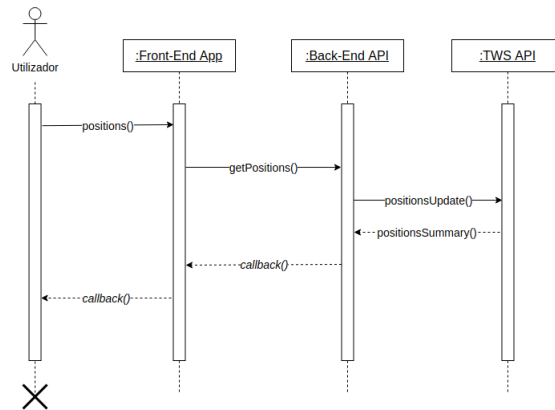


Fig. 5.3: Diagrama de fluxo para obter as posições da conta.

Como podemos ver na Figura 5.3, as posições da conta sincronizada são obtidas em 6 passos. No primeiro passo, o utilizador solicita as posições através da aplicação que envia um pedido à API do servidor no segundo passo. Em seguida, no terceiro passo, o servidor envia um pedido à API da TWS para recolher as posições. No quarto passo, a TWS responde ao pedido do servidor enviando os dados relativos às posições da conta. No quinto passo, o servidor envia os dados obtidos para a aplicação onde por fim, no sexto passo, as posições são apresentadas ao utilizador.

5.3.3 Obtenção das Ordens em Aberto

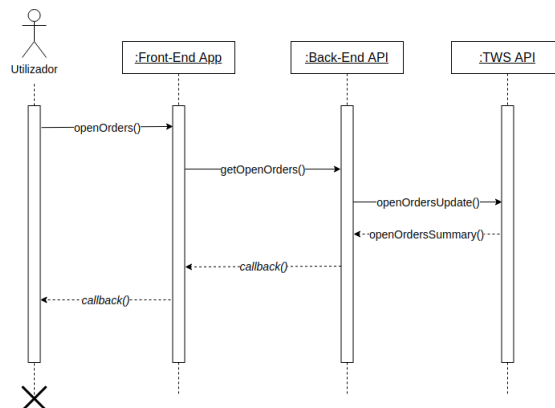


Fig. 5.4: Diagrama de fluxo para obter as ordens em aberto.

Como podemos ver na Figura 5.4, as ordens em aberto da conta sincronizada são obtidas em 6 passos. No primeiro passo, o utilizador solicita as ordens em aberto através da aplicação que envia um pedido à API do servidor no segundo passo. Em seguida, no terceiro passo, o servidor envia um pedido à API da TWS para recolher as ordens em aberto. No quarto passo, a TWS responde ao pedido do servidor enviando os dados relativos às ordens em aberto da conta. No quinto passo, o servidor envia os dados obtidos para a aplicação onde por fim, no sexto passo, as ordens em aberto são apresentadas ao utilizador.

5.3.4 Obtenção das Ordens Enviadas

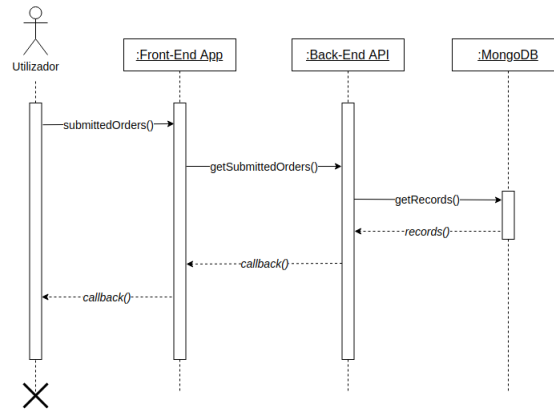


Fig. 5.5: Diagrama de fluxo para obter as ordens submetidas.

Como podemos ver na Figura 5.5, as ordens enviadas através do sistema são obtidas em 6 passos. No primeiro passo, o utilizador solicita as ordens enviadas através da aplicação que envia um pedido à API do servidor no segundo passo. Em seguida, no terceiro passo, o servidor envia um pedido ao SGBD do *MongoDB* para recolher as ordens enviadas. No quarto passo, o SGBD responde ao pedido do servidor enviando os dados relativos às ordens enviadas. No quinto passo, o servidor envia os dados obtidos para a aplicação onde por fim, no sexto passo, as ordens enviadas são apresentadas ao utilizador.

5.3.5 Obtenção dos dados recolhidos pelo *Trading Bot*

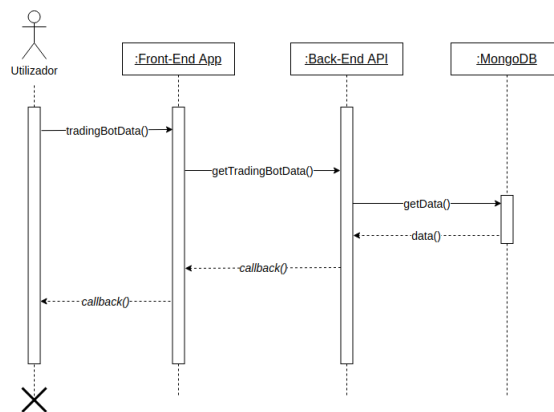


Fig. 5.6: Diagrama de fluxo para obter os dados recolhidos pelo algoritmo de negociação automatizada.

Como podemos ver na Figura 5.21, os dados recolhidos pelo algoritmo de negociação automatizado são obtidas em 6 passos. No primeiro passo, o utilizador solicita os dados recolhidos pelo algoritmo através da aplicação que envia um pedido à API do servidor no segundo passo. Em seguida, no terceiro passo, o servidor envia um pedido ao SGBD do *MongoDB* para recolher os dados do algoritmo. No quarto passo, o SGBD responde ao pedido do servidor enviando os dados relativos à análise do algoritmo. No quinto passo, o

servidor envia os dados obtidos para a aplicação onde por fim, no sexto passo, as os dados recolhidos pelo algoritmo são apresentados ao utilizador.

5.3.6 Obtenção de Dados Históricos

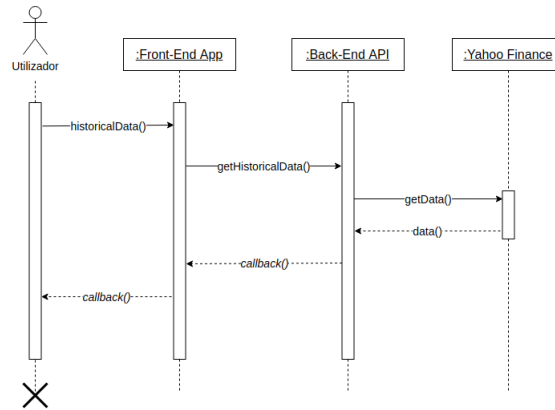


Fig. 5.7: Diagrama de fluxo para obter os dados históricos de um determinado ativo.

Como podemos ver na Figura 5.7, os dados históricos de ativos são obtidos em 6 passos. No primeiro passo, o utilizador solicita os dados históricos de um determinado ativo através da aplicação que envia um pedido à API do servidor no segundo passo. Em seguida, no terceiro passo, o servidor envia um pedido à API do *Yahoo Finance* para recolher os dados históricos. No quarto passo, a API responde ao pedido do servidor enviando os dados relativos aos dados históricos. No quinto passo, o servidor envia os dados obtidos para a aplicação onde por fim, no sexto passo, os dados históricos são apresentadas ao utilizador.

5.3.7 Envio de Ordens

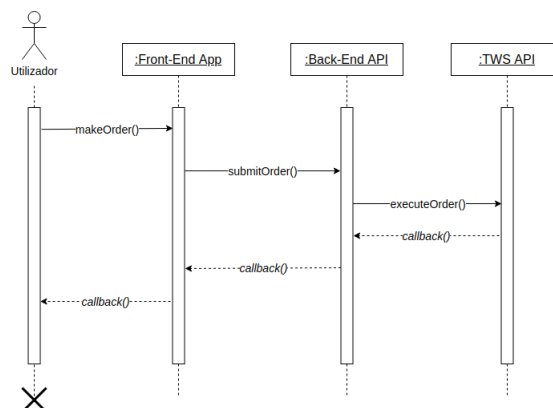


Fig. 5.8: Diagrama de fluxo para submeter uma ordem.

Como podemos ver na Figura 5.8, a realização de uma ordem é feita em 6 passos. No primeiro passo, o utilizador escolhe o tipo de ordem (compra ou venda) a ser enviada (incluindo o ativo, o número de posições e a opção) através da aplicação que envia a mesma

ordem à API do servidor no segundo passo. Em seguida, no terceiro passo, o servidor envia a ordem a ser realizada à API da TWS para executar a ordem enviada. No quarto passo, a TWS responde com sucesso ou insucesso. No quinto passo, o servidor reencaminha a resposta onde por fim, no sexto passo, a resposta é apresentada ao utilizador.

5.3.8 Cancelamento de Ordens

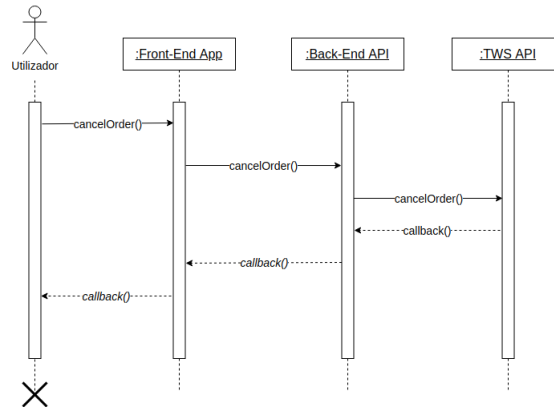


Fig. 5.9: Diagrama de fluxo para cancelar uma ordem.

Como podemos ver na Figura 5.9, o cancelamento de uma ordem é feita em 6 passos. No primeiro passo, o utilizador escolhe a ordem em aberto que pretende cancelar através da aplicação que envia a mesma ordem à API do servidor no segundo passo. Em seguida, no terceiro passo, o servidor envia a ordem a ser cancelada à API da TWS para executar o cancelamento. No quarto passo, a TWS responde com sucesso ou insucesso. No quinto passo, o servidor reencaminha a resposta onde por fim, no sexto passo, a resposta é apresentada ao utilizador.

5.3.9 Funcionamento do *Trading Bot*

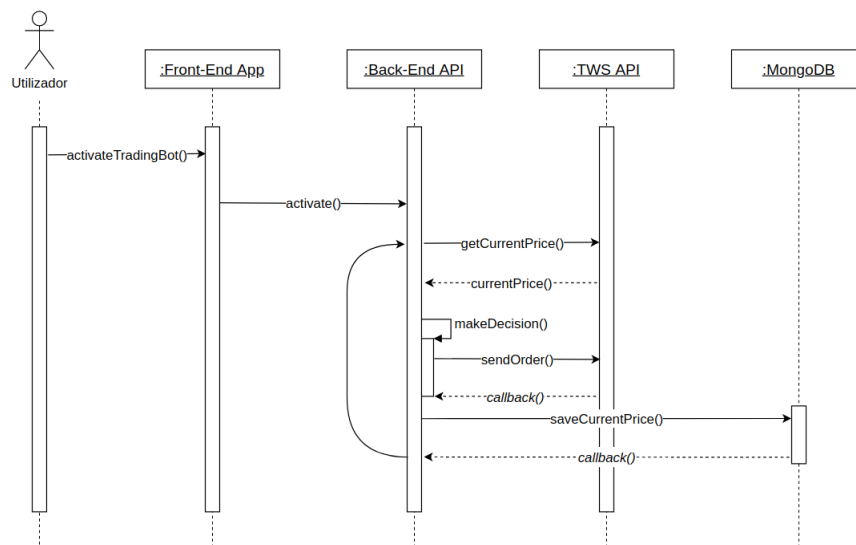


Fig. 5.10: Diagrama de fluxo para inicializar o algoritmo de negociação automatizada.

Como podemos ver na Figura 5.10, a ativação do algoritmo de negociação automatizado é realizada em 2 passos. No primeiro passo, o utilizador escolhe ativar o *trading bot* através da aplicação que envia um pedido de ativação para a API do servidor no segundo passo. Em seguida, o servidor inicializa o algoritmo iniciando um processo cíclico de 8. No primeiro passo, o algoritmo envia um pedido do preço atual do ativo em negociação à API da TWS que envia uma resposta ao pedido com o preço atual no segundo passo. Estes 2 passos são repetidos 300x em intervalos de 10 segundos. Após obter estes dados, são calculadas as médias móveis de 30 e 100. Uma vez calculadas, o algoritmo toma uma decisão tendo em conta a posição destas duas médias no terceiro passo. Se a média de 30 no instante atual for maior que a média de 100, o algoritmo envia uma ordem de compra para a API da TWS no quarto passo, recebendo a conclusão da ordem no quinto passo. Caso contrário o algoritmo emite uma ordem de venda. Em seguida no sexto passo, o servidor envia o dado recolhido juntamente com a ordem enviada para o SGBD do *MongoDB* que se encarrega de guardar o dado na base de dados, enviando uma resposta ao servidor que o dado foi guardado no sétimo passo. Por fim no oitavo passo, o algoritmo volta ao terceiro passo mas desta vez salta a etapa de recolher os 300 dados fechando assim o ciclo. Este ciclo é repetido em intervalos de 10 segundos, consequentemente emitindo ordens de 10 em 10 segundos. Uma vez que o algoritmo emite uma ordem de venda, este não vai emitir mais nenhuma ordem até que a ordem a ser enviada seja uma ordem de venda, e vice versa. O utilizador pode interromper o algoritmo em qualquer instante. Neste caso, na próxima ativação o algoritmo recomeça o de novo processo descrito a nesta subsecção.

5.4 Funcionalidades da Aplicação

Nesta secção é apresentado todas as componentes da aplicação desenvolvida para o utilizador, juntamente com instruções para explicar o correto funcionamento da aplicação com o sistema desenvolvido.

5.4.1 Página Inicial

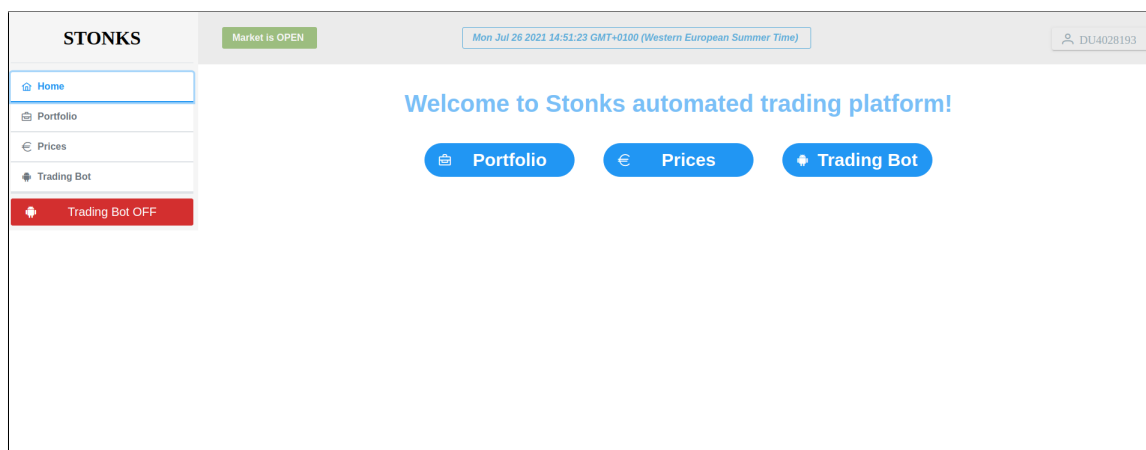


Fig. 5.11: Página inicial da aplicação.

Como podemos ver na Figura 5.11, o utilizador é recebido com uma mensagem de boas-vindas e três opções de consulta, portfólio, preços e *trading bot*. Na barra superior, é apresentada a informação sobre o estado do mercado (aberto ou encerrado), a data e hora atual do sistema e o nome da conta IBKR sincronizada com o sistema. No lado esquerdo o utilizador pode navegar pelas três opções de consulta mencionadas anteriormente, incluindo a página principal. Ainda no menu lateral, o utilizador pode inicializar e verificar o estado do *trading bot*.

5.4.2 Portefólio

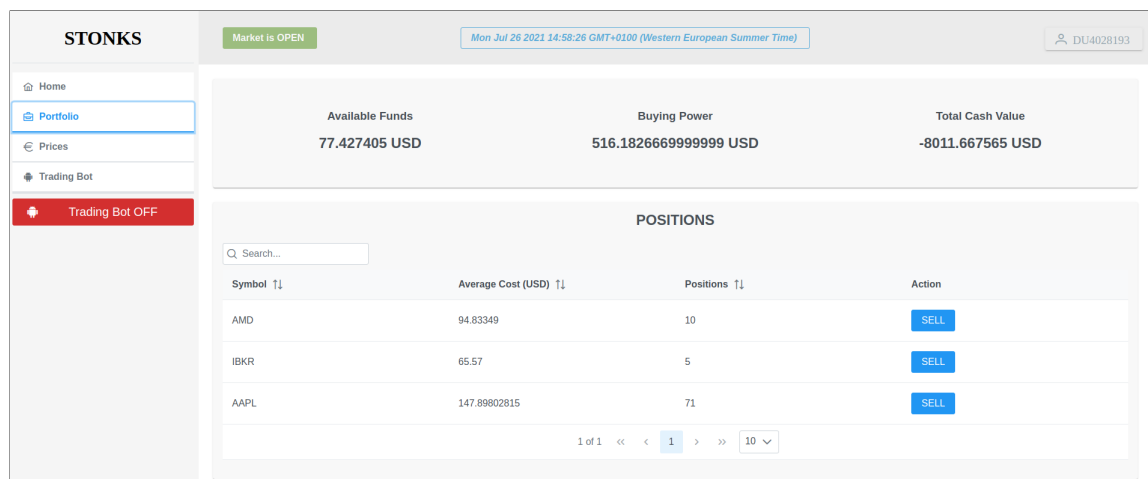


Fig. 5.12: Página do portefólio.

Como podemos ver na Figura 5.12, o utilizador pode consultar os dados relativos ao saldo da conta sincronizada. É também apresentada uma listagem das posições da conta e ainda, uma listagem das ordens em aberto e das ordens enviadas através do sistema.

5.4.2.1 Tabela de Posições

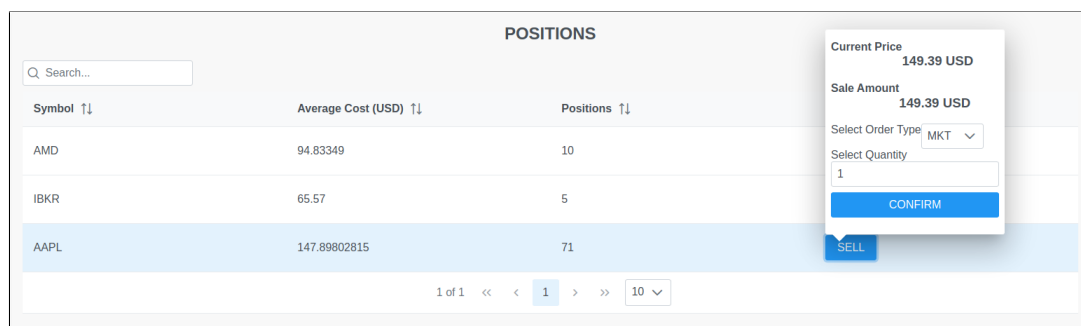


Fig. 5.13: Tabela de posições.

Nas tabela de posições, representada na Figura 5.13, o utilizador pode consultar todas as posições da conta. São apresentadas informações como o símbolo do ativo, o custo médio e o número de posições. Para cada ativo o utilizador pode optar por vender as suas posições numa *po-up* onde é mostrado o preço atual do ativo selecionado, o valor total da

venda, o tipo de venda que pretende submeter e a quantidade de posições que pretende vender.



Fig. 5.14: Opções de venda.

Na Figura 5.14 são apresentadas as opções de venda que o utilizador pode efectuar. Na Figura 5.14a o utilizador está a realizar uma venda ao preço do mercado, apenas precisa de seleccionar a quantidade e submeter a venda. Na Figura 5.14b o utilizador está a realizar uma venda com preço limite, apenas precisa de indicar a que preço pretende vender e a quantidade antes de submeter a venda. Após submeter a venda, é apresentada uma mensagem com os valores que o utilizador introduziu semelhante ao que é apresentado na Figura 5.14c.

5.4.2.2 Tabela de Ordens em Aberto

OPEN ORDERS								
Order ID ↑↓	Permission ID ↑↓	Order Type ↑↓	Limit Price ↑↓	Action ↑↓	Symbol ↑↓	Total Quantity ↑↓	Status ↑↓	Action
3758	313622819	LMT	150	SELL	AAPL	1	Submitted	CANCEL

1 of 1 << < 1 > >> 10 ↓

Fig. 5.15: Tabela de ordens em aberto.

Como podemos ver na Figura 5.15, o utilizador pode consultar todas as ordens que no momento estão em aberto. Para cada ordem o utilizador pode consultar o ID da ordem, o ID de permissão, o tipo da ordem, o preço limite estabelecido, a ação, o símbolo do ativo, a quantidade e o status da ordem. Cada ordem tem um opção para o seu cancelamento, basta o utilizador clicar no botão cancelar e a ordem é cancelada.

5.4.2.3 Tabela de Ordens Enviadas

ORDERS SENT							
Q Search...							
ReqID ↑↓	Date ↑↓	Symbol ↑↓	Action ↑↓	Quantity ↑↓	Price (USD) ↑↓	Option ↑↓	Order Type ↑↓
8084	2021-07-26 15:16:50.384104+01:00	AAPL	SELL	1	149.57	MKT	MANUAL
8078	2021-07-26 15:16:06.417048+01:00	AAPL	SELL	1	149.54	MKT	MANUAL
8070	2021-07-23 16:17:33.331910+01:00	AAPL	BUY	10	148	MKT	BOT
8058	2021-07-23 16:15:53.361550+01:00	AAPL	SELL	10	147.96	MKT	BOT
8019	2021-07-23 16:09:42.906212+01:00	AAPL	BUY	10	148.04	MKT	BOT
7954	2021-07-23 15:59:12.450229+01:00	AAPL	SELL	10	147.91	MKT	BOT
7785	2021-07-23 15:31:21.110672+01:00	AAPL	BUY	10	147.92	MKT	BOT
7767	2021-07-23 15:28:41.030571+01:00	AAPL	SELL	10	147.69	MKT	BOT
7744	2021-07-23 15:25:10.954645+01:00	AAPL	BUY	10	147.82	MKT	BOT
7686	2021-07-23 15:15:50.655788+01:00	AAPL	SELL	10	147.72	MKT	BOT

1 of 6 << < 1 2 3 4 5 > >> 10 ▾

Fig. 5.16: Tabela de ordens enviadas.

Como podemos ver na Figura 5.16, o utilizador tem acesso a uma tabela com o registo de todas as ordens enviadas pelo sistema, incluindo ordens manuais ou automáticas (realizadas pelo algoritmo de negociação automatizado). Cada registo contém informações como o ID do requerimento, a data e hora da ordem (*timezone* de Lisboa), o símbolo do ativo, a ação realizada, a quantidade, o preço do ativo na altura, a opção da ordem (ao mercado ou ao limite) e o tipo de ordem (manual ou automática).

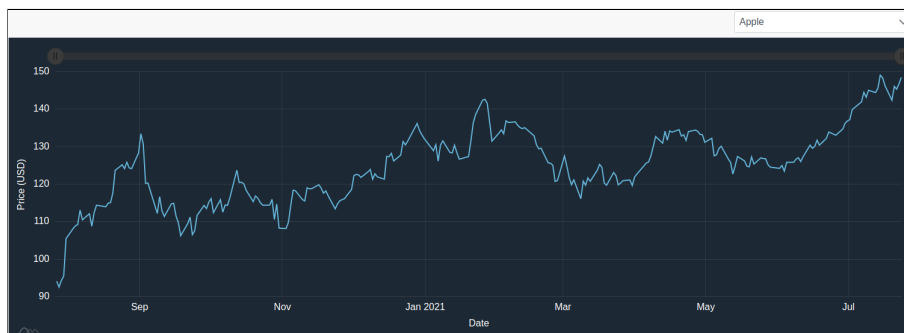
5.4.3 Preços



Fig. 5.17: Página de preços.

Na Figura 5.17 podemos ver a página para a consulta de preços e compra de posições. Nesta página o utilizador pode consultar os preços históricos de qualquer ativo com o recurso a um gráfico interactivo e reactivo. Este também pode consultar o preço atual de qualquer ativo e realizar ordens de compra.

5.4.3.1 Histórico de Preços



(a) Histórico de preços da *Apple*.



(b) Histórico de preços da *Amazon*.

Fig. 5.18: Gráfico de consulta de preços históricos.

Como podemos ver na Figura 5.18, o utilizador pode navegar entre diferentes ativos e consultar o seu preço histórico num intervalo de um ano tendo em conta o dia atual. Os preços apresentados são os preços da hora de fecho do mercado de cada dia. O utilizador pode filtrar o gráfico em qualquer posição e analisar o comportamento para cada mes ou semana.

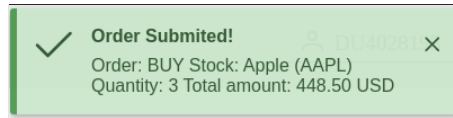
5.4.3.2 Comprar Posições

The screenshot shows a 'Place Order' form for Apple stock. The 'Select Stock' dropdown is set to 'Apple'. The 'Select Order Type' dropdown is set to 'MKT'. The 'Select Quantity' input field contains the number '2'. A blue 'BUY' button is visible at the bottom. To the right of the form, the following prices are displayed: Bid Price: 149.51 USD, Ask Price: 149.51 USD, and Purchase Amount: 299.02 USD.

(a) Comprar posições ao valor do mercado.

The screenshot shows a 'Place Order' form for Amazon stock. The 'Select Stock' dropdown is set to 'Amazon'. The 'Select Order Type' dropdown is set to 'LMT'. The 'Select Limit Price' input field contains the number '3680'. The 'Select Quantity' input field contains the number '2'. A blue 'BUY' button is visible at the bottom. To the right of the form, the following prices are displayed: Bid Price: 3690.18 USD, Ask Price: 3691.38 USD, and Purchase Amount: 7382.76 USD.

(b) Comprar posições a um valor limite.



(c) Notificação de compra.

Fig. 5.19: Opções de compra.

Como podemos ver na Figura 5.19, o utilizador pode optar por efectuar uma compra ao valor do mercado ou a um valor limite pré-estabelecido, podendo também consultar o valor atual do ativo selecionado e o valor da compra. Na Figura 5.19a o utilizador está a realizar uma compra ao valor do mercado, apenas precisa de escolher o ativo e o número de posições que pretende comprar e submeter a ordem compra. Na Figura 5.19b o utilizador está a realizar uma compra com um valor limite, apenas precisa de seleccionar o ativo, definir o limite de compra, o número de posições que pretende adquirir a aquele preço e submeter a ordem de compra. No fim de submeter a ordem de compra, uma *popup* irá aparecer no canto superior direito da aplicação contendo as informações sobre a ordem enviada tal como podemos ver na Figura 5.19c.

5.4.4 Trading Bot



Fig. 5.20: Página do *trading bot*.

Na Figura 5.20 podemos ver a página de consulta do *trading bot*. Nesta página o utilizador pode examinar com recurso a um gráfico os valores recolhidos pelo algoritmo em tempo real, proporcionando uma visão gráfica do comportamento do algoritmo. O gráfico é reactivo e interactivo para possibilitar a filtragem dos dados para o utilizador poder focar em zonas de interesse.

5.4.4.1 Dados Recolhidos

RECORDS					
Date ↑↓	Symbol ↑↓	Price (USD) ↑↓	SMA30 ↑↓	SMA100 ↑↓	Action ↑↓
2021-07-23 16:28:45.690558	AAPL	147.96	147.97533333333337	147.95209999999986	None
2021-07-23 16:28:35.685950	AAPL	147.96	147.979	147.95189999999985	None
2021-07-23 16:28:25.679640	AAPL	147.93	147.98266666666666	147.95219999999986	None
2021-07-23 16:28:15.669962	AAPL	147.93	147.98733333333333	147.95279999999985	None
2021-07-23 16:28:05.665999	AAPL	147.93	147.98966666666664	147.95309999999984	None
2021-07-23 16:27:55.658122	AAPL	147.93	147.98966666666666	147.95359999999985	None
2021-07-23 16:27:45.651510	AAPL	147.9	147.99099999999996	147.95389999999983	None
2021-07-23 16:27:35.641999	AAPL	147.88	147.994	147.95469999999986	None
2021-07-23 16:27:25.632763	AAPL	147.83	147.99766666666665	147.95559999999983	None
2021-07-23 16:27:15.622876	AAPL	147.83	148.00166666666666	147.95739999999984	None

Fig. 5.21: Tabela de dados recolhidos pelo algoritmo de negociação automatizado.

Para além do gráfico, o utilizador pode também consultar os dados recolhidos numa tabela atualizada em tempo real tal como podemos ver na Figura 5.21. Cada dado recolhido pelo algoritmo contém informações relevantes como a data e hora da recolha, o sim-

bolo do ativo, o preço na altura, a média dos 30, a médias dos 100 e a ação (compra, venda ou *null*) tomada pelo algoritmo.

5.5 Resultados

Nesta secção são contextualizados os dados recolhidos pelo algoritmo de negociação automatizado e o seu comportamento.

DIA 1			DIA 2			DIA 3		
BUY	143.36		BUY	146.66		BUY	147.66	
SELL	143.38	+0.02	SELL	146.92	+0.26	SELL	147.52	-0.14
BUY	143.68		BUY	147.17		BUY	147.42	
SELL	144.09	+0.41	SELL	147.70	+0.53	SELL	147.72	+0.30
BUY	144.09		BUY	147.97		BUY	147.82	
SELL	143.91	-0.18	SELL	147.73	-0.24	SELL	147.69	-0.13
BUY	144.07		BUY	147.51		BUY	147.92	
SELL	144.53	+0.46	SELL	147.33	-0.18	SELL	147.91	-0.01
			BUY	147.39		BUY	148.04	
			SELL	147.29	-0.10	SELL	147.96	-0.08
			BUY	146.97				
			SELL	146.68	-0.29			
			BUY	147.01				
			SELL	147.12	+0.11			
TOTAL		+0.71			+0.09			-0.06

Table 5.1: Listagem de ordens submetidas pelo algoritmo de negociação automatizado.

Na Tabela 5.1 temos os resultados das ordens emitidas pelo algoritmo de negociação automatizado em três dias separados. O algoritmo esteve ativo durante 3-4 horas no período de mercado aberto. Ao analisar os dados obtidos presentes na tabela podemos verificar que o algoritmo, em suma, teve um comportamento positivo, garantindo lucro no conjunto dos três dias. Mesmo tendo obtido lucro, podemos concluir que o comportamento do algoritmo, apesar de positivo, não é o ideal, mostrando ter tipo muitas perdas sucessivas em particular no segundo e terceiro dia. Contudo, não é do interesse para este projeto a eficácia do algoritmo de negociação. O objetivo é usar um algoritmo simples que apesar de não ser o ideal poder ser usado para um sistema de negociação automatizado executando as ordens pretendidas.

5.6 Conclusões

Neste capítulo foi apresentada a plataforma TWS da IBKR, as tecnologias usadas para o desenvolvimento deste projeto, as linguagens de programação e a contextualização do SGBD NoSQL do *MongoDB*. Foram descritas as metodologias usadas para a realização do sistema incluindo o funcionamento do algoritmo de negociação, foi descrito um manual de utilização de modo a entender as funcionalidades que a aplicação oferece ao utilizador e, por fim, os resultados obtidos pelo algoritmo de negociação automatizada.

Capítulo 6

Reflexão Crítica e Problemas Encontrados

Neste capítulo são apresentados os objetivos propostos vs. alcançados após a elaboração do projeto. Também é mencionada uma reflexão crítica retirada com o desenvolver do projeto.

6.1 Objetivos Propostos vs. Alcançados

Os objetivos propostos na Secção 1.2 do Capítulo 1 foram todos alcançados com sucesso. Na Tabela 6.1 podemos verificar os objetivos alcançados na realização do projeto e que o sistema desenvolvido possui como funcionalidades.

Objetivos	Cumprido
Realizar um sistema completamente automatizado	✓
Realizar um algoritmo de decisão simples para gerar indicações de compra e venda de ativos financeiros;	✓
Interagir em tempo real com a API da IBKR para lançar ordens de compra e venda, ou para modificar ou retirar;	✓
Registrar e apresentar de todas as ordens realizadas de compra e venda dos ativos financeiros;	✓
Recolher informação contextual adicional de uma fonte externa;	✓
Proporcionar ao utilizador uma interface informativa e eficaz.	✓

Table 6.1: Lista de objetivos alcançados.

6.2 Problemas Encontrados e Reflexão Crítica

Ao longo do desenvolvimento do projeto, foram encontrados alguns obstáculos que dificultaram a realização de algumas tarefas do mesmo. A destacar, a conexão com a API da TWS, visto que a interação com esta requer um bom entendimento das vastas funcionalidades que proporciona ao desenvolvedor. A recolha de dados como o portefólio, posições e preços de ativos revelaram-se desafiantes para desenvolver uma logia funcional para a recolha destes dados. Outro ponto desafiante foi o envio de ordens de compra, venda e cancelamento, entender como gerar um contrato e executar ordens de vários tipos para a TWS era um ponto chave para alcançar este objetivo. A sincronização dos preços apresentados na aplicação e os que aparecerem na TWS também se revelou desafiante, atualmente o utilizador consegue ver o preço atual na altura que vai realizar uma ordem, mas entre o clicar no botão para submeter e a ordem ser executada no lado da TWS o preço do

ativo pode mudar. Para o algoritmo de negociação automatizado este problema também é notável mesmo enviando ordens muito mais rápido que o utilizador comum.

Ainda assim, e refletindo sobre o projeto desenvolvido, o autor revela-se satisfeito com o trabalho realizado tendo cumprido todos os objetivos propostos. O projeto desenvolvido revela ter uma base forte para expandir no futuro, podendo albergar numerosas funcionalidades com base nos dados já recolhidos pelo sistema. Algumas das funcionalidades são descritas na Secção 7.2 do Capítulo 7.

6.3 Conclusão

Este capítulo salienta os objetivos alcançados na realização do projeto, assim como os obstáculos que o autor enfrentou o no decorrer deste. A reflexão crítica apresenta uma opinião pessoal face a toda uma abordagem geral acerca do sistema de negociação automatizado, também refletindo o ponto de vista do autor por parte do que foi elaborado neste projeto.

Capítulo 7

Conclusão

Neste capítulo é apresentada as conclusões principais em relação ao projeto desenvolvido e ainda alguns pontos a ter em conta como trabalho futuro para o crescimento do sistema realizado.

7.1 Conclusões Principais

Construir este projeto envolveu muito estudo sobre o funcionamento dos mercados financeiros e de estratégias para negociação automatizada. Apesar da existência de alguns impasses na realização do mesmo, o autor sente que adquiriu experiência, não só ao nível das ferramentas utilizadas, a destacar a TWS e linguagem *Python* para a interação com a API, como também no conhecimento sobre o mundo dos mercados financeiros e como utilizar as ferramentas mencionadas para realizar um sistema tal como o que foi desenvolvido para este projeto. Desta forma, e concluindo, o autor aprofundou conhecimentos na área dos mercados financeiros e utilização de *softwares* de apoio à interação com estes mercados, conhecimentos estes que se poderão revelar bastante úteis em projetos a desenvolver no futuro.

7.2 Trabalho Futuro

No final do trabalho realizado, é possível identificar algumas tarefas que podem ser acrescentadas e/ou melhoradas, entre as quais as seguintes:

- Desenvolver uma *dashboard* com dados estatísticos já recolhidos pelo sistema (balanço diário, número de ordens enviadas no dia, etc...);
- Realizar testes intensivos ao algoritmo de negociação desenvolvido;
- Desenvolver e melhorar a estratégia de negociação automatizada implementada;
- Permitir ao utilizador parametrizar o algoritmo de negociação implementado (alterar as médias, o intervalo de ação do algoritmo, etc...);
- Implementar novos algoritmos de negociação e disponibilizar uma lista de opções para o utilizador.

Bibliografia

- [1] Penny Stock, "Análise Interactive Brokers", <https://www.pennystocks.pt/analise-interactive-brokers/>. Último acesso no dia 5 de Fevereiro de 2021. ix, 9, 15
- [2] Henrique Garcia, "Sistemas negociação automatizados para investir na bolsa", <https://www.rankia.pt/bolsa/sistemas-negociacao-automatizados-para-investir-na-bolsa/>. Último acesso no dia 5 de Fevereiro de 2021. 5
- [3] Interactive Brokers, <https://www.interactivebrokers.com/en/home.php>. Último acesso no dia 5 de Fevereiro de 2021. 3
- [4] Dicionário Financeiro, "Ativos financeiros: o que são e como são classificados?", <https://www.dicionariofinanceiro.com/ativos-financeiros/>. Último acesso no dia 5 de Fevereiro de 2021.
- [5] Algorithmic Trading Market, <https://www.marketsandmarkets.com/Market-Reports/algorithmic-trading-market-179361860.html>. Último acesso no dia 5 de Fevereiro de 2021. ix, 5
- [6] CNBC, "80% of the stock market is now on autopilot", <https://www.cnbc.com/2019/06/28/80percent-of-the-stock-market-is-now-on-autopilot.html>. Último acesso no dia 5 de Fevereiro de 2021. 5
- [7] SmarttBot, "Como Funciona, Plataforma completa com interface amigável", <https://smarttbot.com/como-funciona/>. Último acesso no dia 5 de Fevereiro de 2021. 7
- [8] "History of Automated Trading", pág. 22, <https://www.nirvanasystems.com/wp-content/uploads/2018/07/HISTORY-OF-AUTOMATED-TRADING.pdf>. Último acesso no dia 5 de Fevereiro de 2021. 6
- [9] ADAM HAYES, "What Is Simple Moving Average (SMA)?", <https://www.investopedia.com/terms/s/sma.asp>. Último acesso no dia 5 de Fevereiro de 2021. 10