

# PROTÓTIPO PARA PREVISÃO DE AÇÕES UTILIZANDO REDES NEURAIS ARTIFICIAIS

Rafael Bertoldi Rossi, Andreza Sartori – Orientadora

Curso de Bacharel em Ciência da Computação  
Departamento de Sistemas e Computação  
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

rbrossi@furb.br, asartori@furb.br

**Resumo:** Operar no mercado financeiro é uma atividade complexa, e não raro são os casos de perdas financeiras significativas devido a erros operacionais. Por este motivo, este trabalho visa desenvolver um protótipo utilizando Redes Neurais Artificiais Long Short-Term Memory (LSTM) a fim de realizar automaticamente a previsão dos próximos movimentos de ações cotadas na B3 S.A. São abordados tópicos sobre Redes Neurais Recorrentes, LSTM, análise técnica e trading. Os resultados obtidos foram, de modo geral, satisfatórios, obtendo a taxa de 70% de acerto para o movimento ocorrido na semana, e apresentando em diversos casos uma previsão muito próxima da realidade com comprovação estatística utilizando Valor-P, corroborando que uma abordagem utilizando Aprendizado de Máquina no setor financeiro é relevante.

**Palavras-chave:** Redes Neurais Artificiais. Previsão. Ações. Trading. LSTM.

## 1 INTRODUÇÃO

Entre meados da idade média e do século XVII, as negociações – a viva voz – de ativos financeiros resumiam-se à compra e venda de moedas, letras de câmbio e metais preciosos. A falta de crédito e a língua eram verdadeiros obstáculos para os mercadores. Tais adversidades marcaram a origem da especulação e do surgimento Bolsa de Valores. Em 1309, os encontros diários, que ocorriam no prédio da família belga Van der Buerse, na cidade de Bruges, na Bélgica, foram formalizados e passaram a chamar-se de “Bruges Bourse”. Foi no final do século XIII que a palavra bolsa ganhou conotação comercial e financeira, advento da família, que possuía um brasão de armas com três bolsas de pele, denotando honra e méritos, devido à excepcional atuação na esfera mercantil (PINHEIRO, 2014).

Em 1600, no meio do processo de colonização da Ásia, já havia seis empresas holandesas que tinham a capacidade de navegar até a Índia, em busca de temperos, afinal as especiarias faziam parte do comércio mais lucrativo do mundo, àquela época. A maior dificuldade era arranjar fundos para financiar as viagens e banqueiros dispostos a correr o enorme risco da empreitada ser um desastre – em 1598, vinte e duas embarcações saíram da Holanda rumo ao Oriente e só doze voltaram. A ideia dos holandeses foi dispersar o risco dividindo o custo da operação entre vários investidores. O governo uniu as seis Companhias das Índias e formou uma grande estatal, nominada como Companhia Unida Holandesa das Índias Orientais. A empresa foi dividida em várias partes e podia ser comprada por qualquer pessoa. Para realizar essa negociação, foi constituída uma bolsa em Amsterdã, e 1143 pessoas realizaram o investimento. O nome de cada parte da empresa foi chamado de “ação” (VERSIGNASSI, 2015, p. 74 - 77).

Foi no final dos anos 80, com a execução de ordens de forma totalmente eletrônica, através dos *home brokers*, e em 1998, com a autorização de negociações via High Frequency Trading (HFT) por parte da U.S. Securities and Exchange Commission (SEC) o estopim para que nos anos seguintes ocorresse um verdadeiro *boom* no mercado financeiro. Em 2012 surgiam os primeiros serviços utilizando inteligência artificial para identificar os assuntos mais comentados em redes sociais, usando processamento de linguagem natural em tempo real para analisar e classificar reações dos clientes de determinado produto, e prover *insights* ao investidor para estabelecer um viés correto ao decidir no que apostar (GUPTA, 2015).

Segundo Elder (2017), os sentimentos do *trader* exercem impactos imediatos na conta de investimento, e o sucesso ou o fracasso do operador depende não só das técnicas empregadas, mas também do controle de suas emoções. Estudos feitos por Barber et al. (2010) analisaram a performance de investidores na Bolsa de Taiwan entre 1992 e 2006, e concluiu-se que 40% de todos os *day-traders* desistem no seu primeiro mês de operação, 80% desistem nos primeiros 2 anos, sendo que a grande maioria acumula perdas. Kahneman e Tversky (1997) propuseram a Teoria da Perspectiva, estudo relacionado à psicologia cognitiva, que sugere que uma perda tem um impacto emocional muito maior do que um ganho financeiramente equivalente. Isso faz com que o investidor seja induzido a segurar o prejuízo como forma de evitar uma perda, e realizar o lucro precipitadamente, corroborando com a concepção de que o *trading* – conceito econômico envolvendo a compra e venda de ativos (FRANKENFIELD, 2019) – é uma atividade árdua e desvantajosa.

Em contrapartida, trabalhos recentes como os apresentados por Bao, Yue e Rao (2017) e Fabbri e Moro (2018) obtiveram excelentes resultados utilizando técnicas de Redes Neurais Artificiais no desenvolvimento de sistemas para previsão de ativos do mercado financeiro. Este trabalho propõe uma ferramenta que utiliza um modelo de Long Short-Term Memory (LSTM) para prever os próximos movimentos de ações brasileiras cotadas na B3 S.A., através de séries

temporais. As análises são executadas por um modelo preditivo, disponibilizando ao investidor uma visão das possíveis direções que as ações tendem a seguir, tornando este protótipo uma importante ferramenta para os *traders*, haja vista tamanha complexidade do mercado. Os objetivos específicos são: (i) desenvolver um protótipo para prever os próximos movimentos das ações utilizando Redes Neurais Artificiais; (ii) realizar análise estatística a fim de avaliar a acurácia para validar o desempenho do modelo preditivo; (iii) disponibilizar as previsões para o usuário por meio de uma página web com cotações e análises atualizadas em tempo real; (iv) identificar os algoritmos e técnicas de Redes Neurais Artificiais mais adequados para predição das operações.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais assuntos relacionados à análise técnica e estilos de investimentos, bem como as Redes Neurais Artificiais (RNA). Ao final, são descritos os trabalhos correlatos a este protótipo.

### 2.1 TRADING

Segundo Folger (2019), *trading* é um estilo de investimento que busca obter retornos acima do tradicional *buy-and-hold* – estratégia de investimento em que o investidor fica posicionado por um longo período, apesar das flutuações do mercado (CHEN; BEERS, 2019). Ao invés de manter posições por vários meses, os *traders* visam o lucro num curto prazo, independente se os mercados estão em alta ou em baixa (FOLGER, 2019).

Diferentemente da análise fundamentalista, em que o investidor avalia o balanço patrimonial da empresa, fatores macroeconômicos, e tudo o que possa influenciar no valor futuro da empresa, o *trading* está muito mais ligado a avaliar o comportamento dos preços através de gráficos. Desta forma, procura-se por padrões conhecidos na análise técnica como médias móveis, regiões de suporte e resistência (regiões onde encontram-se uma grande quantidade de compradores e vendedores defendendo o preço), e indicadores, como pode ser visto na Figura 1 (MAJASKI, 2019).

Figura 1 – Gráfico de *candlesticks* do Ibovespa entre julho e novembro de 2019



Fonte: elaborado pelo autor.

Estes gráficos são representados geralmente por *candlesticks*, um modo de visualização criado no século XVIII, no Japão, para representar os valores de abertura, máxima, mínima e fechamento, respectivamente. Os *candles* retratam a variação do preço dentro de um determinado período, a depender do *timeframe* (intervalo de tempo) utilizado. Estes *candlesticks* são interpretados na análise técnica como uma representação da emoção dos *traders*, de acordo com o padrão formado, podendo indicar que os vendedores estão mais fortes do que os compradores, por exemplo. *Candles* que estão com a coloração verde apresentam o valor de fechamento maior do que o valor da abertura, significando que houve uma valorização no preço do ativo, já os *candles* com a coloração vermelha, que houve uma queda na cotação (GRANT, 2019).

Segundo Hayes (2019), a análise técnica é utilizada para identificar oportunidades de *trading* em tendências e padrões de preços observados nos gráficos, introduzida pela Teoria de Charles Dow no final do século XIX. Analistas técnicos acreditam que os padrões encontrados nos preços são indicadores de movimentos futuros. Essencialmente, a análise técnica se baseia em três pilares:

- o mercado desconta tudo: todos os fatores, desde os fundamentos da empresa até os fatores psicológicos das massas estão precificados no valor do ativo, eliminando a necessidade de considerar os diversos fatores isoladamente;
- os preços se movem em tendências de longo, médio e curto prazo;
- a história tende a se repetir: isso é atribuído à psicologia do mercado, que tende a ser previsível, visto que as emoções, como o medo ou a excitação, estão presentes no *trader* e produzem padrões gráficos com o

passar do tempo.

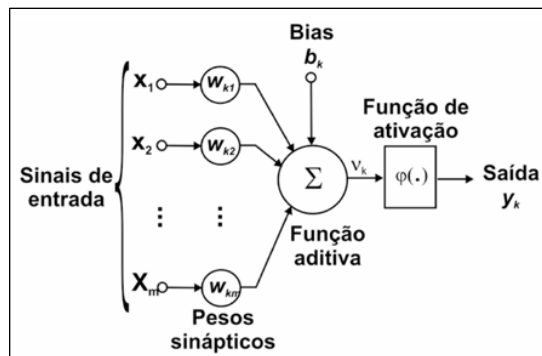
Há basicamente quatro estilos de *trade*, que estão ligados diretamente ao tempo de duração das posições. *Position Trading* visa combinar métodos para identificar a tendência do ativo, e ficar posicionado no investimento por vários dias ou semanas. Para este estilo, utiliza-se gráficos no *timeframe* diário, semanal e mensal. Os *Swing Traders* buscam identificar uma nova tendência e a operá-la a seu favor, com regras baseadas na análise técnica. As operações duram mais de um dia, mas dificilmente ultrapassam uma semana. *Day Trading* é o estilo mais conhecido e como o próprio nome sugere, as operações têm apenas um dia de duração. Por último, o *Scalping* objetiva explorar pequenas variações no preço, com *trades* que duram minutos ou segundos. *Scalpers* optam por mercados com maior volatilidade e liquidez com o intuito de aumentar o número de oportunidades (ZUCCHI, 2018).

## 2.2 REDES NEURAIS ARTIFICIAIS

Modelos matemáticos inspirados na hipótese de que a atividade mental consiste na condução de impulsos nervosos já surgiam no início da década de 40. O trabalho de McCulloch e Pitts (1943) objetivou a criação de um neurônio artificial, simulando a atividade do cérebro humano, através da implementação de um limiar que é ativado de acordo com a combinação linear dos valores submetidos na entrada. Uma Rede Neural Artificial (RNA) é o conjunto de neurônios, e o seu funcionamento depende da topologia, dos pesos dados a cada neurônio e da função de ativação (RUSSELL; NORVIG, 2013).

A Figura 2 apresenta o funcionamento de uma Rede Neural Perceptron, um classificador binário simples, com apenas uma camada. Através dos sinais de entrada  $x_1$ ,  $x_2$  e  $x_n$ , valores são inseridos e pesos sinápticos são atribuídos. Esses valores são repassados à função aditiva, que junto com um viés (Bias  $b_k$ ), tentam obter os melhores valores a partir do processo de treinamento, que consiste no aprendizado dos melhores valores. Por fim, a função de ativação decide com base nos valores gerados, se a saída deve ser 0 ou 1.

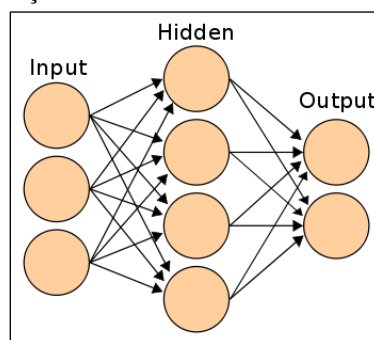
Figura 2 – Funcionamento de um neurônio computacional



Fonte: Margins-filho, Mol e Rocha (2005).

As RNAs são essencialmente compostas por neurônios conectados por *links* direcionais, que servem para propagar a ativação. Pesos numéricos são associados aos *links* para determinar a força e o sinal da conexão, e finalmente há uma função ativação para a aplicação de um limiar. Neurônios artificiais conectados na forma de um grafo com conexões direcionais formam uma RNA (RUSSELL; NORVIG, 2013). A Figura 3 representa uma RNA, contendo três neurônios na entrada (*input*), quatro na camada escondida (*hidden*), e dois na camada de saída (*output*).

Figura 3 – Representação da estrutura básica de uma Rede Neural Artificial



Fonte: Srivastava (2014).

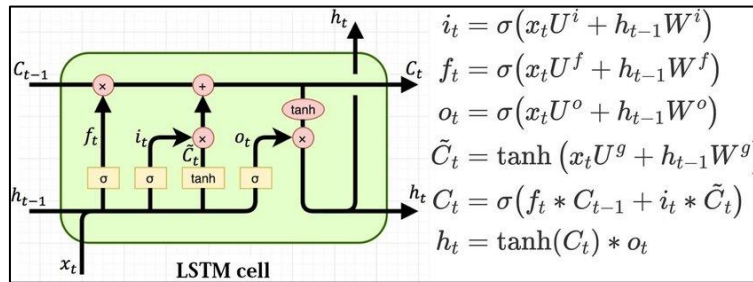
### 2.2.1 Redes Neurais Recorrentes: Long Short-Term Memory (LSTM)

Os avanços trazidos por Hopfield (1982) propuseram um tipo de rede neural que utiliza conexões bidirecionais com pesos simétricos, funcionando como uma memória associativa, considerada a primeira Rede Neural Recorrente

(RNR). Russell e Norvig (2013) definem a RNR como uma rede que alimenta suas saídas de volta às suas próprias entradas. Além de depender do estado inicial, a RNR pode depender de entradas anteriores, o que a torna o modelo complexo, porém mais similar ao funcionamento do cérebro humano, por suportar memórias de curto prazo devido à sua arquitetura.

Apresentada por Hochreiter e Schmidhuber (1997), a Long Short-Term Memory (LSTM) é um tipo de Rede Neural Recorrente que tornou-se muito popular nos últimos anos devido à performance superior e à capacidade de reduzir o problema de *vanish gradient* – dificuldade para encontrar os melhores parâmetros durante o período de treinamento de modelos baseados em gradiente (ARBEL, 2018). A LSTM foi originalmente desenhada para obter informações de longo prazo e resolver um problema antigo das RNRs. É um modelo complexo composto por três portões: *input gate* ( $i_t$ ), *output gate* ( $o_t$ ) e *forget gate* ( $f_t$ ), e também uma célula de memória adicional ( $C_t$ ), conforme apresentado na Figura 4.

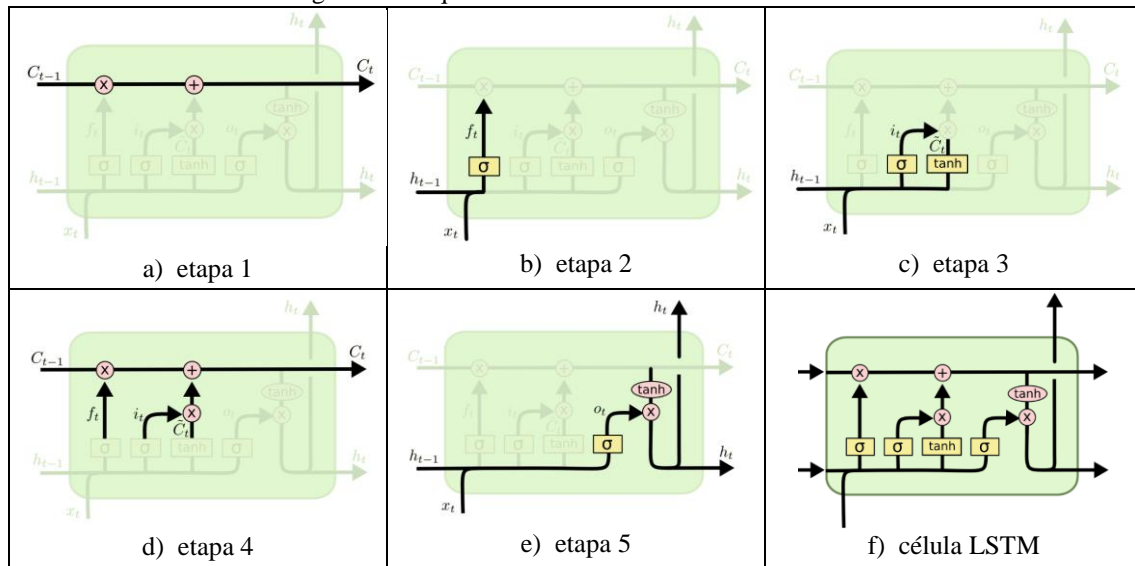
Figura 4 – Equações e célula LSTM



Fonte: Varsamopoulos, Bertels e Almudever (2018).

A ideia principal concentra-se no estado da célula de memória, que está representado na Figura 5, etapa 1. Este estado flui através do tempo  $t$ , com dados sendo recebidos da célula anterior, e também repassando para a próxima célula. A primeira parte do processamento ocorre na função  $h_{t-1}$  que é a saída do tempo anterior e no  $x_t$ , que é o valor atual, representado pela etapa 2. Estas duas informações serão submetidas a uma função sigmoide para decidir o que será apagado, na etapa 3. O próximo passo é decidir o que será armazenado, através da função de tangente hiperbólica, para então adicionar ou apagar os valores (*input gate*  $i_t$  e *forget gate*  $f_t$ ) e adicioná-los na célula, representado na etapa 4.

Figura 5 – Arquitetura e funcionamento da LSTM



Fonte: Olah (2015).

Por fim, o *output gate*  $O_t$  recebe os dados através de outra função sigmoide, decide o que será importante pela função de tangente hiperbólica e retorna o valor nas saídas  $h_t$  (etapa 5). A propagação dos valores da célula atual  $C_t$  também estará acessível pelos próximos neurônios, para dar continuidade ao ciclo (OLAH, 2015).

### 2.3 TRABALHOS CORRELATOS

Esta seção apresenta os trabalhos que possuem características semelhantes aos principais objetivos do protótipo proposto. O Quadro 1 descreve o *framework* de Bao, Yue e Rao (2017) que utilizou técnicas de Aprendizado Profundo em conjunto com Stacked Autoencoders (SAE) e Wavelet Transforms (WT), a fim de diminuir os ruídos dos dados de entrada do framework. No Quadro 2, o trabalho de M'ng e Mehralizadeh (2016) buscou facilitar a previsão de índices futuros através da eliminação de ruídos, utilizando RNA. Por fim, no Quadro 3, o trabalho de Chen G., Chen Y. e Fushimi

(2017) apresenta uma aplicação predição da ação Intel Corporation utilizando Redes Neurais Long Short-Term Memory (LSTM).

Quadro 1 – Trabalho Correlato 1

Referência	Bao, Yue e Rao (2017).
Objetivos	Uso de Wavelet Transforms (WT), Stacked Autoencoders (SAE) e Long Short-Term Memory (LSTM) para realizar operações de compra e venda em seis índices de ações.
Principais funcionalidades	Predição realizada através de Redes Neurais LSTM, utilizando Wavelet Transform (WT) para diminuição do ruído. O modelo baseia-se em Stacked Autoencoders (SAE), ferramenta utilizada para extração de características de maneira não-supervisionada. No total, são cinco camadas densas treinadas individualmente. Este processo é feito para reduzir o erro entre os dados de entrada e saída.
Ferramentas de desenvolvimento	Long Short-Term Memory, Wavelet Transform, Stacked Autoencoders.
Resultados e conclusões	Os índices de ações CSI 300, Nifty 50, Hang Seng, Nikkei 225, S&P500 e DJIA foram selecionados para testar a capacidade de previsão do modelo proposto. Para os dados de entrada da primeira camada, foram utilizados os dados históricos dos preços de abertura, máxima, mínima e fechamento diários. Os indicadores de análise técnica serviram como <i>inputs</i> para a segunda camada, e por último, variáveis macroeconômicas. O <i>framework</i> WSAEs-LSTM, que é combinação de Wavelet Transform e Stacked Autoencoder, foi treinado utilizando 5 camadas com dados de 2008 a 2016. O trabalho obteve retornos anuais sempre acima de 45%, mostrando-se mais eficiente que o <i>buy-and-hold</i> e algoritmos genéricos.

Fonte: elaborado pelo autor.

O trabalho de Bao, Yue e Rao (2017) relaciona-se com o protótipo apresentado neste trabalho porque utiliza Redes Neurais Recorrentes LSTM para realizar operações de compra e venda nos índices de ações CSI 300, Nifty 50, Hang Seng, Nikkei 225, S&P500 e DJIA.

Quadro 2 – Trabalho Correlato 2

Referência	M'ng e Mehralizadeh (2016).
Objetivos	Facilitar a previsão e identificar o melhor momento de entrada nos índices futuros de países do leste-asiático através de um processo de eliminação de ruídos, combinando Wavelet Transforms (WT), Principal Component Analysis (PCA) e Redes Neurais Artificiais (RNA).
Principais funcionalidades	Predição de índices futuros Hang Seng, Nikkei 225, MSCI, Kospi 20 e Taiex, utilizando os valores diários de abertura, máxima, mínima e fechamento, bem como indicadores conhecidos na análise técnica: Índice de Força Relativa, Média Móvel Convergente e Divergente, Estocástico Rápido, Estocástico Lento, Estocástico e <i>Ultimate Oscillator</i> . Estes indicadores buscam identificar o melhor momento para entrar em cada <i>trade</i> , tornando-se também um objetivo adicional do modelo.
Ferramentas de desenvolvimento	Redes Neurais Artificiais, Wavelet Transform, Principal Component Analysis
Resultados e conclusões	O modelo, apresentado como WPCA-NN (Wavelet Principal Component Analysis – Neural Network), foi treinado entre 2002 e 2015 utilizando valores diários dos índices e indicadores de análise técnica: Índice de Força Relativa, Média Móvel Convergente e Divergente, Estocástico Rápido, Estocástico Lento, Estocástico e <i>Ultimate Oscillator</i> . O WT foi aplicado a fim de reduzir o ruído e o PCA para realizar uma transformação e obter variáveis possivelmente correlacionadas. Em comparação com algoritmos genéricos e <i>buy-and-hold</i> , o modelo mostrou-se superior, desempenhando um lucro médio de 49,9% ao ano.

Fonte: elaborado pelo autor.

O modelo proposto por M'ng e Mehralizadeh (2016) é correlacionado a este trabalho porque visa realizar a predição de ativos do mercado financeiro com base no valor diário dos negócios realizados, utilizando um modelo Redes Neurais Artificiais.

Quadro 3 – Trabalho Correlato 3

Referência	Chen G., Chen Y. e Fushimi (2017)
Objetivos	Previsão do movimento das ações da Intel Corporation (INTC) aplicando Redes Neurais Recorrentes. Para isso, foi implementada uma rede LSTM para comparar com Locally Weighted Regression (LWR), um modelo não-parametrizado usado para previsão de dados temporais.

Principais funcionalidades	Predição utilizando Redes Neurais Recorrentes com valores de abertura, máxima, mínima e fechamento da ação da Intel, indicadores de análise técnica e índices de ações internacionais para análise macroeconômica.
Ferramentas de desenvolvimento	Long Short-Term Memory
Resultados e conclusões	Os dados de entrada, obtidos entre janeiro de 2010 e junho de 2017, foram divididos em três períodos. Para cada período os dados dos dois primeiros anos foram usados para treinamento, os próximos três meses para validação e os três meses seguintes para teste, em repetição, até finalizar a série temporal. Os <i>inputs</i> utilizados no sistema preditivo incluem os preços diários de abertura, máxima, mínima e fechamento, bem como uma série de indicadores conhecidos na análise técnica. Para o cenário macroeconômico, os índices de ações S&P 500, CBOE Volatility Index e PHLX Semiconductor Sector Index também foram inseridos. Os resultados alcançados foram superiores, se comparados ao algoritmo genérico e ao <i>buy-and-hold</i> em todos os períodos. A LSTM teve retorno médio de 9,70% ao ano.

Fonte: elaborado pelo autor.

A aplicação desenvolvida por Chen G., Chen Y. e Fushimi (2017) também tem relação com o presente protótipo porque utiliza Redes Neurais Recorrentes, mais especificamente um modelo Long Short-Term Memory para realizar a previsão do movimento da ação Intel Corporation (INTC).

Como o presente protótipo, os três trabalhos correlatos têm o objetivo de realizar previsões no mercado financeiro, e assim como nas aplicações de Chen G., Chen Y. e Fushimi (2017) e Bao, Yue e Rao (2017) são utilizados modelos baseados em LSTM. Neste trabalho, a utilização da LSTM se difere porque a proposta é realizar previsões para os próximos dias, de ações cotadas na B3 S.A. Além disso, os resultados são apresentados através de uma página WEB, com cotações e análises atualizadas em tempo real.

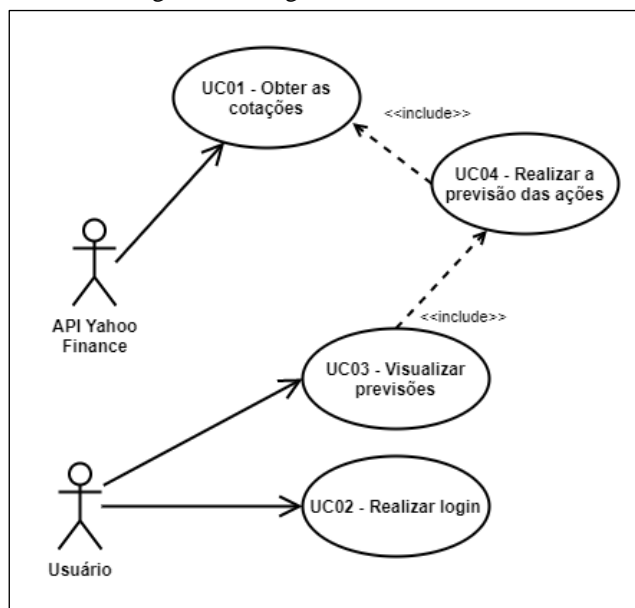
### 3 DESCRIÇÃO DO PROTÓTIPO

Nesta seção são apresentadas as especificações do protótipo que têm como objetivo principal a implementação de uma ferramenta que possibilite realizar a previsão das ações, utilizando técnicas de Redes Neurais Artificiais. O capítulo está dividido em especificação, a arquitetura, *back-end* e *front-end* do protótipo.

#### 3.1 ESPECIFICAÇÃO

Para que o usuário consiga visualizar as previsões, faz-se necessário o acesso a uma plataforma WEB que realize as previsões através de um modelo de Rede Neural Artificial, atualize as cotações atuais de cada ativo, e apresente os resultados. Sendo assim, a aplicação deve conter as funcionalidades apresentadas no Diagrama de Caso de Uso da Figura 6. O ator *Usuário* pode realizar o *login* (UC02) e visualizar as previsões (UC03). Para a realização das previsões (UC04), é necessário obter as cotações atualizadas (UC01), fornecidas pelo ator *API Yahoo Finance*.

Figura 6 – Diagrama de Caso de Uso



Fonte: elaborado pelo autor.

O Quadro 4 apresenta os Requisitos Funcionais (RF) e os Casos de Uso (UC) relacionados. O Quadro 5 relaciona os Requisitos Não Funcionais (RNF).

Quadro 4 – Requisitos Funcionais (RF) e matriz de rastreabilidade

Requisito funcional	Caso de uso
RF01: Obter as cotações atuais e históricas das ações	UC01
RF02: Manter o cadastro de usuários do sistema	UC02
RF03: Possuir um módulo para acompanhamento das previsões	UC03
RF04: Realizar a previsão das ações para os próximos dias	UC04

Fonte: elaborado pelo autor.

Quadro 5 – Requisitos Não Funcionais (RNF)

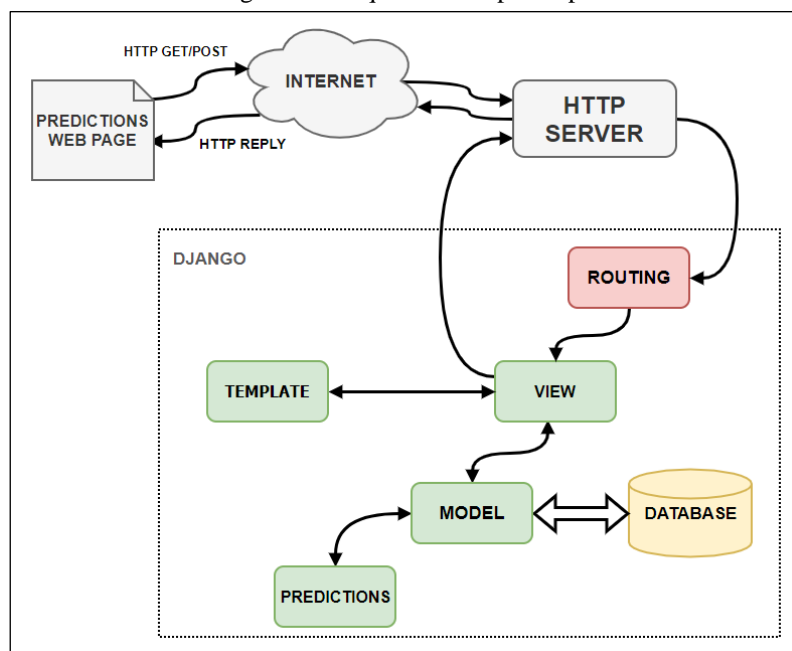
Requisito não funcional
RNF01: Utilizar um modelo preditivo baseado em técnicas de Redes Neurais Artificiais
RNF02: Ser implementado na linguagem Python no ambiente de desenvolvimento PyCharm
RNF03: Utilizar o <i>framework</i> Django para desenvolver a aplicação web

Fonte: elaborado pelo autor.

### 3.2 PROTÓTIPO

A Figura 7 ilustra a arquitetura do protótipo como um todo, podendo ser definida em quatro partes principais: 1) a requisição do tipo GET/POST, passando pela Internet e sendo recebida no HTTP Server, em cinza; 2) o roteamento das páginas, sendo gerenciado pelo arquivo `urls.py`, em vermelho; 3) O modelo estrutural do Django, Model-View-Template (MVT); 4) a conexão com o banco de dados, em amarelo. A estrutura contempla o *front-end*, que compõe principalmente a parte de *template* e *predictions web page*, e também o *back-end*, através das *view*, *model*, *predictions* e *database*.

Figura 7 – Arquitetura do protótipo



Fonte: elaborado pelo autor.

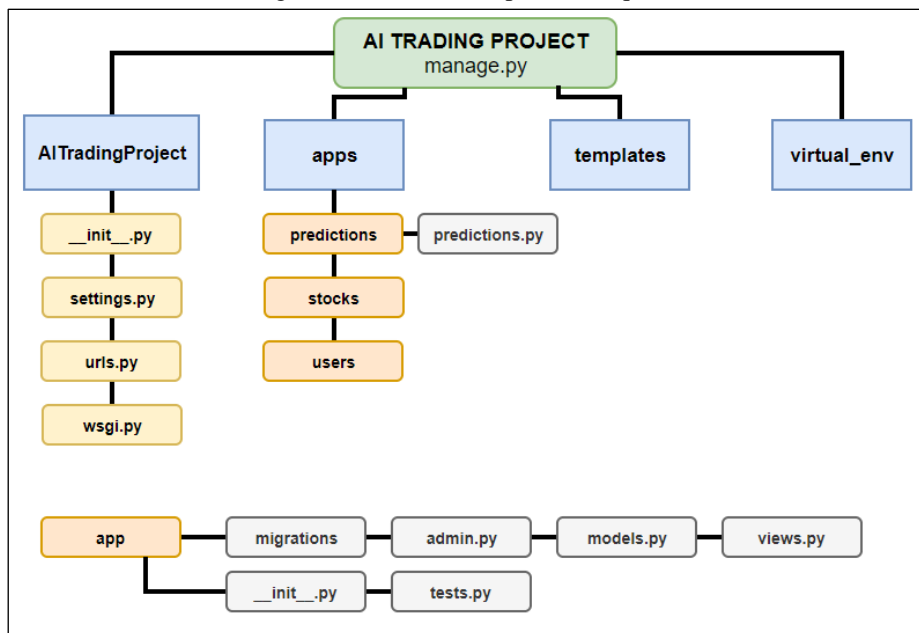
A estrutura de pastas e arquivos do projeto é apresentada na Figura 8. Todo o gerenciamento do projeto Django é realizado pelo arquivo `manage.py`, e sendo executado através de um *virtual environment*, ferramenta para isolamento de ambiente de desenvolvimento. Todos os parâmetros para configuração, como por exemplo roteamento das páginas, endereços IPs permitidos e acesso ao banco de dados, estão descritos sob a pasta `AITradingProject`. Os arquivos relacionados ao *template* da aplicação estão na pasta `templates`.

Para cada funcionalidade da aplicação, foi inicializado um Django App. Cada *app* contém a mesma estrutura de arquivos, de tal forma que funcione independentemente um do outro. Esta estrutura está representada com os itens em cinza, na Figura 8. A pasta `migrations` detém todas as alterações realizadas no banco de dados a nível estrutural, por exemplo, quando é realizado algum incremento ou exclusão de campos em tabelas. Já o arquivo `models.py` é o responsável pela comunicação com o banco de dados. O arquivo `tests.py` concede ao usuário alternativas para realizar

testes unitários no código. O arquivo `views.py` realiza a interface do *back-end* com os *templates*. O `admin.py`, como o próprio nome sugere, faz referência às configurações de administração do *app*. Por fim, o `__init__.py` é responsável pela construção e inicialização dos objetos. A execução da LSTM ocorre no arquivo `predictions.py` e será detalhada na seção 3.3.

O *app* `predictions` contém toda a estrutura para realizar as previsões, desde o carregamento dos valores, atualização das cotações das ações, até a execução da Rede Neural Recorrente. O *app* `stocks` detém as funcionalidades referentes às ações, como por exemplo a lista de ações disponíveis para realizar as previsões e a relação de ações na carteira de cada usuário. A conexão é via banco de dados, através de uma *foreign key*. Por fim, o *app* `users` contém as funcionalidades relacionadas aos usuários da aplicação, como rotinas de `login` e cadastro.

Figura 8 – Estrutura de pastas e arquivos



Fonte: elaborado pelo autor.

### 3.3 BACK-END

Após o usuário realizar o *login* no sistema, a aplicação realiza uma busca através da API do Yahoo Finance para realizar o download das cotações atuais e inseri-las no modelo, conforme Quadro 6. Essa requisição, detalhada na linha 8, utiliza um `DataReader` fornecido pela biblioteca `Pandas`. A saída é um `dataframe`, estrutura de dados fornecida pela biblioteca, que simula uma tabela, contendo linhas e colunas.

Quadro 6 – Rotina para obter cotações atuais

```

1 # obter dia atual
2 day = dt.datetime.today().day
3 month = dt.datetime.today().month
4 year = dt.datetime.today().year
5 # download dos dados da ação de 2015 até dia atual
6 start = dt.datetime(2015,1,1)
7 end = dt.datetime(year,month,day)
8 df = web.DataReader(ativo+'.SA', 'yahoo', start, end)
  
```

Fonte: elaborado pelo autor.

Após o sistema ser iniciado no servidor, é feita uma consulta a fim de encontrar arquivos de treinamento da RNA. Caso estes não forem encontrados ou estiverem desatualizados, o sistema realiza um novo treinamento, que pode ser prolongado devido ao custo computacional, dependendo do poder de processamento do servidor, da quantidade de ações cadastradas, etc. Uma vez que os arquivos de treinamento estejam disponíveis e atualizados, a previsão pode ser realizada, e ocorre através da função `predict(stock)`, que está detalhada no Quadro 7.



Quadro 7 – Função Predict

```

1 def predict(stock):
2     file = open('predictions/weights/'+stock+'.json', 'r')
3     structure = file.read()
4     file.close
5     regressor = model_from_json(structure)
6     regressor.load_weights('predictions/weights/'+stock+'.h5')
7     # download da base atualizada
8     df = web.DataReader(stock+'.SA', 'yahoo', start, end)
9     df = df.reset_index()
10    # selecionar valores de fechamento da ação
11    values = df.iloc[:,4:5].values
12    # realizar normalização dos valores
13    normalizer = MinMaxScaler(feature_range=(0,1))
14    normalized_price = normalizer.fit_transform(values)
15    price = np.squeeze(np.asarray(normalized_price))
16    # definir o número de steps de entrada e saída
17    n_steps_in, n_steps_out = 50, 5
18    # split dos valores
19    X, y = split_sequence(price, n_steps_in, n_steps_out)
20    # reshape de [samples, timesteps] para [samples, timesteps, features]
21    n_features = 1
22    X = X.reshape((X.shape[0], X.shape[1], n_features))
23    x_input= (normalized_price[normalized_price.shape[0]-n_steps_in:])
24    x_input = x_input.reshape(1, n_steps_in, n_features)
25    # realizar a previsão
26    predictions = regressor.predict(x_input, verbose=0)
27    # inverter a normalização
28    predictions = normalizer.inverse_transform(predictions)

```

Fonte: elaborado pelo autor.

A função `predict` necessita de um parâmetro `stock`, que nada mais é do que o *ticker* da ação a ser prevista. O primeiro passo é obter os valores de fechamento e realizar a normalização – processo realizado para aumento do desempenho – através do normalizador `MinMaxScaler` (linha 13) e da função `fit_transform`, (linha 14), ambos disponibilizados pela biblioteca `Sklearn`. Uma vez que os dados estão normalizados, é necessário determinar a quantidade de valores que serão utilizados para realizar a previsão, e a quantidade de valores que serão previstos. Este processo ocorre na linha 17, na atribuição dos valores para as variáveis `n_steps_in` e `n_steps_out`. Neste protótipo, utilizou-se os últimos 50 preços de fechamento como *input* da LSTM, por ser um período em que a ação seguramente passou por cenários de alta e baixa.

O diferencial na estrutura da LSTM deste protótipo, possibilitando realizar a previsão dos próximos 5 valores com base nos últimos 50, é a utilização da função `split_sequence` na linha 17. Esta função foi desenvolvida por Brownlee (2018) que visa identificar e visualizar os próximos movimentos que a ação irá fazer. O código, descrito no Quadro 8, ocorre da seguinte forma: os parâmetros de entrada são 3, uma sequência de valores, que neste caso é o preço, já normalizado como visto no Quadro 7, o número de *steps* de entrada e o número de *steps* de saída, que são 50 e 5, respectivamente. A função retorna dois *arrays* com os dados separados, já prontos para realizar a previsão.

Após a separação dos dados, é realizada a previsão (linha 26, Quadro 7). A função `predict` necessita de um `x_input`, que são os dados que serão informados para a LSTM, ou seja, os últimos 50 valores de fechamento da ação escolhida. Finalmente, na linha 28 do Quadro 7, ocorre a transformação inversa dos dados normalizados através da função `inverse_transform`. Esta, por sua vez, faz com que os valores voltem ao arranjo original, e seja concedido os próximos 5 valores que a ação deve alcançar de acordo com a previsão. Estes valores são repassados para o *front-end*, e exibidos na página de previsões, detalhada na seção 3.4.

Quadro 8 – Função para realizar a separação da base de dados

```

1 def split_sequence(sequence, n_steps_in, n_steps_out):
2     X, y = list(), list()
3     for i in range(len(sequence)):
4         # encontrar o fim do padrão
5         end_ix = i + n_steps_in
6         out_end_ix = end_ix + n_steps_out
7         # verificar se ultrapassou a sequência do padrão (5,50)
8         if out_end_ix > len(sequence):
9             break
10        # reunir as partes de entrada e saída encontradas na iteração
11        seq_x, seq_y = sequence[i:end_ix], sequence[end_ix:out_end_ix]
12        X.append(seq_x)
13        y.append(seq_y)
14    return array(X), array(y)

```

Fonte: Brownlee (2018).

Para realizar a estruturação da Rede Neural, utilizou-se o Jupyter Notebook, plataforma *open-source*, que fornece um ambiente interativo para desenvolvimento em Python. Optou-se pela utilização de uma Long Short-Term Memory (LSTM), por ser uma estrutura de Rede Neural Recorrente adequada à proposta do protótipo.

O desenvolvimento da rede foi feito pela biblioteca Keras, que é uma interface para facilitar o desenvolvimento e utilização do TensorFlow: biblioteca de código aberto desenvolvida pela Google para Aprendizado de Máquina. A estrutura da LSTM deste protótipo contém 200 neurônios na camada de entrada e 1 neurônio na camada de saída, com função de ativação linear, como apresentado no Quadro 9.

Quadro 9 – Estrutura da LSTM

```

1 # inicialização de um modelo sequencial
2 regressor = Sequential()
3 # adição de 200 neurônios na camada de entrada
4 regressor.add(LSTM(units = 200, input_shape = (n_steps_in, 1)))
5 # adição de 1 neurônio na camada de saída, com ativação linear
6 regressor.add(Dense(units=n_steps_out, activation = 'linear'))
7 # compilação do modelo
8 regressor.compile(optimizer = 'rmsprop', loss = 'mse', metrics = ['mae', 'mape'])
9 # adição de ES, RLR e MCP
10 es = EarlyStopping(monitor='loss', min_delta=1e-10, patience=30)
11 rlr = ReduceLROnPlateau(monitor='loss', factor=0.05, patience=30)
12 mcp = ModelCheckpoint(filepath='pesosX.h5', monitor='loss', save_best_only=True)
13 # treinamento do modelo
14 regressor.fit(X, y, epochs = 500, batch_size = 10, callbacks=[es, rlr, mcp])

```

Fonte: elaborado pelo autor.

Para a composição da LSTM optou-se pela utilização de um modelo sequencial com uma estrutura Dense, ou seja, todos os neurônios se conectam entre si e possuem ativação linear. A escolha do otimizador `rmsprop` foi devido a apresentação de resultados ligeiramente melhores, em comparação ao `adam`. A quantidade de neurônios na camada de entrada, bem como a utilização de apenas uma camada oculta e uma camada de saída, foi devido ao tempo gasto no processo de treinamento e previsão. O modelo precisa ser ágil o suficiente para que o tempo gasto no processamento não impacte nos resultados, tornando o protótipo inviável. A adição de neurônios, camadas e épocas influenciam diretamente neste sentido, e não raros modelos de Aprendizado de Máquina levam dias neste processo.

Quanto à não utilização de *dropout* para redução do *overfitting*, processo que ocorre após cada camada do modelo, foi devido aos péssimos resultados obtidos. Ora o modelo funcionava, ora ocorriam *gaps* nas previsões. Isso ocorre devido à aleatoriedade do *dropout* (técnica que remove valores da base de dados aleatoriamente), uma vez que

sejam retiradas características de importância ímpar, o resultado tende a ser ruim. Em se tratando de otimização, optou-se pela utilização da técnica de `EarlyStopping` (Early Stopping - ER) e `ReduceLRonPlateau` (Reduce Learning Rate on Plateau - RLR), implementadas nas linhas 10 e 11 do Quadro 9. O ER é uma técnica, fornecida pelo Keras, que monitora a taxa de perda (*loss*) do modelo durante o processo de treinamento. Se este valor não diminuir por um determinado período de épocas, neste caso 30, o processo de treinamento é encerrado, já que o modelo não obteve melhoras no resultado durante as últimas 30 rodadas. Já o RLR é uma técnica, também fornecida pelo Keras para reduzir o fator de aprendizagem nos casos em que a taxa de perda não diminuir. Ou seja, ambos são instrumentos para garantir um treinamento mais rápido e eficaz, e diminuir o tempo total de espera do usuário.

Por fim, a quantidade de épocas é variável, com um limite máximo de 500. Isto significa que o modelo, através do ER e RLR, provavelmente encerrará o processo de `fit` (treinamento e ajustes de pesos da RNA) antes desse limite, não sendo necessário aguardar até o final do processo, já que os resultados não estão melhorando.

### 3.4 FRONT-END

Para a exibição das previsões realizadas, elaborou-se uma interface web utilizando o *framework* Django com a biblioteca Bootstrap SB Admin 2 utilizando as linguagens HTML, CSS e Javascript. A comunicação com o *back-end* ocorre via Python com consultas ao banco de dados.

O acesso ao sistema é concedido após inserção de usuário e senha previamente cadastrados. Inicialmente é apresentado uma *dashboard* com as informações referentes à conta de investimento do usuário, tela onde encontra-se informações como saldo em conta, rendimento líquido, etc. A consulta e visualização das previsões é feita através de um menu na parte esquerda, selecionando a opção Previsões. Para cada ação cadastrada, é realizada a previsão, e disponibilizada na tela de previsões para o usuário. A interface é responsiva e se ajusta à tela, característica da biblioteca Bootstrap. Na Figura 9, pode-se visualizar a tela de previsões, contendo os valores sugeridos para os próximos 5 dias, com base nos últimos 50 dias.

Figura 9 – Página para visualização das previsões



Fonte: elaborado pelo autor.

## 4 RESULTADOS

Para mensurar a eficácia do modelo proposto e auferir o resultado em situações reais, designou-se 5 ações para previsão dos valores futuros, utilizando *timeframe* diário, com os últimos 50 valores de fechamento para prever os próximos 5. As ações escolhidas foram Raia Drogasil (RADL3), Vale SA (VALE3), Bradesco (BBDC4), Petrobras (PETR4) e WEG SA (WEGE3) por possuírem uma liquidez alta, e estarem entre as mais negociadas da bolsa de valores.

A avaliação do modelo foi realizada no período de treinamento da LSTM, através dos cálculos de erro Mean Squared Error (MSE) e Mean Absolute Error (MAE). Estes cálculos são modelos estatísticos para estimar valores de erro em modelos de regressão. O MSE (Erro Médio Quadrático) calcula a média das diferenças elevadas ao quadrado, sendo que ele dá mais pesos aos maiores erros, e menos pesos aos menores. Já o MAE (Erro Absoluto Médio) é simplesmente a média dos erros, sem elevação ao quadrado. (VANDEPUT, 2019). Os resultados obtidos são apresentados na Tabela 1.

Tabela 1- Cálculos de erro durante o treinamento (em R\$)

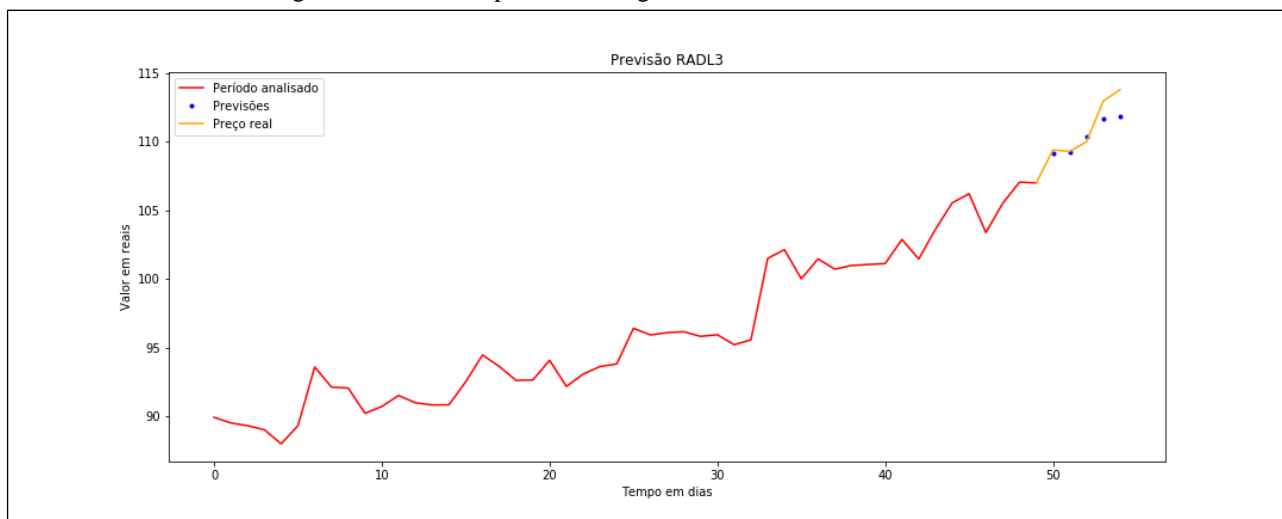
	RADL3	VALE3	BBDC4	PETR4	WEGE3
MSE	0,00046	0,00077	0,00083	0,00085	<b>0,00035</b>
MAE	0,01600	0,01970	0,02150	0,02020	<b>0,01350</b>

Fonte: elaborado pelo autor.

Pode-se perceber que em relação aos cálculos de erro, a WEGE3 e RADL3 tiveram os melhores resultados, sendo que a WEGE3 obteve o menor MSE e MAE. A PETR4 teve o pior resultado no MSE e a BBDC4 resultou no valor mais alto no cálculo de MAE. Com estes valores, é possível observar que as previsões realizadas para WEGE3 foram muito assertivas, com o Erro Absoluto Médio de R\$0,0135 por previsão. O preço real de cada ação pode ser visto nas tabelas 4 e 5.

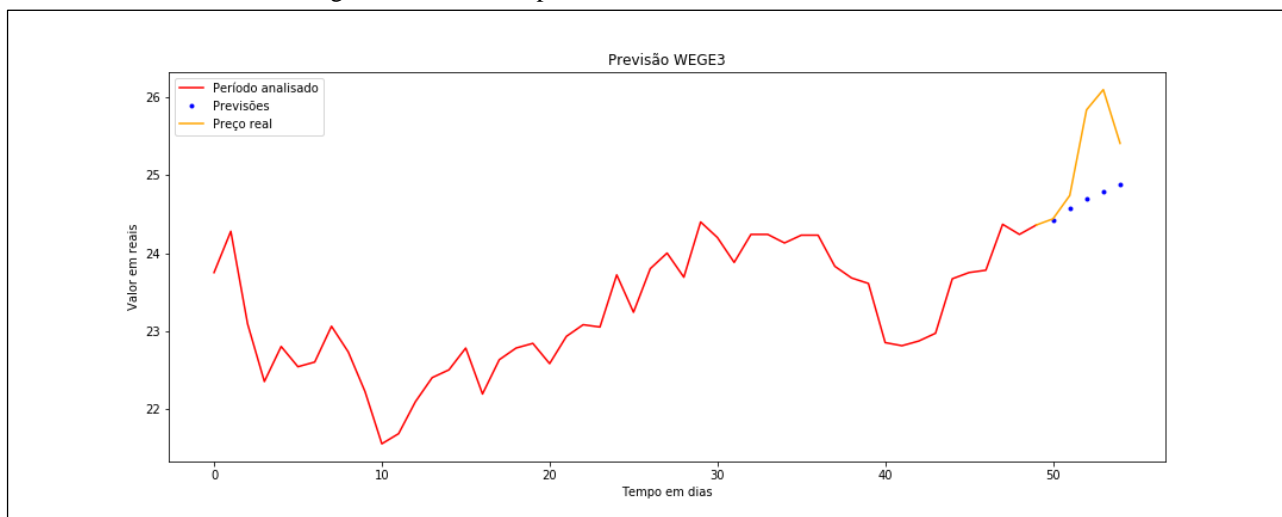
Para validar o modelo durante o período de treinamento, a LSTM foi treinada de 01/01/2015 a 07/10/2019, individualmente para cada ação, com a mesma parametrização, para garantir eficácia perante diferentes características. Os testes foram feitos com o último valor de fechamento de cada semana, para prever os valores de segunda à sexta-feira, iniciando em 16/09/2019 até 12/11/2019. Ao todo, foram 50 previsões realizadas, com a geração de um gráfico que apresenta o período analisado, a previsão e o preço real após a realização da previsão. As previsões entre 28/10/2019 e 01/11/2019 podem ser vistas no Anexo A. Nas Figuras 10, 11 e 12, pode-se visualizar a previsão das ações RADL3, WEGE3 e PETR4, respectivamente.

Figura 10 - Previsão para RaiaDrogasil entre 28/10/2019 e 01/11/2019



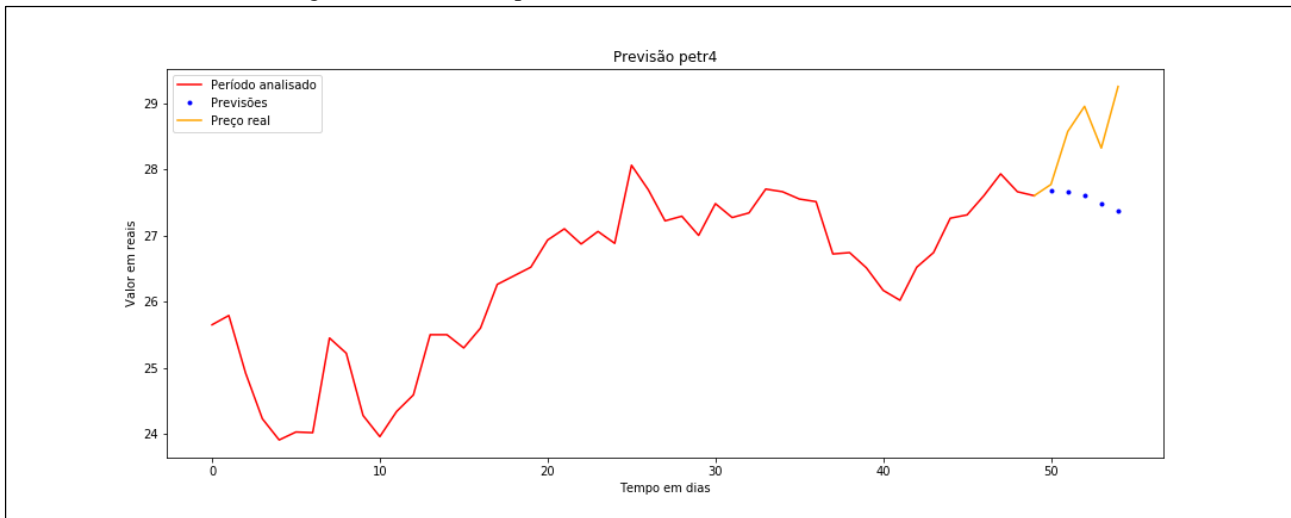
Fonte: elaborado pelo autor.

Figura 11 – Previsão para WEG entre 21/10/2019 e 25/10/2019



Fonte: elaborado pelo autor.

Figura 12 – Previsão para Petrobras entre 21/10/2019 e 25/10/2019



Fonte: elaborado pelo autor.

Como pode ser visto, a previsão apresentada na Figura 10 apresenta um excelente resultado, em que o modelo conseguiu identificar a direção e intensidade que a ação se movimentou, muito próximo do que realmente se sucedeu. Na Figura 11, é possível verificar que o modelo previu que uma alta iria ocorrer, mas em uma acentuação menor do que realmente ocorreu. Por último na Figura 12, é apresentado uma previsão errônea, em que o modelo previu a direção oposta ao movimento. Os valores estão disponíveis na Tabela 4 e 5.

Para auferir a validade do modelo, e garantir estatisticamente a probabilidade de significância, utilizou-se o Valor-P. O Valor-P é utilizado para medir a probabilidade de obter um resultado ao acaso, mensurando se os valores possuem evidência estatística ou não. As forças das hipóteses nula e alternativa são calculadas com o Valor-P, assumindo que a hipótese nula está correta. A hipótese nula significa que não há diferença significativa entre dois grupos, enquanto a hipótese alternativa significa que há diferenças estatisticamente significativas entre os dois grupos (BEERS, 2019). Os valores calculados são apresentados na Tabela 2.

Tabela 2 – Cálculo do Valor-P

RADL3	1,1646E-28
VALE3	1,9498E-05
BBDC4	5,3644E-06
PETR4	4,7789E-15
WEGE3	3,1948E-29

Fonte: elaborado pelo autor.

Todas as ações tiveram resultados estatisticamente significantes, devido ao Valor-P ser menor do que 0,05. Segundo Beers (2019), este é um limiar comumente utilizado, e indica fortes evidências de que os resultados não são devido ao acaso. As ações WEGE3 e RADL3 obtiveram os melhores resultados devido ao Valor-P ser significativamente menor do que as demais.

Outra forma para validação do desempenho do algoritmo nos testes foi o resultado da previsão de alta ou baixa do ativo. Para isso foi validado se na semana em questão, o ativo estava subindo ou caindo, e comparado com a previsão do modelo. Para cada semana, foi constatado se os valores de segunda-feira foram maiores ou menores do que os valores da sexta-feira, classificando em alta ou baixa. Para cada classificação, comparou-se com os valores obtidos pelo modelo para averiguar a taxa de acerto. As validações ocorreram durante todo o período analisado, totalizando 10 semanas, e os resultados encontram-se na Tabela 2.

Tabela 3 – Taxa de acerto das movimentações

Ação	Taxa de acerto
RADL3	70%
VALE3	30%
BBDC4	40%
PETR4	60%
WEGE3	70%

Fonte: elaborado pelo autor.

Assim como apresentado na Tabela 2, os resultados da RADL3 e WEGE3 alcançaram os melhores números, na Tabela 3. O modelo previu corretamente a alta ou baixa na cotação em 7 de 10 semanas. As ações VALE3 e BBDC4 tiveram, respectivamente, os piores resultados, chegando no valor de 30% e 40% de acerto da previsão, provavelmente devido à alta volatilidade dos ativos. Todas as previsões, realizadas no início de cada semana, encontram-se nas tabelas 4 e 5.

Tabela 4 – Histórico dos valores e previsões entre 16/09/2019 e 18/10/2019

data	RADL3		VALE3		BBDC4		PETR4		WEGE3	
	previsão	valor real	previsão	valor real	previsão	valor real	previsão	valor real	previsão	valor real
16/09/2019	94,71	92,63	48,96	48,59	33,15	33,02	28,14	28,06	23,24	23,24
17/09/2019	94,69	94,07	49,14	48,90	33,27	33,90	28,23	27,69	23,38	23,80
18/09/2019	95,87	92,16	49,18	48,40	33,45	34,00	28,35	27,22	23,47	24,00
19/09/2019	97,13	93,06	49,33	48,32	33,58	33,59	28,44	27,29	23,57	23,69
20/09/2019	97,32	93,61	49,39	48,42	33,63	34,18	28,59	27,00	23,66	24,40
23/09/2019	97,04	93,80	48,26	48,10	34,28	34,25	27,56	27,48	24,15	24,20
24/09/2019	96,79	96,40	48,36	46,93	34,44	33,81	27,61	27,27	24,31	23,88
25/09/2019	98,32	95,92	48,42	47,86	34,61	34,03	27,65	27,34	24,40	24,24
26/09/2019	100,78	96,08	48,52	47,86	34,75	34,52	27,64	27,70	24,49	24,24
27/09/2019	100,49	96,15	48,54	47,66	34,79	34,50	27,67	26,66	24,58	24,13
30/09/2019	99,21	95,81	47,86	47,75	34,16	33,92	27,63	27,55	24,24	24,23
01/10/2019	99,01	95,93	47,94	47,71	34,36	33,46	27,67	27,51	24,41	24,23
02/10/2019	100,81	95,20	48,02	45,10	34,54	32,15	27,69	26,72	24,51	23,83
03/10/2019	104,66	95,56	48,12	45,44	34,71	32,24	27,70	26,74	24,58	23,68
04/10/2019	103,61	101,50	48,18	46,59	34,79	32,57	27,69	26,51	24,66	23,61
07/10/2019	103,45	102,13	46,08	46,04	32,81	32,36	26,24	26,17	22,85	22,85
08/10/2019	103,47	100,01	46,27	45,32	33,17	32,33	26,30	26,02	23,05	22,80
09/10/2019	104,43	101,46	46,36	45,67	33,54	33,25	26,34	26,52	23,17	22,87
10/10/2019	105,41	100,71	46,55	47,24	33,87	33,63	26,69	26,74	23,26	22,97
11/10/2019	105,57	100,98	46,67	48,64	34,05	34,00	26,43	27,26	23,32	23,67
14/10/2019	103,55	101,05	48,08	47,99	34,77	34,49	27,39	27,31	23,76	23,75
15/10/2019	103,50	101,12	48,26	47,91	34,98	34,77	27,54	27,60	23,88	23,78
16/10/2019	104,81	102,88	48,29	46,80	34,05	35,60	27,63	27,93	23,97	24,37
17/10/2019	106,52	101,45	48,49	46,71	35,15	35,00	27,79	27,66	24,06	24,24
18/10/2019	106,59	103,63	48,34	46,03	35,21	33,79	27,56	27,60	24,14	24,36

Fonte: Elaborado pelo autor.

Tabela 5 – Histórico dos valores e previsões entre 21/10/2019 e 23/11/2019

data	RADL3		VALE3		BBDC4		PETR4		WEGE3	
	previsão	valor real	previsão	valor real	previsão	valor real	previsão	valor real	previsão	valor real
21/10/2019	107,30	105,55	47,42	47,21	34,34	34,03	27,66	27,77	24,47	24,44
22/10/2019	107,33	106,21	47,48	47,30	34,44	35,00	27,59	28,57	24,64	24,74
23/10/2019	108,39	103,38	47,68	47,11	34,53	35,30	27,49	28,95	24,75	25,84
24/10/2019	109,54	105,50	47,76	46,75	34,61	35,26	27,34	28,32	24,85	26,10
25/10/2019	109,79	107,05	47,90	48,56	34,79	35,72	27,25	29,25	24,94	25,41
28/10/2019	109,15	107,00	48,67	48,70	36,91	37,01	29,55	29,60	25,62	25,58
29/10/2019	109,20	109,40	48,79	48,65	36,90	36,50	29,50	29,82	25,79	25,52
30/10/2019	110,35	109,30	48,83	48,59	36,86	36,67	29,52	30,08	25,90	25,85
31/10/2019	111,69	110,00	48,97	47,20	36,85	35,17	29,57	30,39	25,99	25,50
01/11/2019	111,90	112,97	49,10	48,44	36,92	34,87	29,64	30,43	26,08	25,88
04/11/2019	115,30	113,80	49,74	49,82	35,11	34,89	30,51	30,36	26,08	26,13
05/11/2019	115,49	111,89	49,74	49,87	35,25	35,36	30,50	29,65	26,25	26,74
06/11/2019	116,38	112,64	49,80	49,69	35,44	35,17	30,51	29,71	26,34	27,17
07/11/2019	117,37	112,05	49,85	50,11	35,62	35,26	30,48	30,90	26,42	28,09
08/11/2019	117,65	111,50	49,95	49,18	35,84	34,24	30,43	30,02	26,50	28,24
11/11/2019	114,45	112,00	48,49	48,13	34,64	34,16	30,30	30,45	29,04	29,09
12/11/2019	114,63	109,99	48,68	47,90	34,96	33,84	30,21	30,02	29,14	28,50
13/11/2019	115,90	108,49	48,74	47,12	35,23	33,49	30,19	29,90	29,21	29,21
14/11/2019	117,43	112,20	48,87	47,00	35,50	33,35	30,09	29,30	29,27	29,26
15/11/2019	117,76	111,35	48,91	47,61	35,76	33,13	30,12	29,08	29,39	29,30
19/11/2019	114,50	111,80	48,16	48,17	33,74	33,07	28,92	28,78	29,05	29,20
20/11/2019	114,52	112,70	48,21	48,42	34,19	33,26	29,01	29,85	29,27	28,88
21/11/2019	116,02	113,75	48,29	50,03	34,53	33,81	29,04	29,98	29,37	29,06
22/11/2019	117,59	111,54	48,38	50,86	34,86	33,64	29,05	29,73	29,45	29,10
23/11/2019	117,87	111,90	48,46	51,22	35,12	32,92	29,11	29,19	29,54	28,63

Fonte: elaborado pelo autor.

## 5 CONCLUSÕES

Este trabalho apresentou o desenvolvimento de um protótipo para trading automatizado utilizando Redes Neurais Artificiais. Para isso foi utilizado o modelo Long Short-Term Memory (LSTM), um tipo de Rede Neural Recorrente, objetivando a previsão do preço de ações cotadas na bolsa de valores brasileira, B3 S.A., alcançando resultados consistentes. O projeto não aborda questões como o envio de ordens de compra e venda, considerada parte vital para o funcionamento da plataforma de *trading*. Isso devido à complexidade técnica e inviabilidade financeira para o protótipo, tendo em vista que todas as APIs para o roteamento têm um custo elevado.

Para a construção do modelo Long Short-Term Memory foram necessárias fazer alterações na sua estrutura, com o intuito de realizar a previsão de mais de um valor por vez. Tal mudança traz consigo uma abordagem que possibilita ao investidor ter uma visão mais abrangente, mas por outro lado, a imprevisibilidade e fragilidade do mercado perante as notícias, reduz a acurácia do modelo. Mesmo assim, essa abordagem mostrou-se num primeiro momento ideal para o protótipo, porque além de aumentar a perspectiva, evita o custo computacional e tempo na realização das previsões diárias para cada *trade* realizado. A interface com o usuário através de uma plataforma WEB foi de suma importância para acompanhamento das previsões e gestão dos investimentos, embora ainda esteja em fase de desenvolvimento.

Para mensuração da acurácia, foram estabelecidas 10 semanas de teste, contemplando um período ideal, tendo em vista que o mercado passou por altas e baixas. A validação ocorreu através do cálculo do Valor-P, provando que todas as cinco ações tiveram resultados estatisticamente significantes, indicando com fortes evidências de que os resultados não são devido ao acaso. Além disso, realizou-se uma comparação, semana-a-semana, para determinar se a previsão foi

assertiva e teve resultados consistentes, alcançando resultados superiores a 60% de acerto para três de cinco ações, sendo que duas resultaram em 70% de acerto.

A solução ainda conta com atraso de 15 minutos na cotação, em decorrência da API do Yahoo Finance, mas não compromete nenhum resultado apresentado. Como extensões para este trabalho, é essencial o incremento de mais *inputs* para a Rede Neural Artificial, como indicadores utilizados na análise técnica baseados em cálculos, como Índice de Força Relativa, Média Móvel Convergente e Divergente, volume de negociações, etc. Estes dados tendem a conter informações relevantes sobre a movimentação dos preços, como regiões de sobrecompra, sobrevenda, força da tendência, etc. Com isso, o incremento nas taxas de acerto deve ocorrer. Além do mais, faz-se necessário a implementação referente ao envio das ordens ao mercado, gerenciamento da posição, ou uma conexão através de API que possibilite as negociações de fato ocorrerem. Quanto ao desenvolvimento interface web, é necessário dar continuidade à implementação, principalmente nas telas para visualização das ações em carteira, criação de uma área para gerenciamento dos investimentos e controle de risco.

Através da utilização de Redes Neurais Artificiais LSTM, foi possível atingir resultados íntegros para previsão de ações no mercado financeiro, embora essas técnicas ainda sejam pouco utilizadas. Dessa forma, o protótipo visa contribuir para os trabalhos e pesquisas futuros no meio acadêmico que venham a utilizar Aprendizado de Máquina como alicerce para o mercado financeiro.

## REFERÊNCIAS

- ARBEL, Nir. **How LSTM networks solve the problem of vanishing gradients**. 2018. Disponível em: <<https://medium.com/datadriveninvestor/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577>>. Acesso em: 26 nov. 2019.
- BARBER, Brad et al. Do day traders rationally learn about their ability. **Unpublished Working Paper. University of California at Berkeley**, 2010.
- BAO, Wei; YUE, Jun; RAO, Yulei. **A deep learning framework for financial time series using stacked autoencoders and long-short term memory**. 2017. 24 f. Monografia (Especialização) - Curso de Negócios, Central South University, Changsha, 2016. Disponível em: <<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0180944>>. Acesso em: 23 mar. 2019.
- BEERS, Brian. **P-Value Definition**. 2019. Disponível em: <<https://www.investopedia.com/terms/p/p-value.asp>>. Acesso em: 26 nov. 2019.
- BROWNLEE, Jason. **How to Develop LSTM Models for Time Series Forecasting**. 2018. Disponível em: <<https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/>>. Acesso em: 12 out. 2019.
- CHEN, Guanting; CHEN, Yatang; FUSHIMI, Takahiro. **Application of Deep Learning to Algorithmic Trading**. 2017. 6 f. Monografia (Especialização) - Curso de Engenharia Computacional e Matemática, Stanford University, Califórnia, 2017.
- CHEN, James; BEERS, Brian. **Buy and Hold Definition: What Is Buy and Hold?** 2019. Disponível em: <<https://www.investopedia.com/terms/b/buyandhold.asp>>. Acesso em: 09 abr. 2019.
- ELDER, Alexander. **Como se transformar em um operador e investidor de sucesso**. 15. ed. São Paulo: Elsevier, 2017. 320 p. (41). Tradução de: Afonso Celso de Cunha Serra.
- FABBRI, Mirco; MORO, Gianluca. **Dow Jones Trading with Deep Learning: The Unreasonable Effectiveness of Recurrent Neural Networks**. 2018. 12 f. Monografia (Especialização) - Curso de Ciência da Computação, Ciência da Computação e Engenharia, Universidade de Bolonha, Cesena, 2018. Disponível em: <<https://www.scitepress.org/papers/2018/69221/69221.pdf>>. Acesso em: 23 mar. 2019.
- FRANKENFIELD, Jake. **Trade: What is a Trade**. 2019. Disponível em: <<https://www.investopedia.com/terms/t/trade.asp>>. Acesso em: 06 abr. 2019.
- GRANT, Mitchell. **Understanding Basic Candlestick Charts**. 2019. Disponível em: <<https://www.investopedia.com/trading/candlestick-charting-what-is-it/>>. Acesso em: 23 nov. 2019.
- GUPTA, Anupriya. **History of Algorithmic Trading, HFT and News Based Trading**. 2015. Disponível em: <<https://www.quantinsti.com/blog/history-algorithmic-trading-hft/>>. Acesso em: 26 mar. 2019.
- HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. **Neural computation**, v. 9, n. 8, p. 1735-1780, 1997.
- HOPFIELD, John J. Neural networks and physical systems with emergent collective computational abilities. **Proceedings of the national academy of sciences**, v. 79, n. 8, p. 2554-2558, 1982.
- MARGINS-FILHO, Luiz; MOL, Adriano.
- MAJASKI, Christina. **Fundamental vs. Technical Analysis: What's the Difference?** 2019. Disponível em: <<https://www.investopedia.com/ask/answers/difference-between-fundamental-and-technical-analysis/>>. Acesso em: 26 out. 2019.
- MCCULLOCH, Warren S.; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115-133, 1943.



M'NG, Jacinta Chan Phooi; MEHRALIZADEH, Mohammadali. **Forecasting East Asian Indices Futures via a Novel Hybrid of Wavelet-PCA Denoising and Artificial Neural Network Models**. 2016. 29 f. Monografia (Especialização) - Curso de Negócios e Contabilidade, University Of Malaya, Kuala Lumpur, 2016. Disponível em: <<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0156338>>. Acesso em: 23 mar. 2019.

OLAH, Christopher. **Understanding LSTM Networks**. 2015. Disponível em: <<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>>. Acesso em: 23 out. 2019.

PINHEIRO, Juliano Lima. **História da Bolsa de Valores**. 2014. Disponível em: <[http://www.mercadocomum.com/site/artigo/detalhar/historia\\_da\\_bolsa\\_de\\_valores/materias-publicadas](http://www.mercadocomum.com/site/artigo/detalhar/historia_da_bolsa_de_valores/materias-publicadas)>. Acesso em: 23 mar. 2019.

ROCHA, Ronilson. **Desenvolvimento de ferramenta computacional para auxílio ao projeto de gemas lapidadas**. 2005. 7 f. Monografia (Especialização) - Curso de Ciência da Computação, Universidade Federal de Ouro Preto, Ouro Preto, 2005. Disponível em: <<http://www.scielo.br/pdf/rem/v58n4/a060.pdf>>. Acesso em: 11 nov. 2019.

RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência Artificial**. 3. ed. Rio de Janeiro: Elsevier Editora Ltda, 2013. 1056 p. Tradução de: Regina Célia Simille.

SRIVASTAVA, Tavish. **How does Artificial Neural Network (ANN) algorithm work? Simplified!** 2014. Disponível em: <<https://www.analyticsvidhya.com/blog/2014/10/ann-work-simplified/>>. Acesso em: 24 nov. 2019.

VANDEPUT, Nicolas. **Forecast KPI: RMSE, MAE, MAPE & Bias**. 2019. Disponível em: <<https://medium.com/analytics-vidhya/forecast-kpi-rmse-mae-mape-bias-cdc5703d242d>>. Acesso em: 25 nov. 2019.

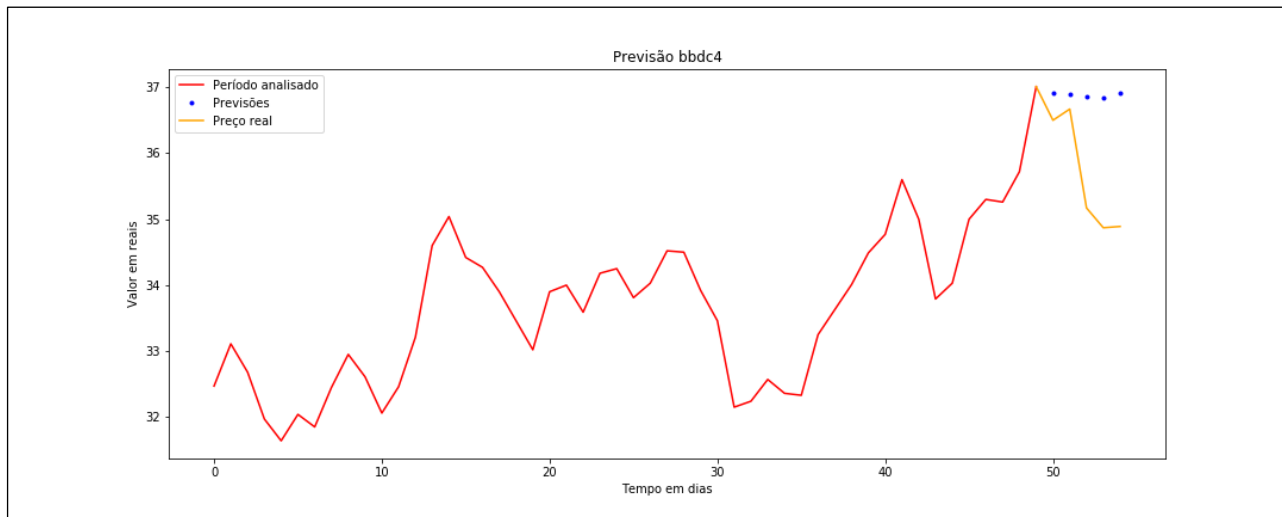
VARSAMOPOULOS, Savvas; BERTELS, Koen; ALMUDEVER, Carmen G. Designing neural network based decoders for surface codes. **arXiv preprint arXiv:1811.12456**, 2018.

VERSIGNASSI, Alexandre. **CRASH: Uma breve história da economia - da Grécia antiga ao século XXI**. 2. ed. São Paulo: Leya, 2015.

## APÊNDICE A – PREVISÕES REALIZADAS ENTRE 28/10/2019 E 01/11/2019

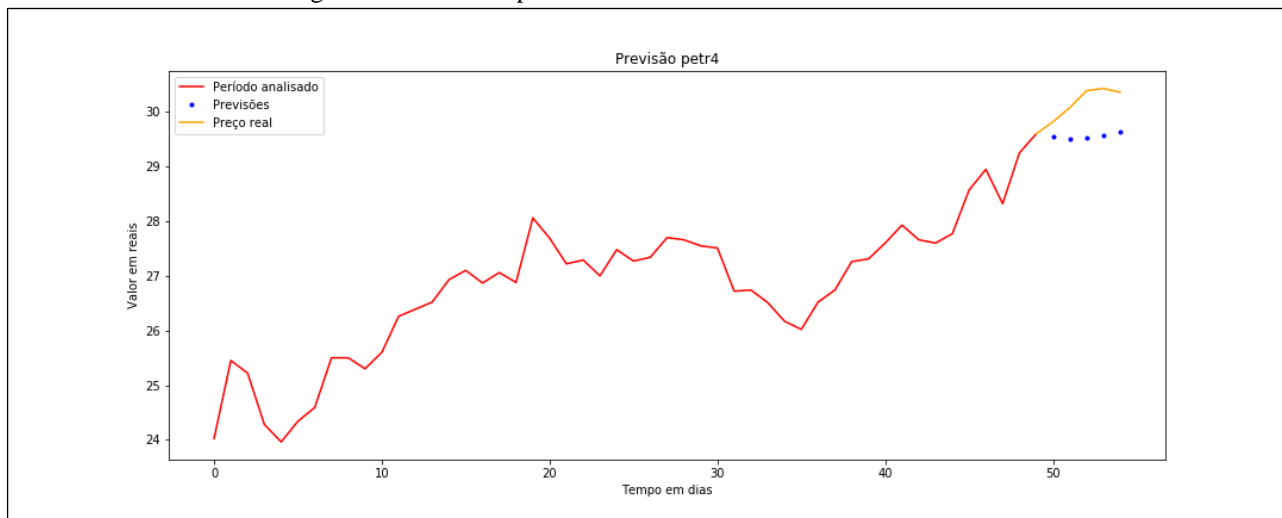
Abaixo, as previsões para Bradesco, Petrobras, Raia Drogasil, Vale SA e WEG entre as datas de 28/10/2019 e 01/11/2019.

Figura 13 – Previsão para Bradesco entre 28/10/2019 e 01/11/2019



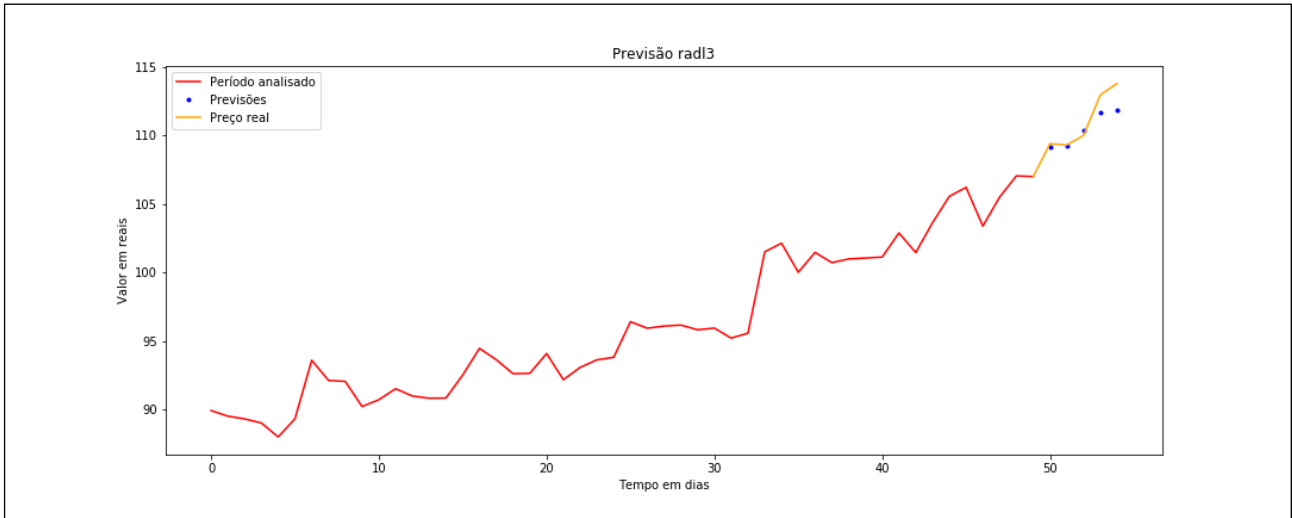
Fonte: elaborado pelo autor.

Figura 14 – Previsão para Petrobras entre 28/10/2019 e 01/11/2019



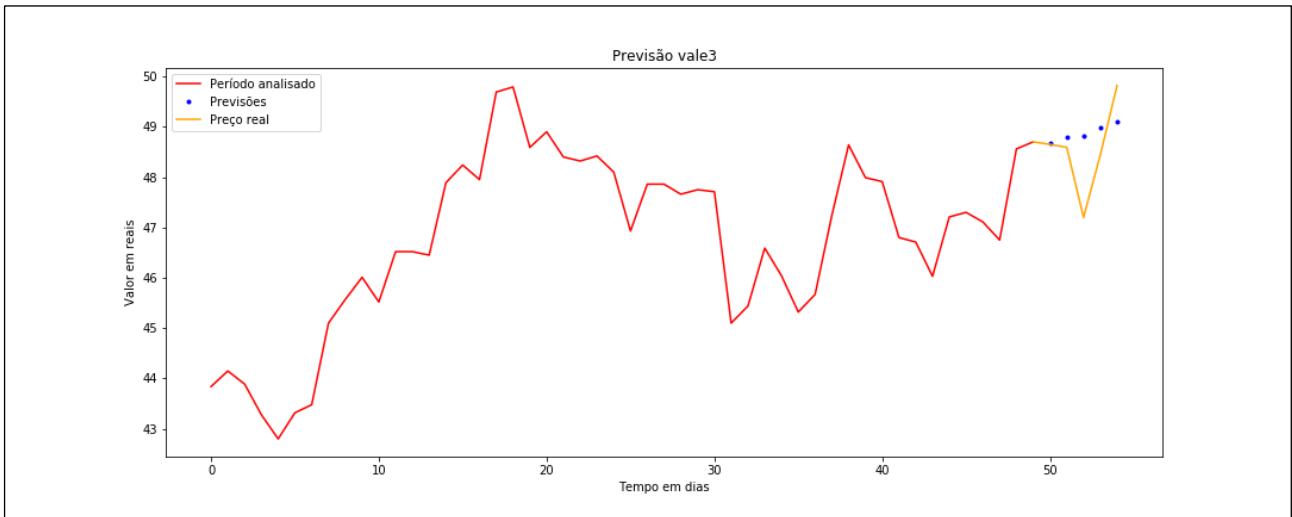
Fonte: elaborado pelo autor.

Figura 15 – Previsão para Raia Drogasil entre 28/10/2019 e 01/11/201



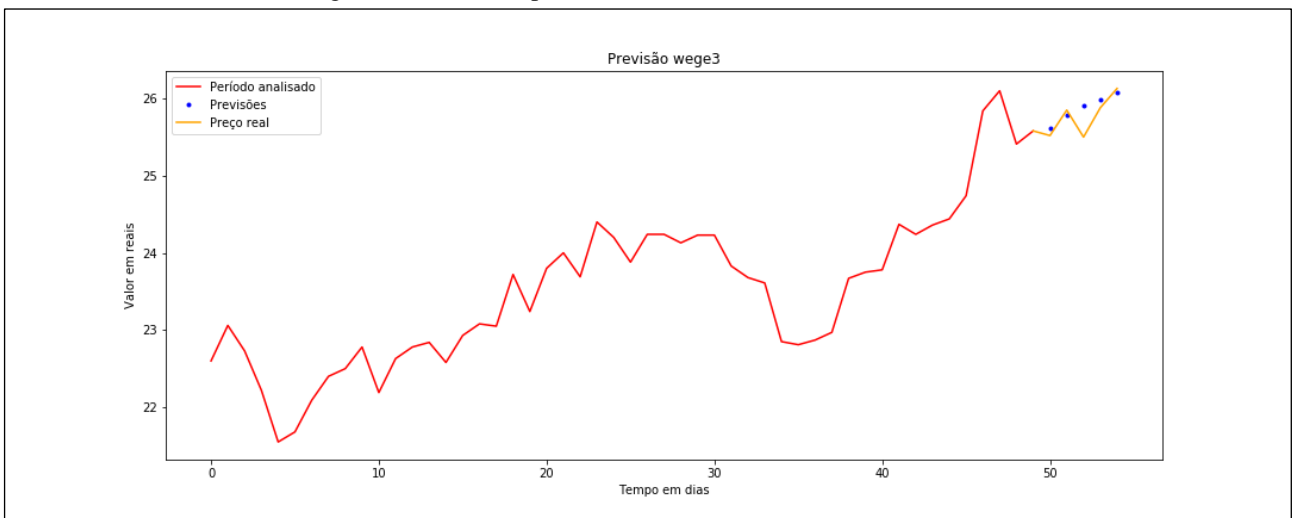
Fonte: elaborado pelo autor.

Figura 16 – Previsão para Vale SA entre 28/10/2019 e 01/11/201



Fonte: elaborado pelo autor.

Figura 17 - Previsão para WEG entre 28/10/2019 e 01/11/2019



Fonte: elaborado pelo autor.