

INSTITUTO FEDERAL DE SANTA CATARINA

MARCELO BITTENCOURT DO NASCIMENTO FILHO

**Desenvolvimento de um sistema de apoio à
tomada de decisões para o setor da
agroindústria utilizando dados microclimáticos e
técnicas de aprendizado de máquina**

São José - SC

Março/2022

DESENVOLVIMENTO DE UM SISTEMA DE APOIO À TOMADA DE DECISÕES PARA O SETOR DA AGROINDÚSTRIA UTILIZANDO DADOS MICROCLIMÁTICOS E TÉCNICAS DE APRENDIZADO DE MÁQUINA

Trabalho de conclusão de curso apresentado à Coordenadoria do Curso de Engenharia de Telecomunicações do campus São José do Instituto Federal de Santa Catarina para a obtenção do diploma de Engenheiro de Telecomunicações.

Orientador: Mário de Noronha Neto

Coorientador: Hamilton Avinco

São José - SC

Março/2022

MARCELO BITTENCOURT DO NASCIMENTO FILHO

**DESENVOLVIMENTO DE UM SISTEMA DE APOIO À
TOMADA DE DECISÕES PARA O SETOR DA
AGROINDÚSTRIA UTILIZANDO DADOS
MICROCLIMÁTICOS E TÉCNICAS DE APRENDIZADO
DE MÁQUINA**

Este trabalho foi julgado adequado para obtenção do título de Engenheiro de Telecomunicações, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

São José - SC, 17 de março de 2022:

Mário de Noronha Neto, Dr.
Orientador
Instituto Federal de Santa Catarina

Arliones Hoeller Jr, Dr.
Instituto Federal de Santa Catarina

Richard Demo Souza, Dr.
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus pais e a minha irmã.

AGRADECIMENTOS

Agradeço aos meus pais e a minha irmã pelo suporte desde o início de minha graduação, me incentivando a sempre seguir em frente e a enfrentar as dificuldades. Vocês são e serão a minha base de segurança e carinho.

Agradeço ao professor Mário de Noronha por toda a sua atenção, paciência e entusiasmo diante desta pesquisa. Foram muitas horas de conversas e reuniões para o desenvolvimento do trabalho que valeram o esforço.

Agradeço ao Instituto Federal de Santa Catarina - Câmpus São José e a todos os seus colaboradores pelo ótimo trabalho desempenhado e pela entrega de uma educação pública de qualidade.

Por fim, agradeço a todos que de alguma forma fizeram parte dessa jornada desafiadora. Obrigado!

*“Talvez não devêssemos jamais procurar
um novo traje, por mais gasto e sujo
que esteja o velho, até nos sentirmos
novos homens dentro das roupas antigas.“*
(H.D. Thoreau)

RESUMO

O setor agropecuário é essencial para o desenvolvimento socioeconômico do país, sendo o Brasil um dos principais exportadores de grãos do mundo. Porém, durante o processo de produção ocorrem perdas significativas causadas pela dificuldade de gestão e de monitoramento do plantio. Um dos pontos mais importantes para a redução de gastos e na assertividade em resolver esse problema é realizando análises microclimáticas, ou seja, avaliar mais especificamente os dados climáticos da região. Dessa forma, o proprietário pode tomar decisões precisas, como por exemplo se deve ou não realizar certa atividade nos próximos dias. A relação entre esse setor e com soluções tecnológicas que seguem o conceito de *Internet das Coisas* e que aplicam a análise de dados para um melhor gerenciamento das atividades está cada vez maior, gerando assim o cenário do Agro 4.0. Portanto, este trabalho visa aplicar um algoritmo baseado em técnicas de Aprendizado de Máquina para a previsão de parâmetros climáticos, a fim de proporcionar ao agricultor maior segurança em suas ações e o aumento do valor de seu produto.

Palavras-chave: Internet das Coisas. Aprendizado de Máquina. Agroindústria. Micro-clima.

ABSTRACT

The agricultural sector is essential for the socioeconomic development of the country, with Brazil being one of the main grain exporters in the world. However, during the production process there are significant losses caused by the difficulty of managing and monitoring the plantation. One of the most important points for reducing expenses and assertiveness in solving this problem is performing microclimate analyses, that is, evaluating more specifically the climatic data of the region. In this way, the owner can make precise decisions, such as whether or not to carry out a certain activity in the next few days. The relationship between this sector and technological solutions that follow the concept of the Internet of Things and that apply data analysis for better management of activities is increasing, thus generating the scenario of Agro 4.0. Therefore, this work aims to apply an algorithm based on Machine Learning techniques for the prediction of climatic parameters, in order to provide the farmer with greater security in his actions and increase the value of his product.

Keywords: Internet of Things. Machine Learning. Agribusiness. Microclimate.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de aplicação <i>Internet of Things</i> (IoT) para o controle de variáveis meteorológicas e do solo.	19
Figura 2 – Escalas climáticas	21
Figura 3 – Estrutura de um Sistema de Apoio à Decisões.	22
Figura 4 – Estação Meteorológica para um sistema de <i>Internet das Coisas</i>	23
Figura 5 – Comparação entre tecnologias de transmissão sem fio.	24
Figura 6 – Exemplo de um Banco de Dados Relacional.	26
Figura 7 – Divisões da Inteligência Artificial.	30
Figura 8 – Etapas do Aprendizado de Máquina.	31
Figura 9 – Estrutura biológica de um neurônio.	32
Figura 10 – Modelo não-linear de um neurônio artificial.	33
Figura 11 – Exemplo de estrutura de uma Rede Neural Artificial.	34
Figura 12 – Exemplo básico de uma rede RNN.	35
Figura 13 – Exemplo rede RNN aberta.	35
Figura 14 – Célula RNN.	36
Figura 15 – Célula LSTM.	38
Figura 16 – Exemplo de sazonalidade.	40
Figura 17 – Exemplo de dataset.	40
Figura 18 – Estações Automáticas Instituto Nacional de Meteorologia (INMET) selecionadas para estudo.	45
Figura 19 – Bibliotecas <i>Machine Learning</i> (ML) mais utilizadas.	48
Figura 20 – Comportamento da variável Temperatura Máxima.	52
Figura 21 – Comportamento da variável Umidade Relativa do Ar.	53
Figura 22 – Comportamento da variável Pressão Atmosférica.	54
Figura 23 – Comportamento da variável Velocidade do Vento.	55
Figura 24 – Informações contidas em um <i>boxplot</i>	56
Figura 25 – <i>Boxplot</i> das variáveis em estudo.	57
Figura 26 – Lógica do Cenário de Próxima Hora.	59
Figura 27 – Resultados Cenário 1.	60
Figura 28 – Lógica do Cenário de Modelos em períodos do dia	62
Figura 29 – Resultados Cenário 2.	65
Figura 30 – Resultados Cenário 3.	68
Figura 31 – Página de monitoramento do sistema.	70
Figura 32 – Página de gráfico dos dados reais.	71
Figura 33 – Página de gráfico dos dados previstos.	72

Figura 34 – Previsão de temperatura utilizando a plataforma.	74
Figura 35 – Previsão de umidade utilizando a plataforma.	75
Figura 36 – Estrutura <i>back-end</i> com Django <i>REST framework</i>	82

LISTA DE TABELAS

Tabela 1 – Estações Automáticas INMET do Estado do Paraná.	44
Tabela 2 – Estações Automáticas INMET do Estado de Santa Catarina.	44
Tabela 3 – Estações Automáticas INMET do Estado do Rio Grande do Sul.	45
Tabela 4 – Análise descritiva da estação A806.	57
Tabela 5 – Métricas de erros para o Cenário de Próxima Hora.	59
Tabela 6 – Modelos e seus respectivos horários de predição para o Cenário 2. . . .	61
Tabela 7 – Métricas de erros para o Cenário de Modelos em períodos do dia ($a = 20$). .	64
Tabela 8 – Métricas de erros para o Cenário de Modelos em períodos do dia ($a = 1$). .	64
Tabela 9 – Modelos e seus respectivos horários de predição para o Cenário 3. . . .	66
Tabela 10 – Métricas de erros para o Cenário de Modelos subamostrados ($a = 20$). .	67
Tabela 11 – Métricas de erros para o Cenário de Modelos subamostrados ($a = 1$). .	67
Tabela 12 – Métricas de erros para simulação de temperatura e umidade na plataforma.	73

LISTA DE CÓDIGOS

Código 2.1 – Exemplo de formatação JSON	25
Código 3.1 – LSTM utilizando Keras	49
Código A.1 – API <i>endpoint</i> para previsão	83

LISTA DE ABREVIATURAS E SIGLAS

IoT <i>Internet of Things</i>	8
M2M <i>Machine-to-Machine</i>	18
IA Inteligência Artificial	19
API <i>Application Programming Interface</i>	24
SAD Sistema de Apoio à Decisão	22
LPWAN <i>Low Power Area Wide Network</i>	23
HTTP <i>Hypertext Transfer Protocol</i>	24
BD Banco de Dados	25
INMET Instituto Nacional de Meteorologia	8
AGI <i>Artificial General Intelligence</i>	28
ANI <i>Artificial Narrow Intelligence</i>	28
ML <i>Machine Learning</i>	8
DL <i>Deep Learning</i>	29
RNA Rede Neural Artificial	29
ST Série Temporal	39
RNA Redes Neurais Artificiais	29
LSTM <i>Long-Short Term Memory</i>	37
MAE <i>Mean Absolute Error</i>	41
RMSE <i>Root Mean Square Error</i>	41
MAPE <i>Mean Absolute Percentage Error</i>	41
RNN <i>Recurrent Neural Network</i>	34

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivo Geral	16
1.2	Objetivos específicos	17
1.3	Organização do texto	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Internet das Coisas na Agroindústria	18
2.2	Microclimas	20
2.3	Componentes de um sistema IoT para apoio à decisão	22
2.3.1	Captura dos dados	22
2.3.2	Comunicação sem fio	23
2.3.3	Envio dos dados para uma API	24
2.3.4	Armazenamento dos dados	25
2.3.5	Computação em Nuvem	27
2.3.6	Aplicação web e móvel	27
2.4	Inteligência Artificial	28
2.4.1	Aprendizado de Máquina	29
2.4.2	Etapas do Aprendizado de Máquina	30
2.5	Redes Neurais Artificiais	31
2.5.1	Perceptron	32
2.6	Redes Neurais Recorrentes	34
2.7	Long-Short Term Memory	37
2.8	Séries Temporais	39
2.8.1	Tipos de variações em Séries Temporais	39
2.8.2	Estrutura dos dados para Séries Temporais	40
2.9	Cálculo de erros	41
2.9.1	Mean Absolute Error	41
2.9.2	Root Mean Square Error	41
2.9.3	Mean Absolute Percentage Error	41
3	DESENVOLVIMENTO	42
3.1	Base de dados	42
3.2	Estações meteorológicas selecionadas	43
3.2.1	Estações meteorológicas do Paraná	44
3.2.2	Estações meteorológicas de Santa Catarina	44

3.2.3	Estações meteorológicas do Rio Grande do Sul	44
3.3	Descrição da base de dados	46
3.3.1	Dados Brutos	46
3.3.2	Seleção dos arquivos utilizados	46
3.4	Biblioteca Tensorflow	48
3.5	Limpeza e formatação dos dados	50
3.6	Análise comportamental dos dados	51
3.6.1	Variáveis climáticas e estações meteorológicas de estudo	51
3.6.2	Análise descritiva dos dados da estação meteorológica A806	56
3.7	Estratégias desenvolvidas e resultados obtidos	58
3.7.1	Cenário 1: Próxima Hora	58
3.7.1.1	Modo de funcionamento	58
3.7.1.2	Resultados	59
3.7.2	Cenário 2: Modelos em períodos do dia	61
3.7.2.1	Modo de funcionamento	61
3.7.2.2	Resultados	63
3.7.3	Cenário 3: Modelos Subamostrados	66
3.7.3.1	Modo de funcionamento	66
3.7.3.2	Resultados	67
3.8	Desenvolvimento da página web	69
3.8.1	Tecnologias utilizadas	69
3.8.2	Páginas referentes ao monitoramento	69
3.8.3	Exemplo de previsão utilizando a plataforma	73
4	CONCLUSÕES	76
4.1	Trabalhos futuros	77
REFERÊNCIAS		78
APÊNDICES		81
APÊNDICE A – TRECHO ENDPOINT GET API PARA PREVISÃO LSTM		82

1 INTRODUÇÃO

O Brasil é um país privilegiado no quesito área territorial e de condições climáticas favoráveis para a prática agropecuária. Com isso, esse setor sempre foi destaque na economia brasileira, principalmente em cultivos para exportação como soja, milho e café. No ano de 2019 o Produto Interno Bruto do ramo agropecuário cresceu em 1,3%, mais que o dobro do crescimento de outras grandes áreas da economia como a Indústria, que teve uma elevação de 0,5% (IBGE, 2020).

Dados da *American Soybean Association* confirmam a grande influência do Brasil neste mercado, chegando a exportar cerca de 77 milhões de toneladas métricas na safra de 2019, ficando acima de grandes potências econômicas como os Estados Unidos que no mesmo período exportou 49,7 milhões de toneladas métricas (ASA, 2020).

Porém, mesmo com ótimos resultados, grande parte dos produtores do país ainda não utilizam recursos tecnológicos que aumentam sua produtividade devido a barreiras econômicas e até culturais. Um dos pontos principais para tornar todo o ciclo de produção mais rentável é o gerenciamento eficiente de insumos agrícolas como sementes, água, fertilizantes e defensivos (BNDES, 2017). O Brasil possui uma alta taxa de aplicação de defensivos, sendo quase 2,5 vezes maior que países como Austrália e Canadá (FAO, 2019). Grande parte dos produtos aplicados nas lavouras é devido a má gestão do proprietário, que não prevê determinados eventos que podem neutralizar a ação do mesmo, sendo obrigado a realizar o procedimento novamente, gastando tempo, agredindo o meio ambiente e diminuindo os seus lucros.

Atualmente, existem aplicações disponíveis que preveem o cenário climático de determinada região, porém grande parte delas trabalham com o conceito de macroclima. Com isso, não atendem completamente a necessidade de produtores que precisam maiores informações dos seus locais de plantação, que na maioria das vezes não seguem o que é previsto por essas ferramentas. Dessa forma, é necessário que novas aplicações sejam desenvolvidas, a fim de atuar com microclima, ou seja, coletar informações de temperatura, umidade do ar e do solo e pressão atmosférica com o objetivo de obter informações climáticas e meteorológicas mais específicas para a sua localidade.

Portanto, o profissional que adota as ferramentas corretas para o seu plantio e consegue antecipar a ocorrência de fenômenos que podem prejudicar sua produção, pode agir de forma mais eficiente para evitar gastos desnecessários com atividades não recomendadas para aquele período, gerando assim mais valor ao seu produto e consequentemente aumentando sua produtividade.

Com o avanço da tecnologia e o advento da transformação digital, novas soluções de monitoramento e prevenção foram desenvolvidas visando um melhor rendimento e conservação de insumos em diversos setores da economia. Técnicas de transmissão sem fio, análise de dados com aprendizado de máquinas e redes focadas no conceito de *Internet das Coisas*, são os principais responsáveis por essa mudança no cenário produtivo.

Todos esses conceitos quando aplicados no setor agropecuário geram grandes resultados, facilitando e melhorando o trabalho de diversas etapas da produção. O termo Agricultura 4.0 vem sendo utilizado para caracterizar essa relação entre a tecnologia e a administração do meio agrícola, e estima-se que no ano de 2020 já haja implementado mais de 75 milhões dispositivos conectados nas fazendas e lavouras em todo o mundo (CLERCQ; VATS; BIEL, 2018), gerando dados e monitorando plantios, silos, máquinas e até animais.

Por outro lado, no Brasil ainda existe uma resistência por parte dos proprietários em adotar soluções em suas terras. Isso acontece pois muitos agricultores têm receio sobre como utilizar os serviços e até desacreditam que o investimento pode trazer lucros. Porém, com a adesão de grandes produtores e uma nova geração de agropecuários surgindo, o mercado de tecnologia voltado para esse setor vem crescendo e evoluindo cada vez mais, proporcionando continuamente novas ferramentas para se ter o máximo de produtividade.

Tendo em vista as dificuldades de se obter uma maior eficiência na produção e as barreiras encontradas na previsão e monitoramento de plantios, este trabalho visa aplicar e estudar um algoritmo com técnicas de Aprendizado de Máquina para prever parâmetros climáticos que podem interferir nas atividades agrícolas, assim como fornecer um sistema de apoio para tomadas de decisão o qual disponibilizará de forma gráfica todas essas informações.

1.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um sistema de apoio à tomada de decisões para o setor da agroindústria utilizando dados microclimáticos, técnicas de aprendizado de máquina e estações meteorológicas comerciais. Os dados utilizados para o estudo serão obtidos de estações micrometeorológicas presentes nas áreas de plantio do usuário e de estações meteorológicas já existentes. No sistema, o usuário poderá visualizar seus dados capturados pelas estações e verificar a previsão de até dois dias para algumas variáveis climáticas.

1.2 Objetivos específicos

Visando atingir o objetivo geral, será necessário concluir os seguintes objetivos específicos:

1. Obter uma base de dados histórica, com um longo período de monitoramento;
2. Realizar a limpeza dos dados para sua análise;
3. Análise descritiva da base de dados;
4. Aplicar técnica para previsão de parâmetros climáticos;
5. Implementar um interface *web* para a visualização dos dados capturados e previstos.

1.3 Organização do texto

O trabalho em questão será dividido em três capítulos. O Capítulo 1, sendo este, aborda de forma introdutória o assunto que será estudo, assim como os objetivos que serão atingidos ao término do mesmo.

No Capítulo 2 há a apresentação de conceitos e termos que serão utilizados ao longo da pesquisa. É discutido o cenário de um sistema *Internet das Coisas* no ramo da agroindústria, assim como as suas etapas e componentes. Também é detalhado o conceito de microclima que será de extrema importância para o trabalho, e por fim há uma explicação sobre os principais tópicos relacionados à Inteligência Artificial e Aprendizado de Máquina.

Todas as atividades e etapas elaboradas neste trabalho estão apresentadas de forma detalhada no Capítulo 3, sendo discutidas as formas de tratamento dos dados utilizados, a base de informações escolhida e as estratégias de previsão dos parâmetros climáticos desenvolvidas.

Por fim, no Capítulo 4 há as considerações finais do projeto, onde são discutidos todos os resultados e dificuldades obtidos durante a sua realização. Neste capítulo também é ressaltado quais os trabalhos futuros a serem continuados a partir do resultado final da pesquisa, visando a sua melhoria em determinados pontos e implementação de novas funcionalidades e testes.

2 FUNDAMENTAÇÃO TEÓRICA

O presente capítulo visa detalhar e informar os mais variados conceitos e tecnologias utilizadas na pesquisa, a fim de proporcionar ao leitor um melhor entendimento sobre todo o trabalho desenvolvido.

2.1 Internet das Coisas na Agroindústria

A cada dia o conceito de *Internet* das Coisas, do inglês *Internet of Things* (IoT), vem ganhando maior notoriedade em diversas áreas do ramo tecnológico. Em sua essência, a IoT tem o objetivo de permitir a comunicação entre equipamentos, sensores ou máquinas com uma rede de informação através de comunicações sem fio, a fim de proporcionar um maior controle para o usuário do ambiente o qual está inserido. Um exemplo claro desse fenômeno são as casas autônomas, compostas por equipamentos que captam, processam e compartilham dados para o monitoramento de diversas atividades.

Grandes setores econômicos como o de saúde, indústrias e a agricultura adotaram o uso de máquinas inteligentes com o intuito de analisar seus resultados, para assim reduzir seus custos ou até mesmo prever problemas em suas produções. Dessa forma, a *Internet* das Coisas é uma das principais responsáveis por um novo modelo de negócio que proporcionará uma melhor qualidade de vida para os seus usuários.

É importante destacar que os próprios equipamentos, denominados de dispositivos embarcados, podem formar sub redes para se comunicarem entre si, criando assim os enlaces *Machine-to-Machine* (M2M) capazes de prover um sistema completo de informação independente de qualquer intervenção humana (ANTON-HARO; DOHLER, 2015). Esses dispositivos, em sua grande maioria, possuem características de pouco consumo de energia, baixa taxa de transmissão de dados e comunicam-se através de redes sem fio.

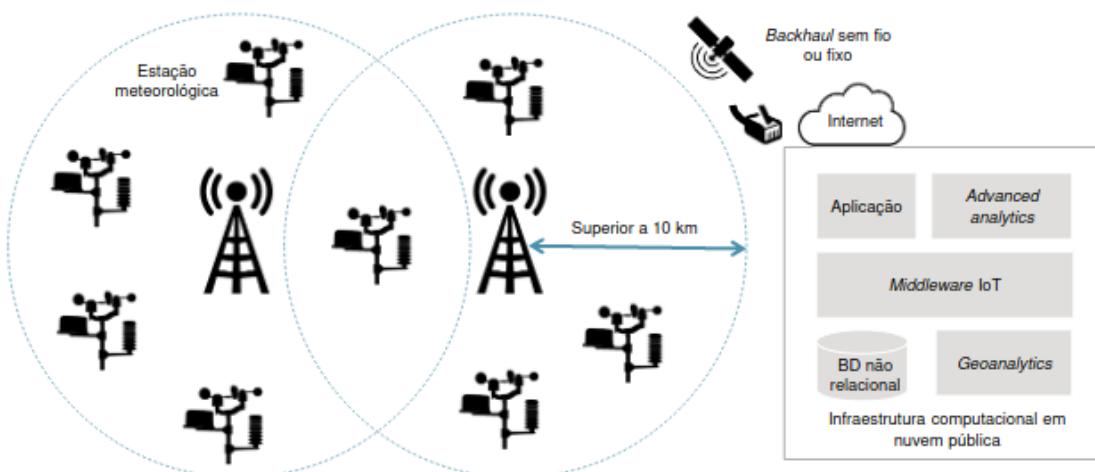
Como citado, um dos maiores setores em que a *Internet* das Coisas atua é na agroindústria. Os sistemas desenvolvidos para atender esse mercado visam a redução de custos e o aumento da produtividade nas mais variadas etapas do processo produtivo de alimentos, criação de animais e uso de máquinas. O foco das aplicações é sempre tornar o gerenciamento das atividades o mais acessível possível para o usuário, trazendo informações de futuros ou atuais problemas existentes nos seus produtos.

Os ramos de atuação da IoT no cenário do agro são diversos, variando desde modelos para gerir a produtividade humana até a supervisão real de máquinas que estão em operação. Porém, uma das áreas que mais necessita e é impactada positivamente por essas aplicações é o monitoramento meteorológico e do solo, as quais realizam a

captura de dados essenciais como temperatura, umidade e pressão atmosférica a partir de estações meteorológicas posicionadas na própria região de plantação a fim de trabalhar com microclimas. Com os dados coletados é realizado o processamento dessas variáveis que irão gerar informações de apoio para tomadas de decisões estratégicas ao agricultor, como por exemplo se o mesmo deve ou não realizar a irrigação em sua plantação (BNDES, 2017).

Através da Figura 1 é possível ter uma visão geral das fases existentes em um sistema dessa finalidade. As estações meteorológicas são implantadas nas localidades da propriedade do usuário, sendo possível existir várias em uma mesma região. Esses sensores instalados nas estações enviam os dados coletados para um servidor através de uma tecnologia de comunicação sem fio. As variáveis são armazenadas em banco de dados, processadas através de algoritmos de Inteligência Artificial (IA) e os resultados das operações são fornecidos ao usuário através de uma aplicação web ou móvel, as quais enviam alertas e demonstram gráficos para um melhor controle das atividades. A seção 2.3 realiza o detalhamento de todas as ferramentas e tecnologias utilizadas nas etapas de um sistema dessa natureza.

Figura 1 – Exemplo de aplicação IoT para o controle de variáveis meteorológicas e do solo.



Fonte: Adaptada de (BNDES, 2017).

2.2 Microclimas

A previsão das variáveis meteorológicas e climáticas é um processo fundamental para a correta realização de diversas atividades no setor agropecuário. Saber se o clima está favorável para o início da plantação de determinada cultura ou ter a certeza de que os parâmetros climáticos da sua região estejam em certos níveis de valores, é essencial para a redução de gastos desnecessários com insumos agrícolas.

Porém, grande parte dos produtores utilizam ferramentas ou informações que preveem esses fenômenos de forma não regionalizada, ou seja, fornecem dados de uma macro região, os quais muitas das vezes diferem das condições das fazendas e plantações. Esse problema é bastante grave, pois causa insegurança ao agricultor perante as suas decisões.

Dessa forma, surgiu a necessidade de se obter dados da própria região de cultivo, tais como temperatura, umidade do ar e do solo, índice de raios solares e entre outros, a fim de se obter informações mais específicas sobre o local. Esse conceito é denominado de microclima, e vem ganhando destaque no desenvolvimento de novas soluções para esse cenário.

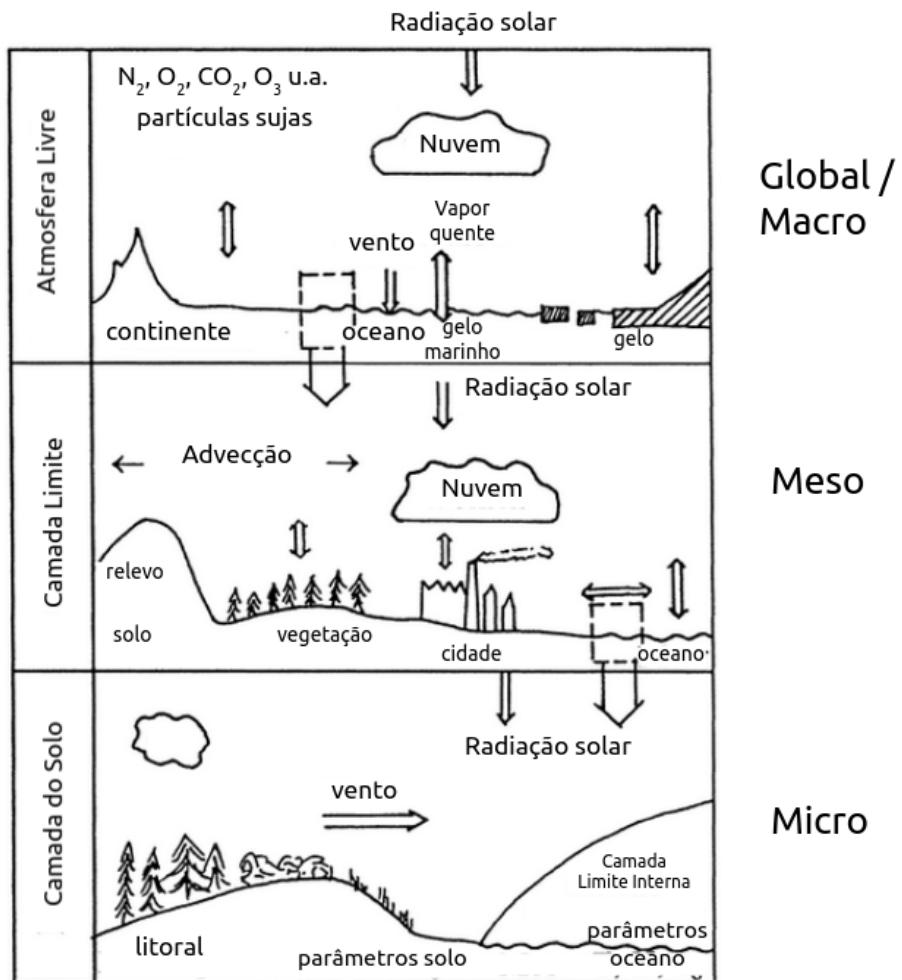
A Figura 2 demonstra as escalas climáticas existentes, sendo a maior delas denominada de macro, que abrange grandes altitudes e distâncias terrestres, e em algumas situações essa análise é aplicada envolvendo diferentes biomas como florestas, oceanos e até regiões de frio intenso. Dessa forma, o macroclima promove o conhecimento de possíveis fenômenos em uma longa região, sem diferenciar localidades.

Reduzindo a área de estudo para um raio de 10 km a 100 km de distância há a classificação do mesoclima. A escala meso geralmente utiliza como base as características do relevo, vegetação e influência da atividade humana como por exemplo as cidades.

Por fim, a análise microclimática é realizada em uma pequena área territorial de no máximo 1 km e em curtos intervalos de tempo (HUPFER, 1996). A escala do microclima leva em consideração fatores que estejam próximos à superfície (acima e abaixo dela) (FOKEN, 2017). Com isso, a percepção de pequenas variações nas características climáticas e meteorológicas da região, quando detectadas por sistemas, é extremamente valiosa pois indicam a real situação do local e tornam as atividades muito mais assertivas.

Dentre os principais parâmetros que influenciam o cenário microclimático, destacam-se: A **umidade solo** que está diretamente relacionada com a capacidade de retenção de calor do solo; a **temperatura do solo** que varia de acordo com o tipo de solo e a incidência de radiação sobre o mesmo; a **temperatura do ar** e **umidade do ar**; as condições do **vento**, como sua velocidade e direção (ISMANGIL et al., 2016).

Figura 2 – Escalas climáticas



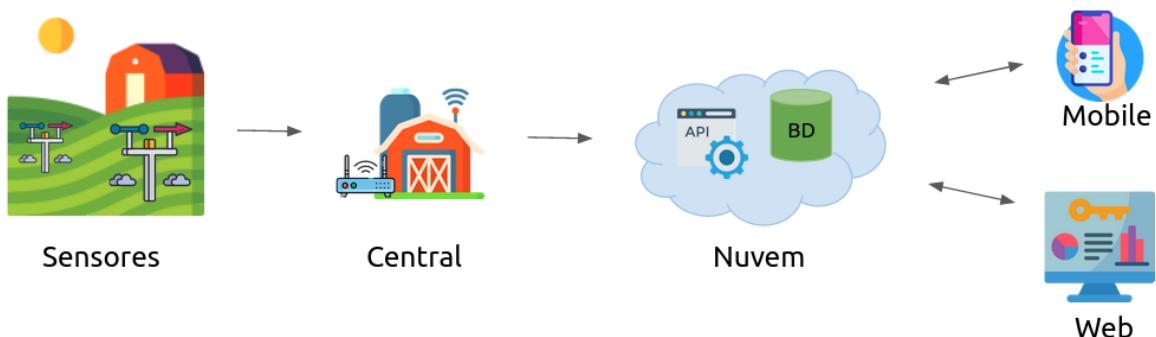
Fonte: Adaptada de (HUPFER, 1996).

2.3 Componentes de um sistema IoT para apoio à decisão

Como discutido na seção 2.1 é essencial que sistemas de IoT no ramo da agroindústria além de implementarem a parte física, como a instalação de sensores e de equipamentos de transmissão, deve ser realizada a análise detalhada dos dados coletados a fim de estabelecer relações, padrões e previsões para assim auxiliar o usuário no gerenciamento de sua propriedade. Aplicações completas que implementam todas as etapas citadas podem ser denominadas de Sistema de Apoio à Decisão (SAD).

A Figura 3¹ demonstra um diagrama de blocos para um melhor entendimento de todo o conjunto de um sistema e como as diferentes partes se interligam. A explicação de cada etapa está dividida nas subseções abaixo:

Figura 3 – Estrutura de um Sistema de Apoio à Decisões.



Fonte: O próprio autor utilizando imagens de Flaticon.

2.3.1 Captura dos dados

Nos sensores ocorre a coleta de dados necessários para o funcionamento da aplicação, sendo eles digitais ou analógicos. A Figura 4 retrata uma estação meteorológica voltada para esse tipo de aplicação, sua estrutura foi pensada para coletar informações como umidade do solo, temperatura do ar, direção e velocidade do vento e entre outros parâmetros. Todos esses dados são armazenados no próprio equipamento para posteriormente serem enviados à uma central, a qual será responsável por outra tarefa.

¹ Figuras de diferentes autores porém todas retiradas de Flaticon

Figura 4 – Estação Meteorológica para um sistema de *Internet das Coisas*.



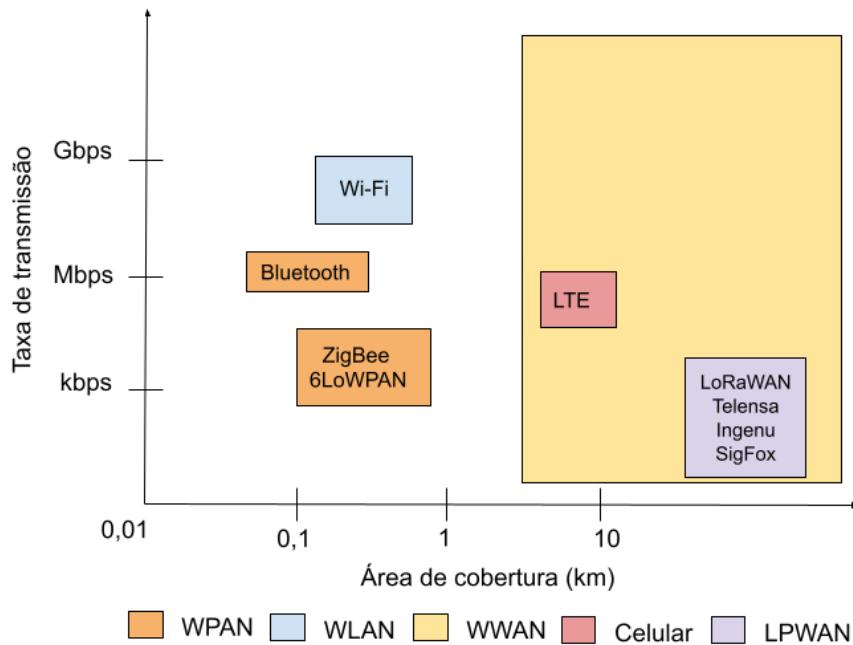
Fonte: Adaptada de (KHOMP, 2020).

2.3.2 Comunicação sem fio

A comunicação realizada entre os sensores da rede (também denominados *endpoints*) com o *gateway* em sua grande maioria ocorre utilizando algum tipo de comunicação sem fio. Porém, não é toda tecnologia de transmissão que se adequa à aplicação, fazendo-se necessário o estudo e a seleção correta da que melhor irá cumprir os requisitos do projeto. Para um sistema funcionar com o melhor desempenho e com o maior grau de autonomia possível, o modelo de comunicação deve proporcionar aos equipamentos um **baixo consumo energético** e uma **longa área de cobertura**, sendo que em alguns cenários as distâncias entre *endpoints* e *gateways* podem alcançar regiões entre 10 km a 40 km em ambientes mais abertos, e de 1 km a 5 km em ambientes urbanos (CHAUDHARI; ZENNARO, 2020).

A Figura 5 demonstra uma comparação de Área de Cobertura *versus* Taxa de Transmissão para algumas das opções mais consolidadas do mercado. É importante destacar o grupo denominado de *Low Power Area Wide Network* (LPWAN), que devido ao valor elevado da área de cobertura do sinal é bastante atrativo para soluções IoT. Por outro lado, também é possível perceber que a taxa de transmissão desse grupo é baixa, porém isso não é um problema visto que não há um fluxo alto de dados nos sistemas. A tecnologia LTE apesar de abranger distâncias consideráveis, sua alta taxa de transmissão faz com que os dispositivos consumam muita energia, tornando assim uma escolha não muito viável.

Figura 5 – Comparaçao entre tecnologias de transmissão sem fio.



Fonte: O próprio autor.

2.3.3 Envio dos dados para uma API

Com os dados já no *gateway*, o mesmo realiza a transmissão através de uma rede *Internet* para uma *Application Programming Interface* (API), que é um conjunto de regras para uma comunicação confiável e íntegra entre produtos e serviços (IBM, 2020c). O princípio de funcionamento é o seguinte: o lado que está requisitando um serviço, denominado de cliente, faz um pedido ao servidor o qual está responsável pela implementação da API. O servidor por sua vez retorna uma mensagem ao cliente informando se a operação foi possível ou se ocorreu algum erro. Esses pedidos realizados à API são denominados de métodos, e são implementados através do protocolo *Hypertext Transfer Protocol* (HTTP). Alguns dos principais métodos estão apresentados abaixo (MOZILLA, 2021):

- GET: Utilizado para solicitar alguma informação por parte do cliente;
- POST: Indica o envio de algum dado para o servidor;
- PUT: Alteração de alguma informação armazenada no lado do servidor pelo parâmetro passado pela requisição;
- DELETE: Pedido de exclusão de dados no servidor.

Os dados enviados pelas requisições HTTP geralmente seguem uma formatação denominada JSON². Um exemplo desse tipo de formato pode ser visualizado no Código 2.1, sendo que é possível criar classes de objetos como *pessoa* e *animal* e inserir ocorrências desses objetos com suas respectivas características.

Código 2.1 – Exemplo de formatação JSON

```

1 {
2     "pessoa": [
3         {
4             "nome": "Maria",
5             "idade": 30,
6             "altura": 1.6
7         },
8         {
9             "nome": "João",
10            "idade": 25,
11            "altura": 1.75
12        }
13    ],
14    "animal": [
15        {
16            "nome": "Leão",
17            "peso": 190
18        },
19        {
20            "nome": "onça",
21            "idade": 90
22        }
23    ]
24 }
```

2.3.4 Armazenamento dos dados

Em aplicações de *Internet das Coisas* há uma quantidade massiva de dados, o que força a utilização de ferramentas adequadas para o armazenamento correto e seguro das informações coletadas. A tecnologia de Banco de Dados (BD) foi desenvolvida visando uma organização estrutural e de acesso eficiente, proporcionando aos sistemas maior desempenho durante a procura dos dados.

² Site oficial JSON

Existem dois formatos de construção de um BD, sendo o Não-Relacional e o Relacional. Em um Banco de Dados Relacional os dados podem ser classificados em diferentes grupos, denominados de Entidade. Cada Entidade possui uma tabela de armazenamento com colunas representando suas características (atributos) e linhas que representam uma ocorrência dessa Entidade, sendo assim mais rígido quanto a modificações em seu esquema.

As tabelas realizam Relacionamentos com outras tabelas associando uma ocorrência com outra, sendo que as relações possuem cardinalidades, ou seja, qual o número mínimo e máximo de associações que uma ocorrência de uma determinada tabela possui com uma ocorrência de outra. Existem casos em que é extremamente recomendado criar uma nova tabela para representar o relacionamento entre duas Entidades (HEUSER, 2009).

A Figura 6 apresenta um exemplo de banco relacional com duas tabelas (entidades). A tabela **Proprietário** irá armazenar dados de pessoas, e possui como atributos os campos *cpf*, *nome* e *idade*. As nomenclaturas *VARCHAR(11)*, *VARCHAR(45)*, e *int* especificam apenas o tipo de dado armazenado, sendo um texto de onze caracteres, texto de quarenta e cinco caracteres e um inteiro, respectivamente. Outro ponto importante de análise é que o atributo *cpf* é único para ocorrência da tabela **Proprietário**, o qual recebe a nomenclatura de *Chave Primária* da entidade.

Outra tabela existente nesse exemplo de banco de dados é a denominada **Carro**, a qual possui os atributos *idCarro* (chave primária), *ano* e *modelo*. As duas entidades possuem um relacionamento com cardinalidade um-para-muitos, ou seja, um Proprietário pode possuir vários Carros, mas um Carro possui apenas um único Proprietário. Devido a essa relação, a tabela Carro ganha o atributo *cpfProprietario*, o qual pode ser denominado de *Chave Estrangeira* e faz referência ao *cpf* do Proprietário. Dessa forma, caso se queira saber quais Carros pertencem a um determinado Proprietário, basta procurar quais ocorrências da tabela Carro possuem o atributo *cpfProprietario* igual ao atributo *cpf* do Proprietário em questão.

Figura 6 – Exemplo de um Banco de Dados Relacional.



Fonte: O próprio autor utilizando a ferramenta WorkBench.

Por outro lado, o banco de dados Não-relacional possui um esquema de armazenamento flexível, ou seja, cada ocorrência de dado pode apresentar campos diferentes uns dos outros, e também apresenta um ótimo desempenho na consulta dos dados armazenados, sendo ideal para aplicações que possuem uma massiva captura de dados (MONGODB, 2021). Existem alguns tipos de bancos para esta classificação, sendo eles: **Chave-valor**, **Documentos** e **Grafos**.

Nesta pesquisa irá ocorrer o uso do tipo **Documentos**, o qual é formado por coleções e documentos em formato JSON que armazenam valores no estilo chave-valor, sendo a chave um identificador único do documento (TIWARI, 2011). Ou seja, no projeto deste BD haverá um *database*, o qual possuirá *coleções* (as coleções podem ser entendidas como as Entidades em um banco relacional), as quais irão armazenar *documentos* (ocorrências de dados) com identificadores únicos.

Na pesquisa irá se utilizar os dois tipos de Banco de Dados (BD) apresentados, porém para funções distintas. O banco relacional irá ser responsável por armazenar dados do sistema referentes a usuários, equipamentos, endereços e suas associações. Já o não-relacional será utilizado para o armazenamento dos dados capturados pelos dispositivos do sistema, como temperatura, umidade e pressão. Os bancos escolhidos foram MySQL e MongoDB devido a suas grandes consolidações no mercado.

2.3.5 Computação em Nuvem

Essa tecnologia permite que toda a parte de *software* e de armazenamento de dados fiquem hospedados em grandes servidores de empresas que fornecem esse tipo de serviço. Dessa maneira, o profissional que deseja desenvolver um sistema completo com API, Banco de Dados e processamento de informações terá suas tarefas simplificadas e seus custos reduzidos, pois o mesmo não necessita possuir de forma física computadores que realizam e disponibilizam seu produto para todos na *internet*.

2.3.6 Aplicação web e móvel

Por fim, os resultados e processamentos realizados com os dados coletados são apresentados ao usuário de forma clara e objetiva, sendo possível desenvolver uma plataforma *mobile* através de aplicativos ou de forma *web*, criando um *site*.

2.4 Inteligência Artificial

Diversas tarefas e atividades cotidianas como por exemplo assistir a um filme totalmente novo, o qual foi recomendado pela própria plataforma de entretenimento com base em escolhas anteriores, utilizar um aplicativo de navegação e encontrar o melhor caminho para chegar ao destino ou até mesmo encontrar alguma pessoa do passado em uma rede social sem sequer lembrar o nome dela, só é possível graças às tecnologias do ramo da Inteligência Artificial (IA).

De acordo com (DUAN; EDWARDS; DWIVEDI, 2019), IA é a capacidade de máquinas aprenderem a realizar tarefas de forma mais próxima à capacidade humana, analisando dados e encontrando padrões através de algoritmos que são utilizados nas mais diversas aplicações e áreas de pesquisa.

Um bom caso já citado para entender melhor uma das soluções de IA é a recomendação de filmes ou músicas por serviços de mídia, quando o usuário consome certo tipo de conteúdo como por exemplo filmes de Ação e músicas de *Rock*, através desses dados coletados o algoritmo irá selecionar as melhores opções do catálogo com base no gênero dos filmes, nos atores, na banda e entre outros parâmetros. Quanto mais precisa for essa recomendação, maior será a satisfação do cliente e por consequência a empresa terá maiores lucros e boa reputação.

Com as aplicações de IA ganhando destaque nas últimas décadas, muitas pessoas acreditam que isso é um conceito novo ou até mesmo futurista. Porém, por volta de 1936 o matemático Alan Turing, conhecido como o “pai da IA”, já realizava estudos na área de ciências computacionais, sendo que a publicação do seu artigo “Computing Machinery and Intelligence” no ano de 1950 considerado o marco para o estudo da Inteligência Artificial (TAULLI, 2019).

A IA pode ser dividida em duas vertentes, sendo a Inteligência Artificial Genérica, do inglês *Artificial General Intelligence* (AGI), e a Inteligência Artificial Estreita, do inglês *Artificial Narrow Intelligence* (ANI). A AGI não é algo acessível ainda, poucas empresas desenvolvem estudos nessa área, a qual visa em realmente dar consciência às máquinas, além de sentimentos e emoções. Por outro lado, a ANI são tecnologias que já estão sendo consumidas pelo mercado como por exemplo assistentes virtuais, carros autônomos e reconhecimento de imagens, os quais utilizam técnicas de algoritmos desenvolvidos para essa finalidade (TAULLI, 2019).

2.4.1 Aprendizado de Máquina

Um dos ramos mais importantes da ANI é o Aprendizado de Máquina, do inglês ML, a qual possui conceitos importantes de funcionamentos. De acordo com Lantz (2015) a aprendizagem **Supervisionada** também pode ser classificada como um modelo de predição, e possui como características os dados de entrada e saída conhecidos, e seus algoritmos trabalham para relacionar esses dois conjuntos de informações. Esse grupo pode ainda ser dividido em técnicas de **Ressão** que é utilizado para prever dados numéricos, e em técnicas de **Classificação** para prever categorias. Um exemplo de aplicação do aprendizado supervisionado é detecção de *e-mails* do tipo *spam*, sendo que a partir dos *e-mails* do usuário já existente e de características que um *e-mail spam* possui, é possível classificá-los e consequentemente descartá-los.

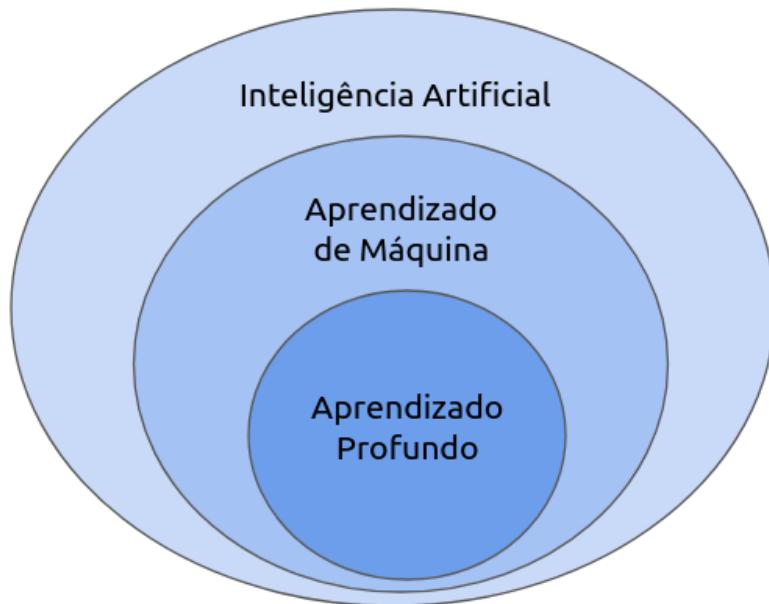
Por outro lado, a aprendizagem **Não Supervisionada** é aplicada quando apenas o conjunto de informação de entrada é conhecido, ocorrendo a classificação dentre esses dados fornecidos. As técnicas utilizadas para essas situações são denominadas de **Agrupamento**, ou *clustering* do inglês. Serviços de recomendação de produtos utilizam em grande escala esses algoritmos, os quais recebem um conjunto de dados de consumidores e realiza o agrupamento dessas informações, a fim entregar as melhores recomendações para os diferentes grupos de usuários identificados (LANTZ, 2015).

Diferente dos grupos anteriores, a Aprendizagem de Máquina por **Reforço** não possui informações de entrada e de saída, a fim de classificá-las ou subdividi-las. Nesse tipo de ML o foco é mapear situações para ações, ou seja, a máquina irá realizar diversas tentativas e estratégias e verificar seus resultados, e dependendo do retorno adquirido, diferentes decisões são tomadas (SUTTON; BARTO, 2020). Um bom exemplo são casos onde máquinas estão em processo de aprendizagem de algum jogo, de acordo com os movimentos realizados por ela e com base nos resultados obtidos, novos caminhos são seguidos para atingir os objetivos definidos.

Dentro do conjunto de algoritmos de Aprendizado de Máquina, existem alguns algoritmos específicos que são classificados de **Aprendizagem Profunda**, do inglês *Deep Learning* (DL), que herdam características gerais do conjunto maior, como por exemplo, podem ser do tipo supervisionados. Nesse grupo há uma alta quantidade de dados e de processamento computacional, capazes de descobrir padrões e comportamentos muito mais complexos e ocultos que outras técnicas de ML (IBM, 2020a). A estrutura principal dos algoritmos de DL é a Redes Neurais Artificiais (RNA), as quais são compostas por camadas de nós, denominados também de neurônio artificial, que formam um rede a fim de imitar a forma como os seres biológicos pensam. As técnicas de RNA são extremamente poderosas, e são utilizadas para aplicações que necessitam de alto desempenho como reconhecimento de imagens, voz e padrões (IBM, 2020b).

Através da Figura 7 é possível visualizar de forma gráfica os conjuntos existentes no conceito de Inteligência Artificial (IA). Como citado, o Aprendizado Profundo está dentro da definição de Aprendizado de Máquina, porém, nele existem algoritmos para atividades específicas, necessitando assim dessa divisão.

Figura 7 – Divisões da Inteligência Artificial.



Fonte: O próprio autor.

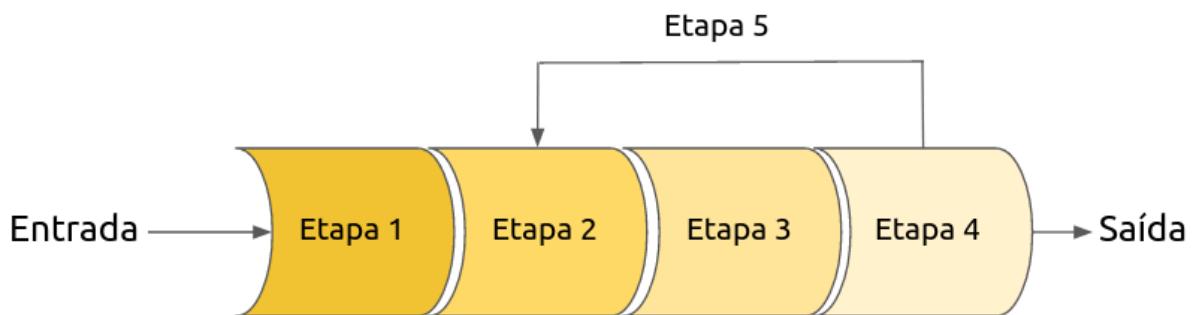
2.4.2 Etapas do Aprendizado de Máquina

O processo de *Machine Learning* (ML) consiste em diversas etapas, dependendo do contexto e da aplicação elas podem sofrer variações. Porém, grande parte das pesquisas e projetos seguem um padrão, o qual pode ser visualizado através da Figura 8. As definições dos estágios de ML são: (LANTZ, 2015):

- **Etapa 1 - Coleta dos dados:** Coleta e armazenamento dos dados que serão estudados, podendo ser em formato de tabelas, arquivos de texto, imagens e entre outros;
- **Etapa 2 - Pré-processamento do dados:** Adequação e normalização dos dados para uma análise mais assertiva;
- **Etapa 3 - Treinamento do modelo:** Escolha do algoritmo ML que melhor atenderá as necessidades, e adaptação ao caso da pesquisa;

- **Etapa 4 - Teste de validação do modelo:** Realizar testes com o algoritmo desenvolvido. É importante avaliar bem o comportamento do modelo, a fim de se evitar que o mesmo seja tendencioso e que gere falsos positivos;
- **Etapa 5 - Melhorias no modelo:** Caso os resultados não sejam satisfatórios, deve-se realizar os processos da etapa 2 em diante novamente, com o objetivo de melhorar as técnicas utilizadas para a criação do modelo.

Figura 8 – Etapas do Aprendizado de Máquina.



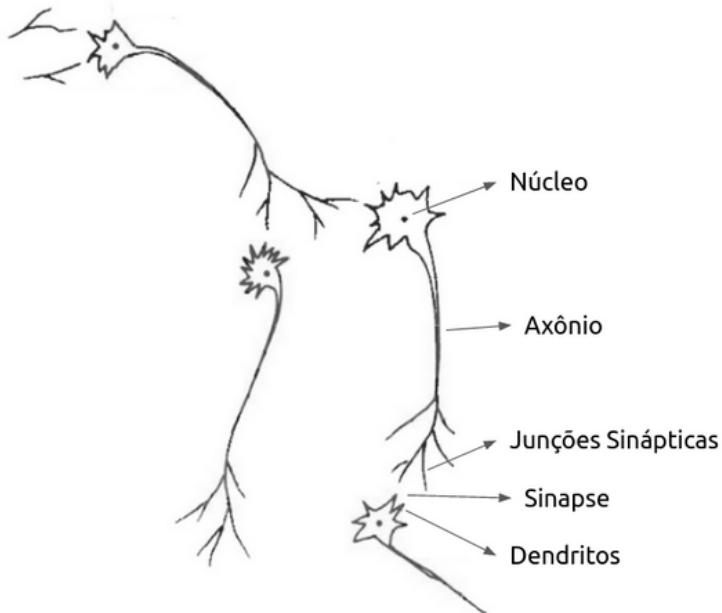
Fonte: O próprio autor.

2.5 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) são redes computacionais com objetivo de reproduzir o modo de análise de dados das células biológicas do sistema nervoso central, os neurônios. O cérebro possui uma estrutura de funcionamento capaz de realizar tarefas complexas de modo quase instantâneo, como o reconhecimento de um rosto em meio a uma multidão, o que para um sistema de computador é algo muito complicado e que exige grande capacidade de processamento. Uma das principais características do cérebro é a capacidade de criar suas próprias regras de decisões, sendo que a cada nova entrada de dados (novas experiências) os resultados se tornam mais precisos e confiáveis (HAYKIN, 2008).

A Figura 9 representa a estrutura de um neurônio biológico, o qual forma uma rede com outros trilhões de neurônios todos interligados para a realização das mais variadas tarefas. Suas principais partes são: **Núcleo**, região de processamento das informações da célula. **Axônio**, transmissões dos sinais eletroquímicos através do neurônio até as **Junções Sinápticas**, as quais realizam as *sinapses* com os **dendritos**. Um neurônio pode realizar centenas de sinapses assim como se conectar com diversos outros neurônios e receber informações desses. Um ponto importante a destacar é que as conexões entre os neurônios possuem prioridades (pesos), onde cada uma tem sua própria relevância quanto à informação a ser processada (GRAUPE, 2013).

Figura 9 – Estrutura biológica de um neurônio.



Fonte: Adaptada de (GRAUPE, 2013)

Dessa forma, a ciência computacional modelou um neurônio artificial baseado na estrutura biológica, mas de acordo com Haykin (2008), é importante ressaltar que o nível de complexidade da célula biológica está distante de ser replicada com alto grau de fidelidade, porém é extremamente satisfatório o progresso no estudo das RNA assim como os ótimos resultados obtidos com essa analogia desenvolvida da célula animal. A subseção 2.5.1 detalha a célula artificial denominada de *Perceptron*.

2.5.1 Perceptron

Como já previamente discutido, o neurônio biológico foi mapeado matematicamente a fim de gerar uma célula artificial, esta qual denominada de *Perceptron*. O *Perceptron* teve seu surgimento em 1958 através do artigo publicado por Frank Rosenblatt, sendo este uma evolução do estudo inicial desenvolvido por McCullough and Pitts ainda na década de 1940.

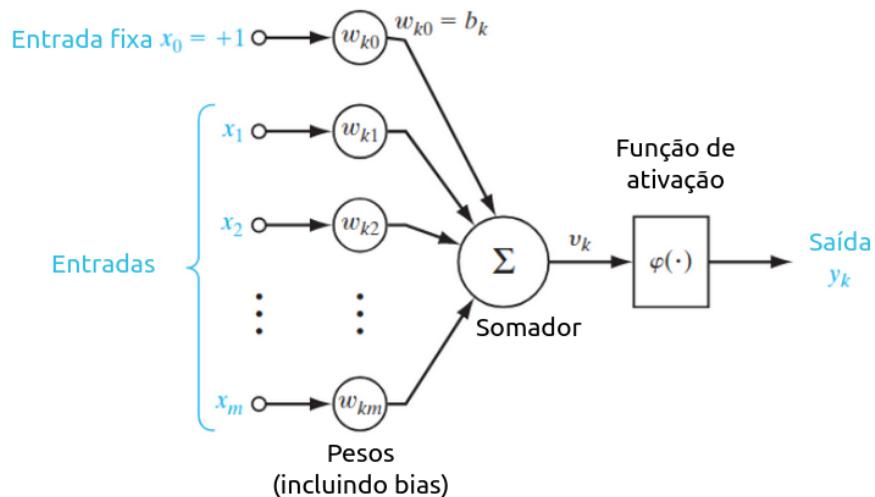
Segundo Rosenblatt (1958), no início de sua publicação, para se conseguir entender como um organismo vivo consegue reconhecer e interpretar os fenômenos ao seu redor, é necessário responder as seguintes perguntas: *Como a informação é detectada?; De qual forma ela é armazenada?; A informação já armazenada tem influência no comportamento desses organismos?*

A partir destes questionamentos, o cientista desenvolveu o modelo matemático para representar o comportamento de algumas atividades do cérebro animal, tais como reconhecimento de padrões, classificação e memória.

A Figura 10 demonstra um *Perceptron* adaptada da célula artificial apresentada por Haykin (2008), a qual é composta pelos seguintes elementos:

- **Sinais de entrada** denominados por x_j que se conectam com seus respectivos **pesos** representados por w_{kj} , sendo que cada sinal terá seu peso associado que poderá se diferenciar dos demais sinais. Ou seja, um sinal pode ser mais relevante que o outro, em outras palavras, terá um peso maior;
- **Somador** é responsável por combinar as entradas ponderadas;
- **Função de Ativação** restringe a amplitude de saída de um neurônio, geralmente limitando nos intervalos fechados $[0,1]$ ou $[-1,1]$;
- **Bias** é um parâmetro externo que tem por objetivo aumentar ou diminuir a entrada líquida da função de ativação, dando maior flexibilidade ao modelo projetado com o neurônio.

Figura 10 – Modelo não-linear de um neurônio artificial.



Fonte: Adaptada de (HAYKIN, 2009)

Matematicamente o neurônio artificial pode ser expressado pelas equações 2.1 e 2.2:

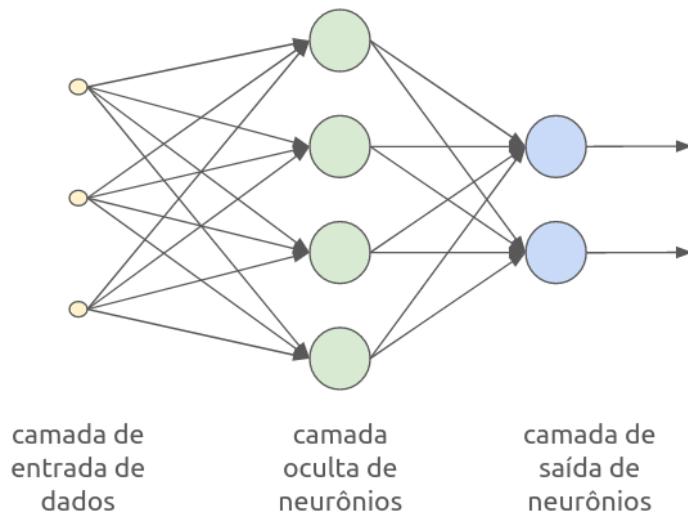
$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (2.1)$$

$$y_k = \varphi(v_k) \quad (2.2)$$

2.6 Redes Neurais Recorrentes

Com o princípio de um neurônio artificial já estabelecido, diversas formas de redes foram desenvolvidas explorando a combinação estrutural dessas células. A Figura 11 demonstra os principais componentes de uma rede neural artificial comumente encontrada, a qual inicia com a **camada de entrada** que representa os dados que serão utilizados como estudo, logo em seguida, se tem a **camada oculta** que é responsável por realizar algumas operações antes de se chegar na última camada, a **camada de saída**, onde são novamente processados e finalizados. Na maioria dos casos, os neurônios de determinada camada recebem seus dados de entradas a partir das camadas precedentes.

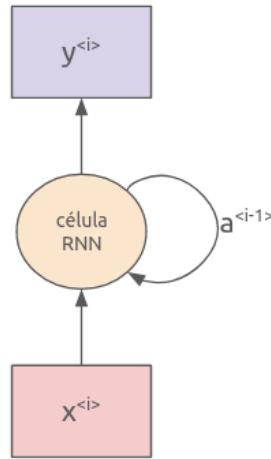
Figura 11 – Exemplo de estrutura de uma Rede Neural Artificial.



Fonte: O próprio autor.

Porém, existe uma estrutura denominada de Rede Neural Recorrente, do inglês *Recurrent Neural Network* (RNN), muito utilizada em tarefas onde a ordem cronológica dos dados é de extrema importância. Segundo a definição de Haykin (2008) uma RNN é caracterizada por possuir laços de realimentação de informação, ou seja, os dados processados por uma camada de neurônios podem ser reinseridos como entrada nesta mesma camada. A Figura 12 demonstra a estrutura básica de uma RNN, sendo possível visualizar uma iteração no processamento dos dados na célula RNN.

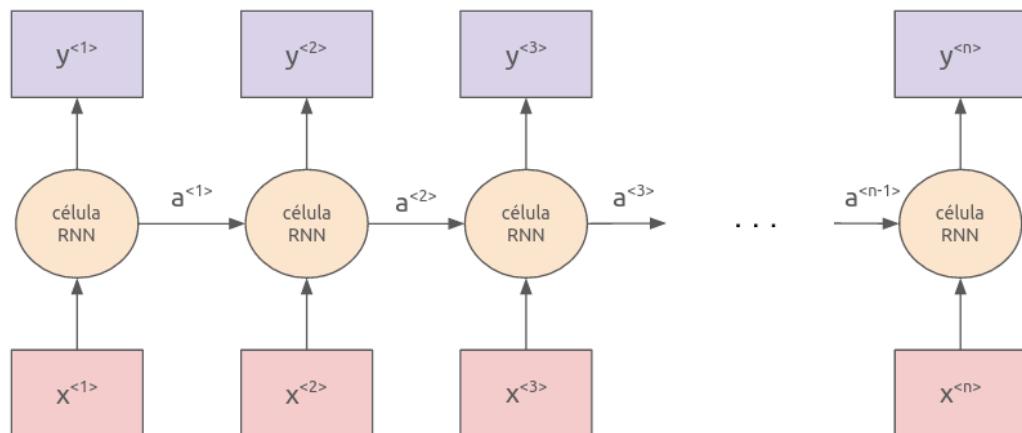
Figura 12 – Exemplo básico de uma rede RNN.



Fonte: O próprio autor.

A Figura 13 demonstra a estrutura básica RNN de forma aberta, sendo que o *loop* existente representa a ligação entre mais de uma célula, formando assim uma sequência delas. O bloco $y^{<i>}$ representa a saída do processamento daquela célula, e o fluxo $a^{<i>}$ é o resultado de uma célula passando para sua rede subsequente. Dessa forma, é possível constar que a rede trabalha com memória, pois o resultado de uma tem influência na próxima.

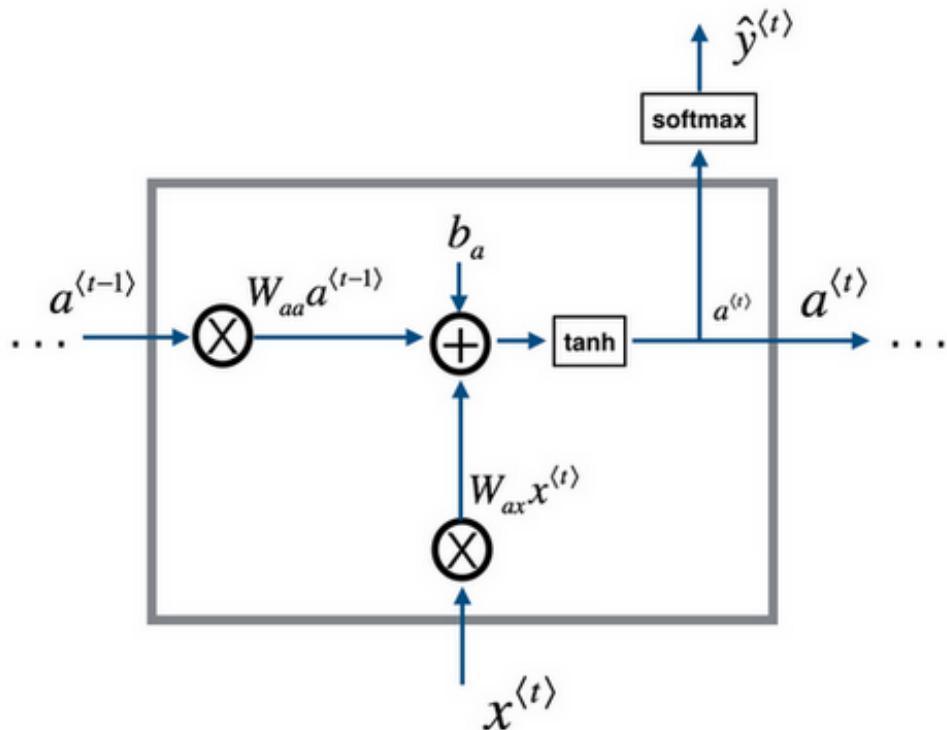
Figura 13 – Exemplo rede RNN aberta.



Fonte: O próprio autor.

A Figura 14 representa uma célula RNN, a qual possui o seguinte funcionamento: A entrada x_t , sendo o dado atual de processamento, é multiplicado pelo seu respectivo peso $W_{ax}x^{<t>}$ e somado com a entrada $a^{<t-1>}$, que representa o valor de saída da célula anterior também multiplicada com o seu peso $W_{aa}a^{<t-1>}$, e com um viés, denominado de b_a . Esse valor é aplicado em uma função de ativação \tanh , e direcionado a uma saída $a^{<t>}$ que será utilizada na próxima célula, e a outra $\hat{y}^{<t>}$ através de um função softmax , a qual representa o resultado de saída daquela célula (YANG, 2018).

Figura 14 – Célula RNN.



Fonte: Adaptada de (YANG, 2018).

As Equações 2.3 e 2.4 representam matematicamente este processo interno de uma célula RNN.

$$a^{<t>} = \tanh(W_{ax}x^{<t>} + W_{aa}a^{<t-1>} + b_a) \quad (2.3)$$

$$\hat{y}^{<t>} = \text{softmax}(W_{ya}a^{<t>} + b_y) \quad (2.4)$$

2.7 Long-Short Term Memory

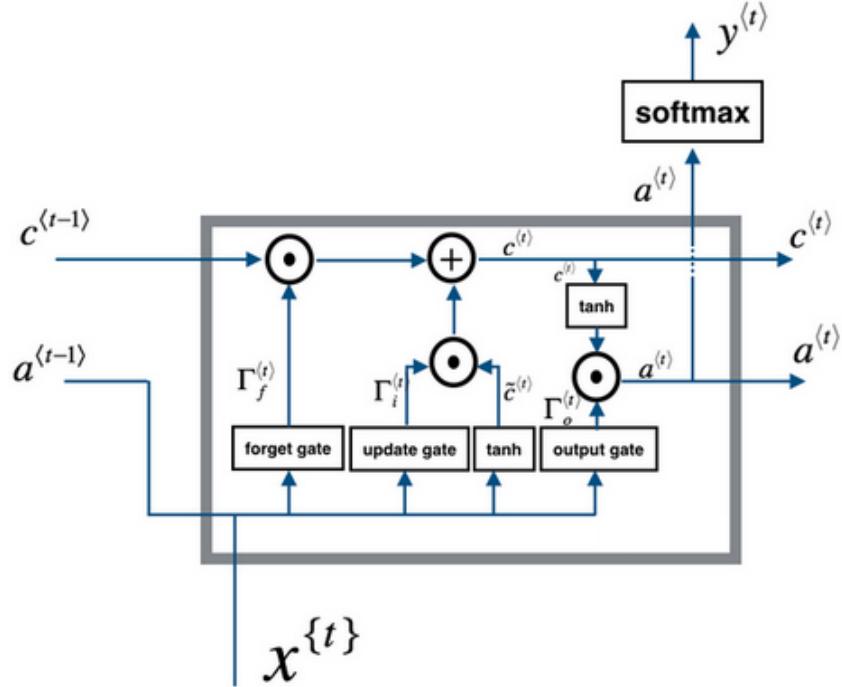
As RNN têm a importante característica de influenciar o resultado de uma saída com base no resultado da célula anterior, conforme discutido na seção 2.6. Porém, para longas sequências de dados, essa influência diminui gradativamente fazendo com que as primeiras amostras não interfiram nas últimas. Em alguns casos, esse comportamento não é interessante pois todos os eventos, desde o início até o fim do processamento, de alguma maneira podem possuir relações em seus comportamentos.

Com isso, as redes *Long-Short Term Memory* (LSTM) foram desenvolvidas, as quais, de acordo com Hennayake, DInalankara e Mudunkotuwa (2021), é uma variação de *Recurrent Neural Network* (RNN) que possui a capacidade de armazenar informações dos dados durante um longo período de tempo, e utilizá-las para o processamento de dados atuais, tornando assim uma ótima solução para um conjunto de informação com uma grande quantidade de amostras ao longo do tempo.

A Figura 15 representa uma célula LSTM, a qual é composta por importantes partes que realizam toda a sua lógica de memória. O seu modo de funcionamento é o seguinte:

1. A entrada atual $x^{\{t\}}$ juntamente com o resultado da célula anterior $a^{<t-1>}$ são concatenados e inseridos no **forget gate**, onde são multiplicados por pesos W_f a fim de selecionar quais informações devem possuir maior influência nos próximos resultados. O vetor $\Gamma_f^{(t)}$ resultante desta operação é multiplicado por $c^{<t-1>}$, denominado de estado da célula anterior, elemento por elemento. Quando o resultado de um desses elementos é igual a zero, o mesmo é descartado. Ou seja, esse elemento será esquecido, do inglês *forget*, do processamento;
2. Em paralelo, as entradas $x^{\{t\}}$ e $a^{<t-1>}$ passam pelo **update gate** sendo multiplicadas pelos pesos W_u a fim de atualizarem, do inglês *update*, a célula com a influência que este novo dado poderá gerar para a sua sucessora. O vetor $\Gamma_u^{(t)}$ é multiplicado por $\tilde{c}^{(t)}$ e \tanh , correspondente ao estado intermediário atual da célula, e posteriormente somado com o resultado do item 1. O vetor resultado desta etapa é lançado para a saída $c^{(t)}$ e também utilizado na próxima fase;
3. Por fim, para gerar os dados de saídas, as duas entradas são concatenadas e multiplicadas por pesos W_o no **output gate**, e o seu vetor $\Gamma_o^{(t)}$ é multiplicado com o resultado do item 2 aplicado a uma $\tanh()$. No final desse processo há o resultado final da célula, que é repassada para a próxima através de $a^{<t>}$, e fornecida para fora da rede como $y^{<t>}$ aplicada em uma função softmax.

Figura 15 – Célula LSTM.



Fonte: Adaptada de (YANG, 2018).

As Equações 2.5 a 2.11 representam matematicamente o processo interno de uma célula LSTM (YANG, 2018).

$$\Gamma_f^{<t>} = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f) \quad (2.5)$$

$$\Gamma_u^{<t>} = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u) \quad (2.6)$$

$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c) \quad (2.7)$$

$$c^{<t>} = \Gamma_f^{<t>} \circ c^{<t-1>} + \Gamma_u^{<t>} \circ \tilde{c}^{<t>} \quad (2.8)$$

$$\Gamma_o^{<t>} = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o) \quad (2.9)$$

$$a^{<t>} = \Gamma_o^{<t>} \circ \tanh(c^{<t>}) \quad (2.10)$$

$$y^{<t>} = \text{softmax}(a^{<t>}) \quad (2.11)$$

2.8 Séries Temporais

De acordo com Lazzeri (2020) uma Série Temporal (ST) pode ser definida como um tipo de dado que possui informações de determinado evento ao longo do tempo, tornando possível a análise do seu comportamento. Uma linha de pesquisa muito importante envolvendo ST é a Previsão de Séries Temporais, sendo viável aplicar simples métodos estatísticos como médias ou correlações, até algoritmos complexos de *Deep Learning* (DL) para prever situações de acordo com a série histórica dos dados armazenados (LAZZERI, 2020).

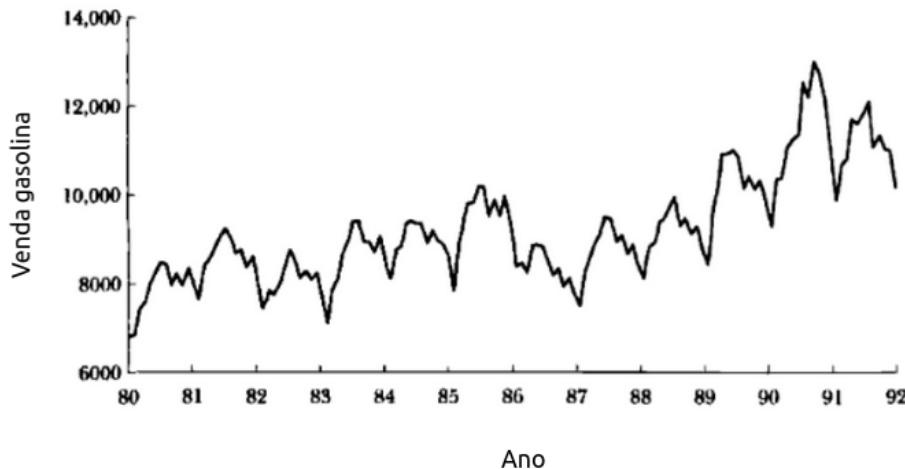
Diversas áreas de conhecimento utilizam Séries Temporais para prever ou estudar as mais variadas situações, como por exemplo na **climatologia** na previsão de parâmetros climáticos, na **economia** para análise de indicadores importantes do mercado, e também na **astronomia** a fim de calcular o movimento de equipamentos e corpos celestes (NIELSEN, 2019).

2.8.1 Tipos de variações em Séries Temporais

O profissional que irá lidar com Séries Temporais deve estar ciente de todas as características que seu conjunto de dados possui, dentre elas, conseguir identificar o tipo de comportamento que a curva apresenta. Uma ST pode ser classificada nas seguintes variações:

- Tendência (*Trend*): É o movimento de aumento ou diminuição dos valores em determinado período de uma série temporal;
- Sazonal (*Seasonal*): Padrão de comportamento dos dados, geralmente está relacionado com a época de observação. A Figura 16 representa a venda de gasolina entre os anos de 1980 e 1992, sendo possível observar durante as férias de verão (meio do ano) de países da América do Norte o aumento do consumo do combustível. Essa tipo de variação ocorre em intervalos de tempo inferiores ou próximos de um ano (DIEBOLD, 2006);
- Cíclico (*Cyclic*): Períodos que se repetem, porém, diferente da sazonalidade não há um intervalo bem definido para esta repetição, podendo ser geralmente superior a um ano. Um exemplo seria o mercado de ações, é normal uma ação ganhar ou perder valor ao longo do tempo, mas essa variação não acontece em períodos fixos de tempo;
- Aleatório (*Random*): Também denominado de ruído, são variações inesperadas na série temporal.

Figura 16 – Exemplo de sazonalidade.



Fonte: Adaptada de (DIEBOLD, 2006).

2.8.2 Estrutura dos dados para Séries Temporais

A estrutura básica de organização para se trabalhar com Séries Temporais é feita através de tabelas, sendo que as colunas representam os parâmetros, e cada observação é armazenada em linhas. A primeira coluna (geralmente denominada de index) informa a data em que determinada medição foi realizada, por isso a ST é extremamente utilizada em análise de dados, pois é possível realizar operações de buscas e filtragens através de períodos de tempo utilizando a coluna de index.

Um *dataset*, termo em inglês utilizado para *conjunto de dados*, pode ser visualizado na Figura 17. A primeira coluna possui a data das observações e as outras representam os parâmetros de análise, sendo este um exemplo fictício de um estudo climático.

Figura 17 – Exemplo de dataset.

Data	Temperatura	Umidade
01-01-2021	25	60
01-02-2021	21	75
01-03-2021	23	81

Fonte: O próprio autor.

2.9 Cálculo de erros

Nesta Seção são discutidas as técnicas para os cálculos dos erros utilizados para a avaliação do processo de previsão que serão demonstrados na Seção 3.7. As subseções a seguir irão detalhar como são realizados os cálculos para cada uma das medições.

2.9.1 Mean Absolute Error

Mean Absolute Error (MAE) é uma métrica de cálculo simples que representa qual o valor médio do erro entre os dados previstos e os dados reais. Assim quanto menor este valor, melhor terá sido a previsão realizada. O seu cálculo é definido pela Equação 2.12, sendo y_i o valor real, \hat{y}_i o valor previsto e n o número de amostras analisadas.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.12)$$

2.9.2 Root Mean Square Error

O cálculo denominado de *Root Mean Square Error* (RMSE) tem por objetivo ser utilizado como um índice de avaliação, o qual possui maior sensibilidade a casos com grande discrepância de valor, ou seja, considera um peso maior a dados *outliers* (CHEN; GUAN; LI, 2021). A fórmula utilizada para esta métrica está demonstrada na Equação 2.13, a qual utiliza os mesmos parâmetros apresentados na métrica anterior.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (2.13)$$

2.9.3 Mean Absolute Percentage Error

Mean Absolute Percentage Error (MAPE) realiza o cálculo do erro médio absoluto entre os dados previstos e reais, porém apresenta o resultado em forma de porcentagem (YU; CAO; ZHU, 2019), conforme demonstrada na Equação 2.14. Esta métrica é usada comumente para medir a acurácia da previsão realizada (ORACLE, 2022).

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \quad (2.14)$$

3 DESENVOLVIMENTO

Neste capítulo será detalhado todo o processo de estudo e desenvolvimento de um sistema para previsão de parâmetros climáticos, visando proporcionar ao usuário o máximo de confiabilidade possível e tornando suas tarefas mais precisas, diminuindo assim os seus custos.

A pesquisa baseou-se em dados históricos de longa data, com aproximadamente 10 anos de informações capturadas de hora em hora por estações meteorológicas pertencentes ao Instituto Nacional de Meteorologia (INMET). Com o intuito de demonstrar todas as etapas de um projeto de *Machine Learning* (ML), o capítulo discutirá todos os tópicos vistos na Sub-subseção 2.4.2, apresentando a base de dados utilizada, o algoritmo escolhido, os testes, validações e resultados obtidos.

Também será apresentado a página web desenvolvida para a interação com o usuário, sendo que o mesmo terá total controle para manipular os dados a serem visualizados através de gráficos.

Com base em estudos e pesquisas das principais tecnologias utilizadas para previsão de variáveis através de séries temporais, foi escolhido o algoritmo *Long-Short Term Memory* (LSTM) devido a sua grande usabilidade e confiabilidade no mundo acadêmico, assim como a biblioteca Keras para o desenvolvimento do projeto.

E para validar essas escolhas e estratégias que serão seguidas, foi realizada uma conversa com um profissional e estudioso da área que integra o time de um órgão de pesquisas climáticas, o EPAGRI/CIRAM. Nessa reunião foi discutida qual a tendência da Inteligência Artificial (IA) neste ramo, e foi constatado que muitos estudos estão sendo desenvolvidos, inclusive com as mesmas tecnologias selecionadas para este trabalho.

3.1 Base de dados

O uso de uma base confiável e completa é essencial para o desenvolvimento de trabalhos que envolvam conceitos de ML. Dessa forma, a base de dados¹ do Instituto Nacional de Meteorologia (INMET) foi escolhida como fonte de informações para a construção do sistema. O Instituto disponibiliza uma plataforma de fácil navegação onde é possível realizar o *download* de arquivos do tipo *csv* com milhares de observações de diversos parâmetros climáticos coletados por estações meteorológicas espalhadas em todo o país.

¹ Dados históricos

Os dados disponíveis mais antigos são do ano de 2000, e logicamente, com o passar do tempo novas estações vão surgindo e sendo incorporadas no projeto do INMET, tornando a base ainda mais rica.

A plataforma disponibiliza uma página gráfica² contendo o mapa brasileiro e as estações em suas respectivas localidades. A partir dela, é possível o usuário selecionar a sua estação de preferência e analisar os dados somente deste equipamento, o que torna a utilização do banco mais fácil e rápida para o usuário.

3.2 Estações meteorológicas selecionadas

O INMET possui centenas de estações meteorológicas espalhadas por todo o território brasileiro, as quais são divididas em **Estações Convencionais**, com dados coletados de forma manual por seus responsáveis, e **Estações Automáticas**, que realizam a captura de informações independente da ação humana. Dessa forma, as do tipo Convencionais não apresentam um histórico muito detalhado, sendo que são poucas as que ainda continuam em operação. Portanto, foi escolhido trabalhar com as Automáticas, devido ao seu grande volume de dados registrados e ao número elevado de equipamentos em atividade.

Cada estação escolhida irá possuir um modelo próprio de previsão de parâmetros, pois, visando a implementação de um sistema em que o usuário, ao cadastrar sua própria estação micrometeorológica, poderá escolher qual estação real (pertencente ao INMET) está mais próxima da sua região, para que assim o modelo de previsão seja o mais semelhante possível a sua localidade.

Em um primeiro momento, foram mapeadas as estações de diferentes regiões geográficas dos estados da região sul brasileira, porém, é de fácil realização ampliar essa seleção para outros estados do país, abrangendo ainda mais a área de monitoramento do sistema. Posto isto, as subseções a seguir detalham as especificações e as localidades de cada equipamento escolhido.

² Mapa estações

3.2.1 Estações meteorológicas do Paraná

A Tabela 1 demonstra as informações básicas das estações meteorológicas automáticas que foram escolhidas para análise no estado do Paraná.

Tabela 1 – Estações Automáticas INMET do Estado do Paraná.

Código	Cidade	Latitude e Longitude	Região
A873	Morretes	[-25.51, -48.81]	Metropolitana
B806	Colombo	[-25.32, -49.16]	Metropolitana
A821	Joaquim Távora	[-23.51, -49.95]	Norte Pioneiro
A823	Inácio Martins	[-25.57, -51.08]	Centro-Sul
A876	Clevelândia	[-26.42, -52.35]	Sudoeste
A875	General Carneiro	[-26.4, -51.35]	Sudeste
A820	Mal. Cândido Rondon	[-24.53, -54.02]	Oeste
B803	Campina da Lagoa	[-24.57, -52.8]	Oeste
A825	Goioerê	[-24.16, -53.03]	Noroeste
A850	Paranapoema	[-22.66, -52.13]	Noroeste

3.2.2 Estações meteorológicas de Santa Catarina

A Tabela 2 contém as estações escolhidas no estado de Santa Catarina.

Tabela 2 – Estações Automáticas INMET do Estado de Santa Catarina.

Código	Cidade	Latitude e Longitude	Região
A806	Florianópolis	[-27.6, -48.62]	Grande Fpolis.
A814	Urussanga	[-28.53, -49.32]	Sul
A862	Rio Negrinho	[-26.25, -49.57]	Norte
A864	Major Vieira	[-26.39, -50.36]	Norte
A895	Chapecó	[-27.09, -52.64]	Oeste
A857	São Miguel do Oeste	[-26.79, -53.51]	Oeste
A865	Lages	[-27.8, -50.34]	Serrana
A898	Campos Novos	[-27.39, -51.22]	Serrana
A870	Rancho Queimado	[-27.68, -49.04]	Serrana
A845	Bom Jardim da Serra	[-28.13, -49.48]	Serrana

3.2.3 Estações meteorológicas do Rio Grande do Sul

Por fim, a Tabela 3 contém as estações escolhidas no estado do Rio Grande do Sul.

Tabela 3 – Estações Automáticas INMET do Estado do Rio Grande do Sul.

Código	Cidade	Latitude e Longitude	Região
A809	Uruguaiana	[-29.84,-57.08]	Sudoeste
A826	Alegrete	[-29.71,-55.53]	Sudoeste
A836	Jaguarão	[-32.53, -53.38]	Sudeste
A811	Canguçu	[-31.4, -52.7]	Sudeste
A884	Campo Bom	[-29.67, -51.06]	Metropolitana
A838	Camaquã	[-30.81, -51.83]	Metropolitana
A837	Soledade	[-28.86, -52.54]	Centro-Leste
A813	Rio Pardo	[-29.87, -52.38]	Centro-Leste
A886	Tupanciretã	[-29.09, -53.83]	Centro-Oeste
A883	Ibirubá	[-28.65, -53.11]	Noroeste
A856	Palmeira das Missões	[-27.92, -53.32]	Noroeste
A828	Erechim	[-27.66, -52.31]	Noroeste
A880	Vacaria	[-28.51, -50.88]	Nordeste
A840	Bento Gonçalves	[-29.16, -51.53]	Nordeste

A Figura 18 demonstra de forma mais clara a distribuição das estações no território da região sul brasileira.

Figura 18 – Estações Automáticas INMET selecionadas para estudo.



Fonte: O próprio autor.

3.3 Descrição da base de dados

Nessa seção é demonstrada de forma detalhada a estrutura dos dados adquiridos para análise na plataforma INMET, a organização dos arquivos utilizados e a formatação da informação para a utilização nos algoritmos de *Machine Learning* (ML) através de um *script*.

3.3.1 Dados Brutos

Como citado na Seção 3.1, o INMET disponibiliza em uma página *web* todos os dados em formato *csv* das estações automáticas desde o ano de 2000. Ao selecionar para *download* os dados de determinado ano, um arquivo compactado será baixado para a sua máquina. Dentro desse arquivo, há os documentos *csv* de cada estação existente naquele ano de monitoramento.

O arquivo bruto é composto por diversas informações, como os dados de identificação da estação, data da coleta e várias colunas que representam cada uma das variáveis capturadas pelo equipamento.

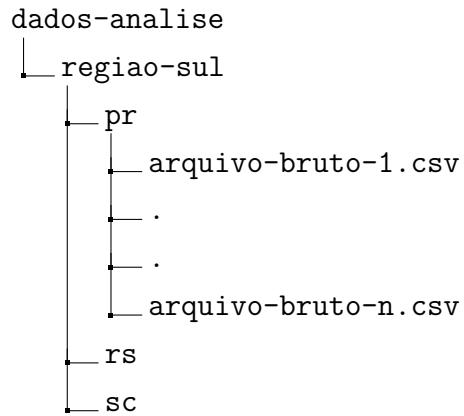
3.3.2 Seleção dos arquivos utilizados

Para o presente estudo foram escolhidos dados entre os anos de 2010 a 2021, pois nesses anos uma boa quantidade de estações meteorológicas estão em operação, e portanto há um ótimo conjunto de dados para serem analisados.

Com os arquivos baixados para cada ano, haverá pastas nomeadas como por exemplo **/2010**. Dentro dessas pastas, há um arquivo *csv* para cada estação naquele respectivo ano, o qual possui a seguinte nomenclatura: **INMET+região+estado+estação+cidade+data início+A+data final**. Por exemplo:

INMET_CO_DF_A001_BRASILIA_07-05-2000_A_31-12-2000

Na Seção 3.2 foram definidas quais estações serão utilizadas no estudo, e para utilizar o *script* de formatação dos dados é necessário realizar a seguinte organização de diretórios:



Sendo que futuramente cada região geográfica brasileira possuirá um diretório contendo outros para cada um de seus estados. Dentro dos estados haverá os arquivos *csv* brutos das estações selecionadas.

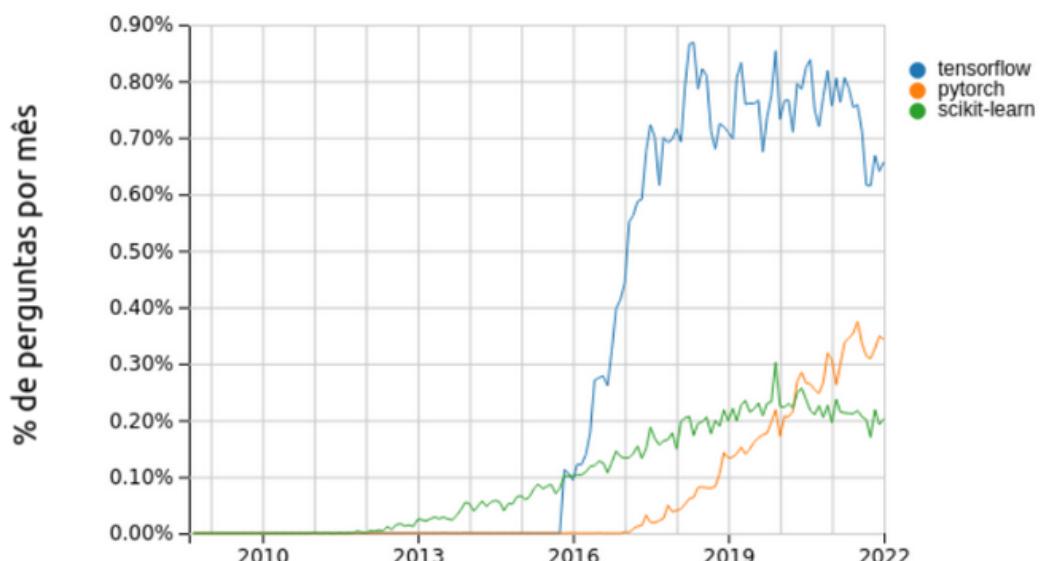
Com essa estrutura pronta, já é possível utilizar o *script* desenvolvido para a junção e formatação dos dados, o qual está disponível [aqui](#).

3.4 Biblioteca Tensorflow

Após estudos iniciais sobre quais tecnologias e ferramentas a serem escolhidas para o desenvolvimento da pesquisa foi decidido utilizar a linguagem de programação **Python**³ devido a sua grande facilidade em trabalhar com ciência de dados. Com isso, foi necessário determinar qual biblioteca de *Machine Learning* (ML) seria a mais apropriada, e após alguns estudos foi selecionada o **TensorFlow**⁴.

O TensorFlow é uma biblioteca que foi desenvolvida para atividades relacionadas a processamento de dados e Inteligência Artificial (IA), a qual implementa diversos algoritmos para a criação e treinamento das mais variadas Redes Neurais Artificiais (RNA). Possui uma ótima integração com a linguagem escolhida e um grande grupo de usuários que estão sempre melhorando e disponibilizando novas funcionalidades. Através da Figura 19 é possível visualizar a sua popularidade comparada à outras bibliotecas de ML, o gráfico apresenta a porcentagem de pesquisas mensal de cada uma delas em toda a plataforma Stack OverFlow⁵, a qual é referência em assuntos voltados à tecnologia e desenvolvimento de *software*.

Figura 19 – Bibliotecas ML mais utilizadas.



Fonte: Adaptada de (STACKOVERFLOW, 2022)

³ Site oficial Python

⁴ Site oficial TensorFlow

⁵ Site oficial Stack OverFlow

Durante o projeto também é utilizado a biblioteca Keras⁶, a qual implementa diversos algoritmos baseados no TensorFlow, simplificando em alguns casos a utilização e permitindo ao cientista de dados investir o seu tempo de desenvolvimento em atividades importantes como a limpeza e criação de estratégias para o seu problema.

Com a biblioteca Keras é possível criar redes neurais artificiais utilizando células *Long-Short Term Memory* (LSTM), alterar seus parâmetros e estrutura, a fim de se obter o melhor resultado. O trecho de código 3.1 demonstra os parâmetros e os métodos utilizados para criar e treinar os modelos LSTM utilizados neste trabalho, sendo estes:

- **units**: Número referente a quantidade células LSTM na rede;
- **input_shape**: Formato dos dados de entrada;
- **activation**: Função de ativação para processos do estado da célula;
- **recurrent_activation**: Função de ativação para os *gates input, forget e output*
- **Dense()**: Define a conexão entre as células LSTM;
- **model.compile()**: Realiza a compilação (montagem) do modelo;
- **model.fit()**: Realiza o treinamento do modelo com os dados de treino, número de épocas e *batch size*.

Código 3.1 – LSTM utilizando Keras

```

1 model.add(LSTM(units=64, input_shape=(24,1),activation="relu",recurrent_activation=
    "sigmoid"))
2 model.add(Dense(1))
3 model.compile(loss='mean_squared_error', optimizer='adam')
4 historico = model.fit(train_X, train_y, epochs = 3, batch_size = 1, verbose = 1)

```

Todas as definições e valores de parâmetros foram retirados da documentação da biblioteca Keras⁷. Também foram realizados alguns testes para verificar quais os melhores parâmetros a serem utilizados, e dentre eles os apresentados acima foram os que obtiveram melhores resultados. Não se viu por necessário detalhar neste trabalho a comparação entre os parâmetros, devido ao foco estar em utilizar o algoritmo na previsão de variáveis climáticas e, e ao encontrar essa combinação optou-se por utilizá-la.

⁶ Site oficial Keras

⁷ Documentação Keras

3.5 Limpeza e formatação dos dados

Antes de iniciar de fato o uso de algoritmos de *Machine Learning* (ML) nos dados de estudo, é necessário entender o conjunto de informações selecionado. Para isto foram realizados alguns ajustes e alterações nos dados brutos para os mesmos se adequarem ao formato ideal da pesquisa.

O primeiro passo foi excluir algumas colunas de informações vindas da base de dados do INMET que não serão utilizadas, tais como localidade da estação, variáveis que não serão objetos de estudo e outras características do equipamento coletor.

Por se tratar de um aparelho eletrônico de medição, a estação meteorológica pode sofrer problemas e não realizar a captura dos dados corretamente. Em alguns períodos de monitoramento, os dados armazenados apresentaram valores incorretos como por exemplo -9999 ou registros onde todas as variáveis foram gravadas como 0. Desta forma, foi necessário realizar uma varredura que identificasse essas anomalias e as excluíssem do *dataset*, para assim não influenciar no treinamento do modelo de ML.

Os nomes das colunas também não estavam em um padrão de fácil manipulação, muitos deles apresentavam caracteres especiais e formatos que causariam problemas durante o desenvolvimento do código. A partir dessa questão, foram todas renomeadas para nomenclaturas de fácil entendimento e que se enquadram em um padrão de escrita universal.

Muitos valores numéricos possuíam suas partes decimais separadas por **vírgula** o que não segue o padrão de linguagens de programação. Com isso foi realizada a troca para o caractere delimitador **ponto** com o intuito de excluir esta operação do código principal, tornando-o mais limpo e objetivo.

Por fim, o ajuste mais complexo realizado foi em relação ao campo de data e hora. Esta pesquisa tem como objetivo trabalhar com series temporais, e como descrito na Seção 2.8 essa maneira de armazenamento requer um campo que identifique uma ocorrência de dado registrado através de um campo no formato de data.

Nesse caso os dados brutos vindos do INMET possuíam duas colunas deste tipo, uma para representar o dia, mês e ano e outra para os horários. Dessa forma, foi feito um agrupamento entre esses dois campos a fim de gerar apenas um, no formato ano-mês-dia hora:00:00. O *script* de formatação dos dados pode ser acessado [aqui](#).

3.6 Análise comportamental dos dados

Na Seção 3.2 foram escolhidas diversas estações meteorológicas dos três estados da região sul do Brasil, a fim de se obter uma maior abrangência e confiabilidade na aplicação dos modelos de acordo com a localidade do cliente.

Nesta Seção são escolhidas três estações (uma de cada estado) com um grande volume de dados armazenados para traçar graficamente a similaridade do comportamento das variáveis climáticas nos locais, para assim se poder trabalhar com apenas uma dessas estações, pois, será sabido que o mesmo poderá ser aplicado a todas as outras estações.

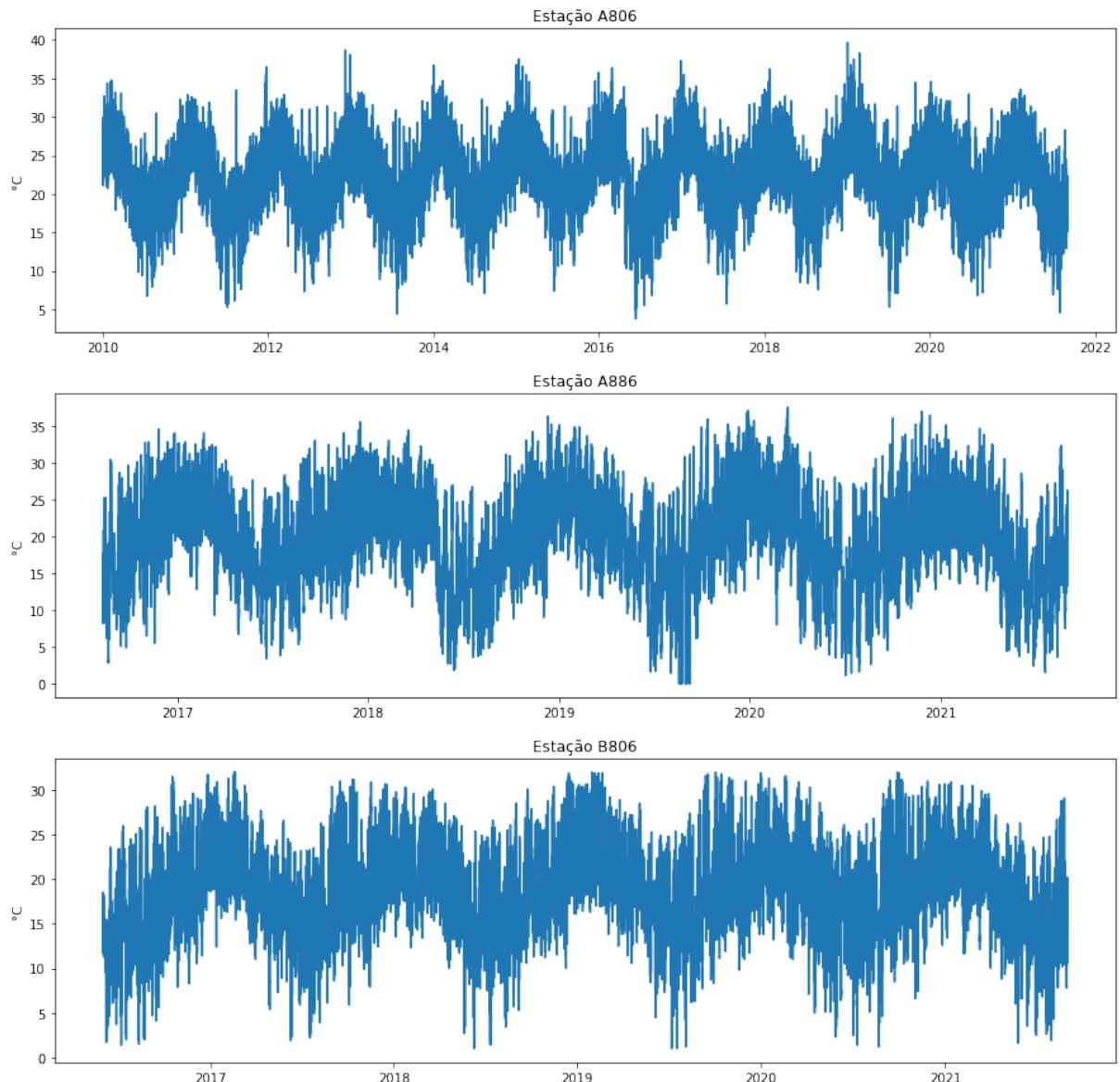
Por fim, após a escolha da estação de estudo, serão realizadas algumas análises estatísticas para comprovar a coerência e confiabilidade dos dados.

3.6.1 Variáveis climáticas e estações meteorológicas de estudo

As estações meteorológicas escolhidas para esta análise são: A806, A886 e B806. Os gráficos apresentados nas Figura 20, 21, 22 e 23 demonstram o comportamento das variáveis selecionadas para estudo ao longo dos onze anos de dados armazenados, sendo elas: **temperatura máxima, umidade relativa do ar, pressão atmosférica e velocidade do vento**.

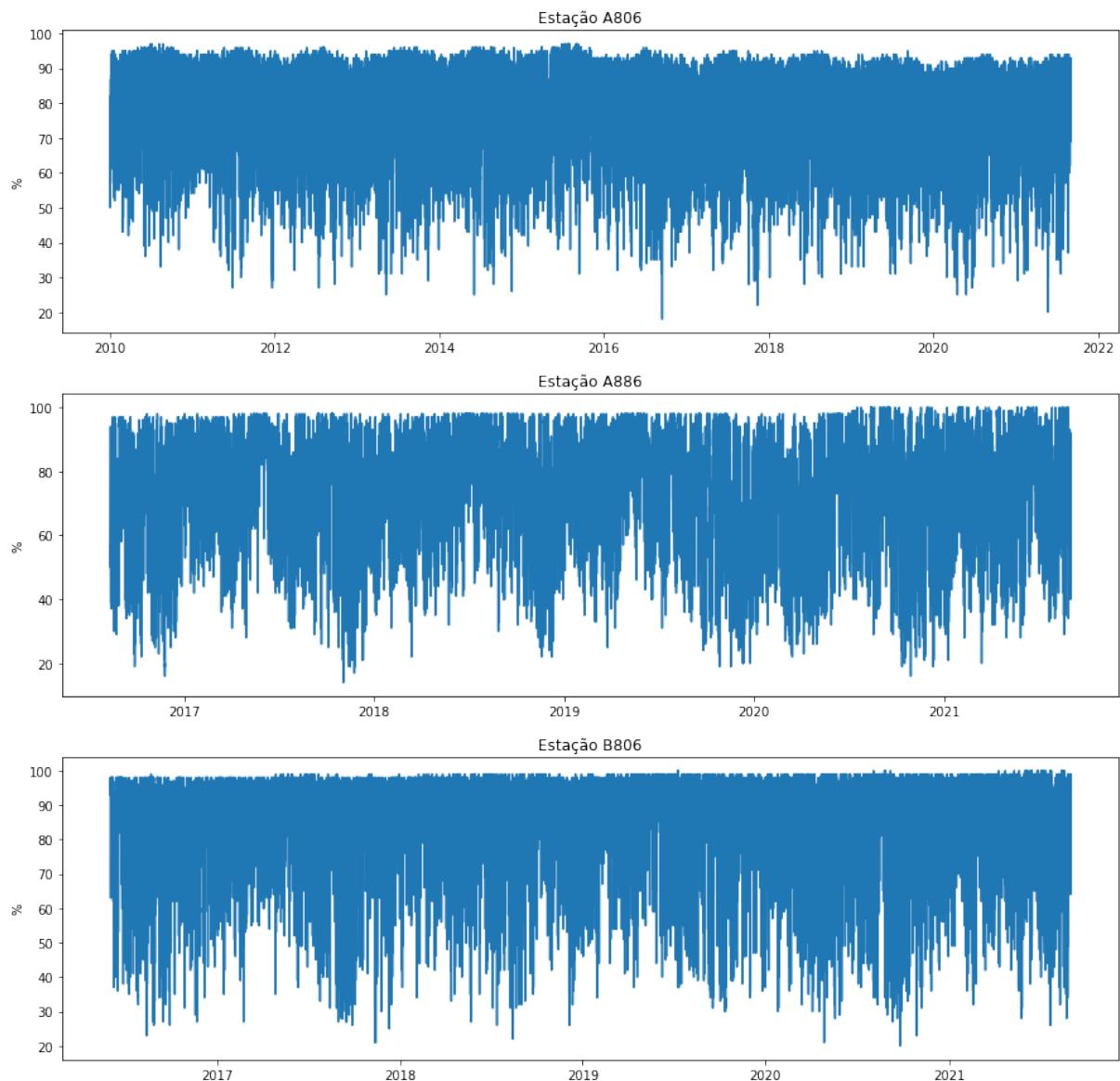
Através dos gráficos, é possível perceber que mesmo estando em localidades muito distantes uma das outras, as variáveis apresentam um comportamento muito similar ao longo de todos os anos. Essa característica é essencial para a otimização de tempo e de recursos computacionais para com o presente trabalho, visto que é possível definir os parâmetros a serem utilizados no modelo de *Machine Learning* (ML) para apenas uma única estação e depois replicar para as outras.

Figura 20 – Comportamento da variável Temperatura Máxima.



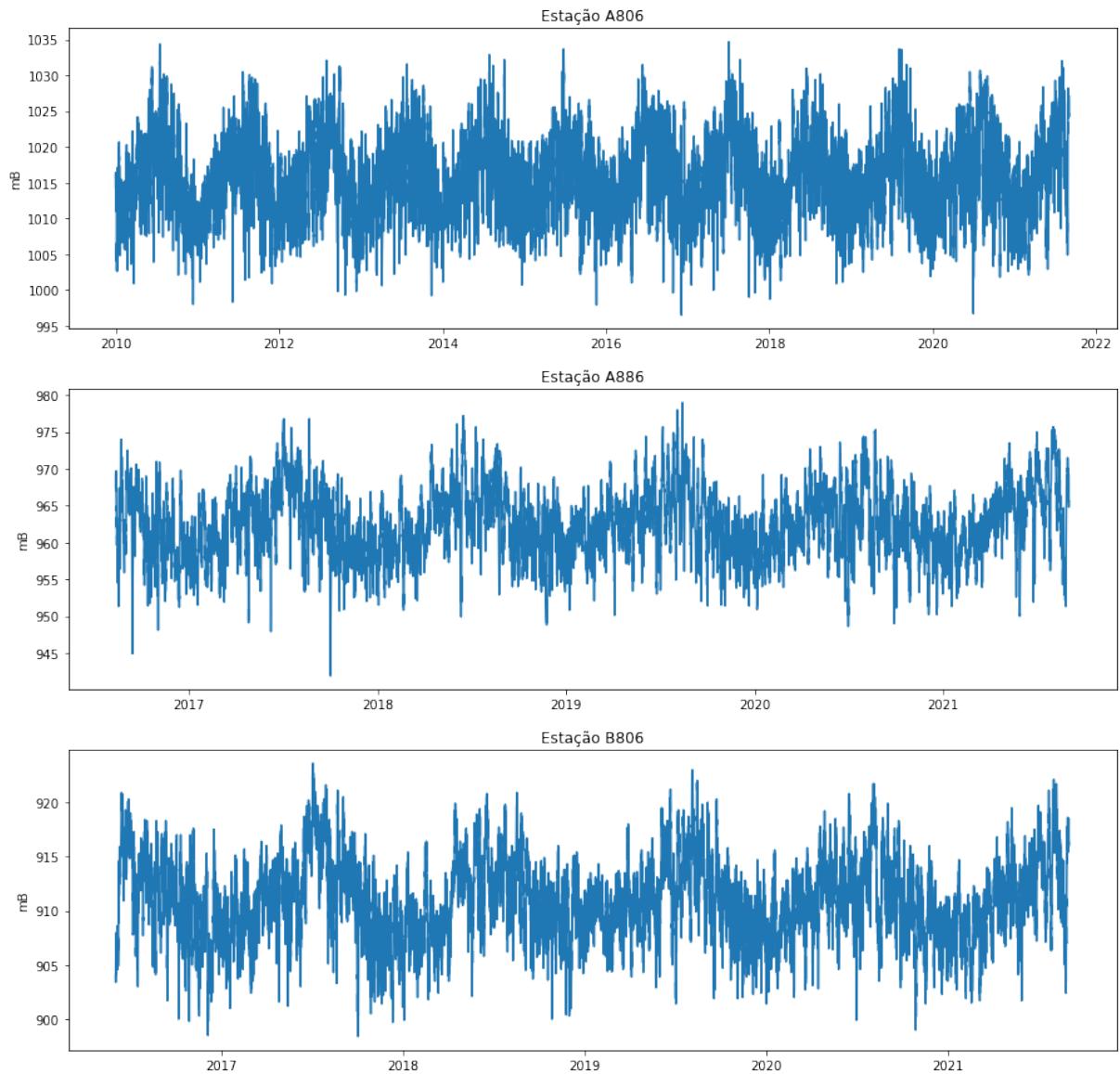
Fonte: O próprio autor.

Figura 21 – Comportamento da variável Umidade Relativa do Ar.



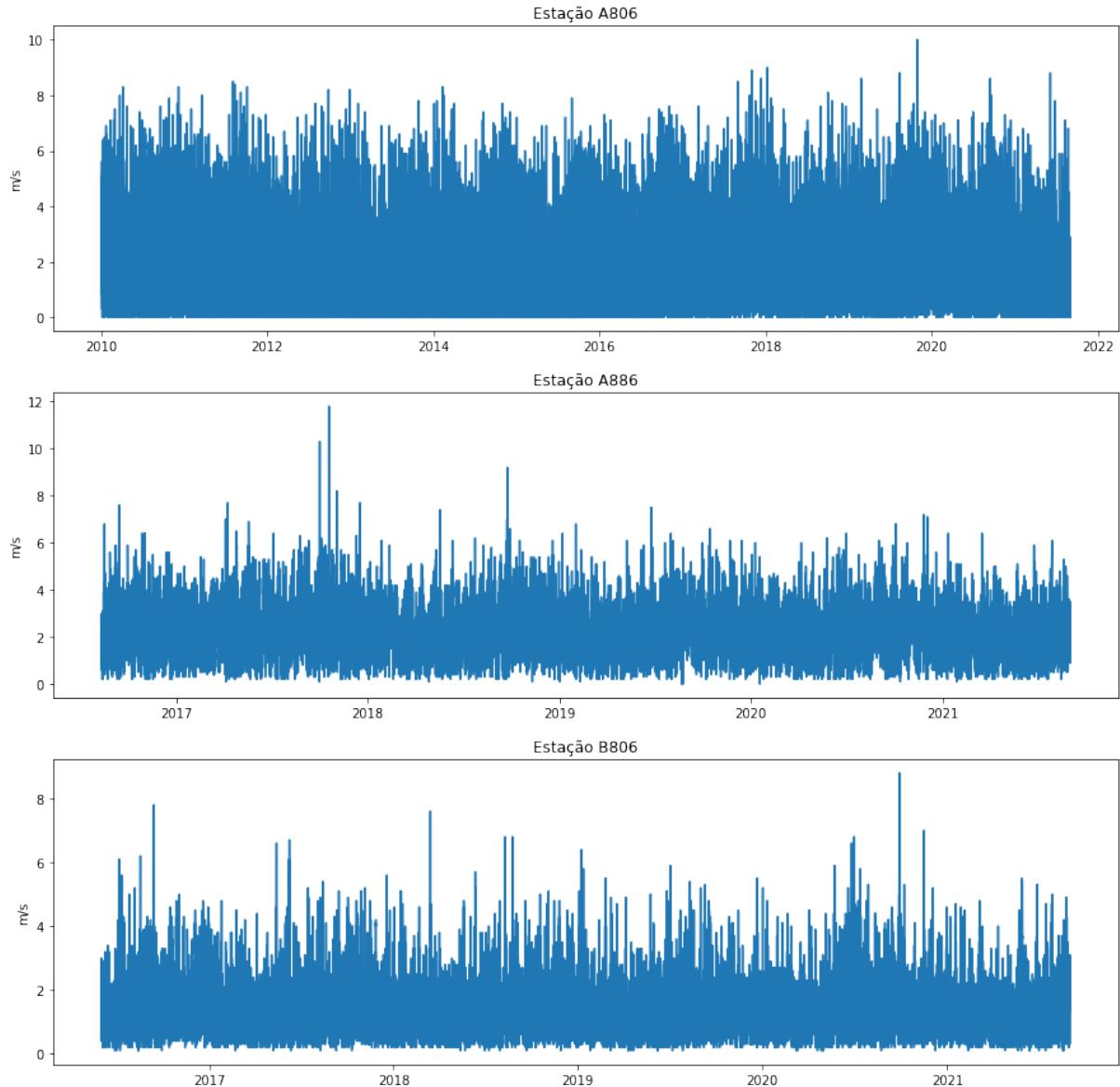
Fonte: O próprio autor.

Figura 22 – Comportamento da variável Pressão Atmosférica.



Fonte: O próprio autor.

Figura 23 – Comportamento da variável Velocidade do Vento.



Fonte: O próprio autor.

Com essa comparação realizada entre as estações, foi determinado que a escolhida será a **A806**, devido a sua localidade ser na Grande Florianópolis, região onde está sendo desenvolvida esta pesquisa. Ao todo se tem 101,614 registros de capturas de dados ao longo dos onze anos de operação do equipamento, o que tornará a geração de modelos LSTM muito confiável, visto que com a grande quantidade de dados o algoritmo terá maior facilidade em encontrar o comportamento das variáveis e assim ter uma melhor acurácia em sua previsão.

3.6.2 Análise descritiva dos dados da estação meteorológica A806

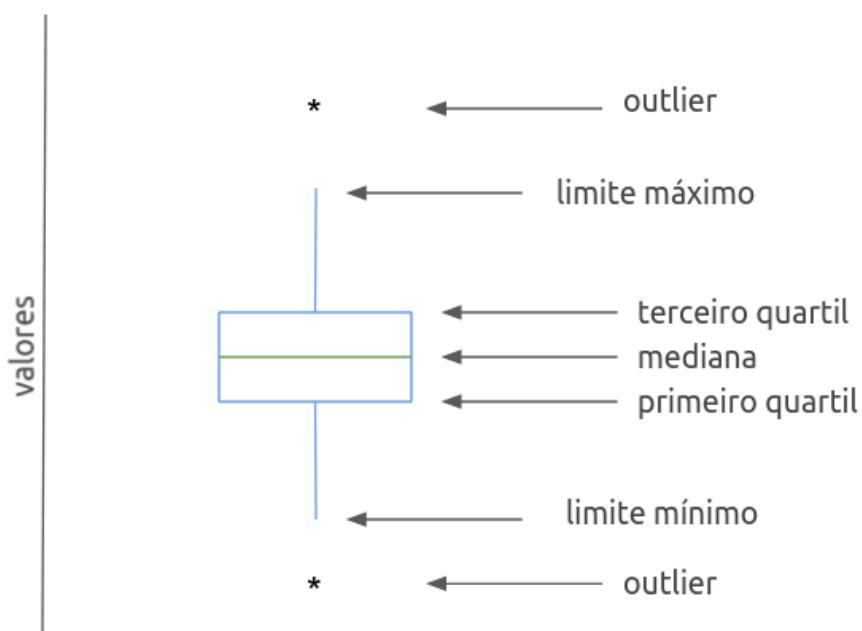
Para este processo de análise dos dados a serem utilizados para o treinamento dos modelos de *Machine Learning* (ML) foi escolhido o gráfico do tipo *boxplot*, pois através dele será possível visualizar valores discrepantes (*outliers*), medianas e os quartis do conjunto.

A Figura 24 demonstra como obter essas informações através do gráfico. Os valores *outliers* são representados por asteriscos, podendo haver *outlier* superior e inferior. Em algumas ocasiões, apesar de serem valores muito diferentes da maioria do conjunto, eles não representam erros e são considerados valores corretos (REIS; REIS, 2002).

Os limites máximos e mínimos estão representados pelos extremos das retas que saem da caixa central do gráfico, porém é importante ressaltar que o comprimento dessas retas não pode ultrapassar 1.5 vezes a altura da caixa, ou seja, em alguns casos o valor máximo descrito pelo gráfico não indica realmente o valor mais alto do conjunto, podendo este estar sendo representado pelo *outlier* mais elevado.

A linha central da caixa indica a mediana, a linha da base o primeiro quartil (Q1) e a do topo o terceiro quartil (Q3). Por fim, o gráfico também indica que 50% dos dados se encontram entre o Q1 e Q3, 25% abaixo de Q1 e 25% acima de Q3.

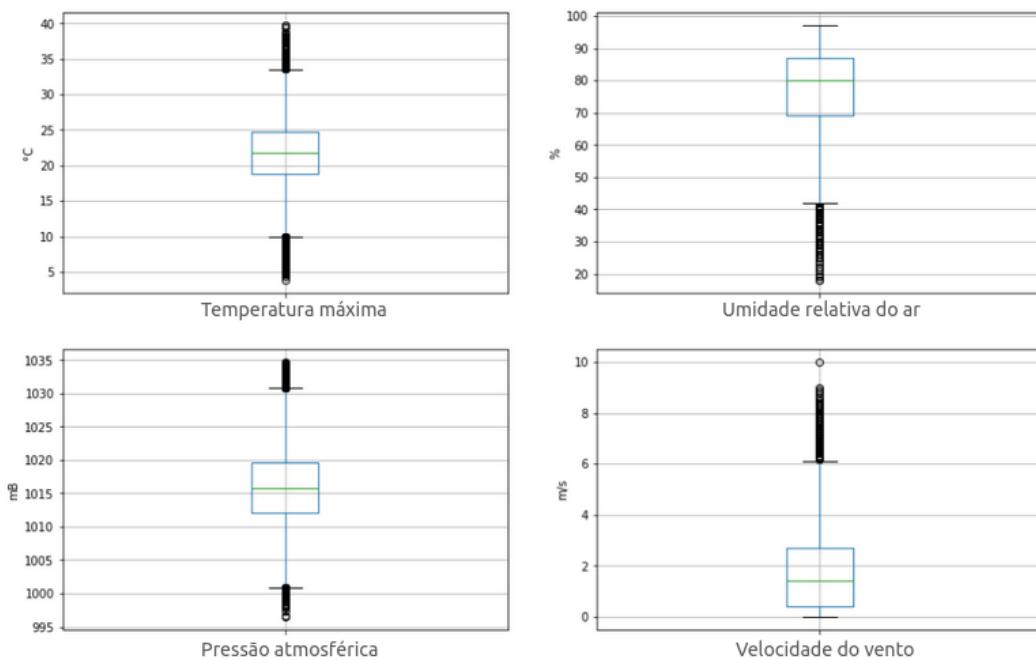
Figura 24 – Informações contidas em um *boxplot*.



Fonte: O próprio autor.

Com essa explicação básica da interpretação do gráfico *boxplot* já é possível realizar a análise descritiva dos dados da estação A806. A Figura 25 demonstra um gráfico para cada variável em estudo. Através desta Figura pode-se perceber que as variáveis **Temperatura** e **Pressão atmosférica** possuem uma simetria em seus valores, visto que as retas possuem praticamente o mesmo comprimento para ambos os sentidos (acima e abaixo). Outro ponto importante de análise é em relação aos *outliers*, que conforme comentado, não significa que são valores errôneos, apenas que estão acima de 1.5 vezes a altura da caixa (Q3-Q1).

Figura 25 – *Boxplot* das variáveis em estudo.



Fonte: O próprio autor.

Com o gráfico *boxplot* se tem uma visualização comportamental dos dados, para uma análise mais quantitativa a Tabela 4 apresenta as principais informações desses valores para comprovar a consistência do conjunto a ser utilizado.

Tabela 4 – Análise descritiva da estação A806.

	Máximo	Mínimo	Média	Mediana	Desvio padrão
Temperatura	39.7	3.8	21.7	21.8	4.39
Umidade	97.0	18.0	77.5	80.0	12.19
Pressão atm.	1034.7	996.5	1015.96	1015.7	5.43
Vel. vento	10.0	0.0	1.69	1.4	1.5

3.7 Estratégias desenvolvidas e resultados obtidos

Nesta Seção são detalhadas as técnicas desenvolvidas para a predição dos parâmetros climáticos. Foram criados três diferentes cenários com abordagens distintas, a fim de se obter os melhores resultados e tornar a pesquisa mais abrangente em caso de utilização em um sistema Sistema de Apoio à Decisão (SAD), pois assim, o usuário poderá escolher o tipo de predição que mais se enquadra em suas necessidades. Os cenários desenvolvidos são: **Próxima Hora**, **Modelos em períodos do dia** e **Modelos Subamostrados**. O *script* desenvolvido para esta etapa está disponível [aqui](#).

3.7.1 Cenário 1: Próxima Hora

Esta proposta tem por objetivo prever uma amostra (1 hora) futura a partir dos dados de entrada reais, sendo o mais intuitivo em relação aos outros cenários que serão explicados adiante.

A estratégia requer sempre a atualização dos dados reais para realizar a previsão da amostra futura, ou seja, a cada nova hora que se passa uma nova previsão será realizada com os últimos 24 dados reais armazenados. Haverá apenas um modelo para cada variável climática, o qual possuirá os mesmos hiperparâmetros propostos na Seção 3.4.

3.7.1.1 Modo de funcionamento

O mecanismo de funcionamento dessa estratégia é simples, são necessárias **24 amostras reais** para realizar a previsão de **1 amostra futura**, ou seja, o valor que aquela variável terá na próxima hora. Porém, para fins teóricos e de melhor visualização, foi realizado uma simulação com uma grande quantidade de dados reais a fim de gerar gráficos que comprovem o bom funcionamento do modelo.

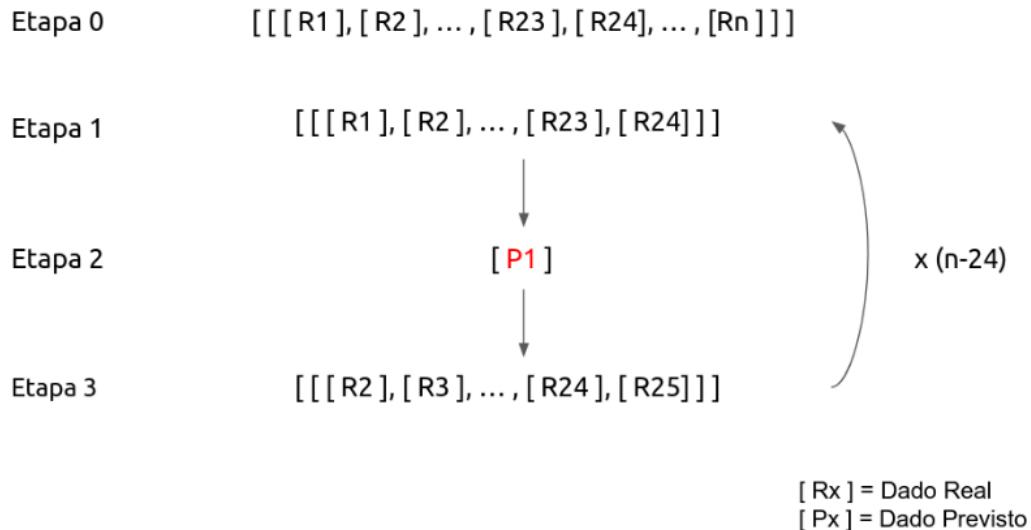
A Figura 26 demonstra as etapas deste processo teórico do Modelo de Próxima Hora. Na **Etapa 0** são selecionados todos os dados reais que serão utilizados na simulação, sendo **n** um número inteiro que representa o tamanho do conjunto. Durante a **Etapa 1** os 24 primeiros dados reais são retirados desse conjunto inicial, e armazenados para serem utilizados em outra etapa do processo.

Já na **Etapa 2** uma amostra é prevista utilizando os 24 dados reais separados, essa nova amostra é armazenada juntamente com os 24 primeiros dados. E por fim, na **Etapa 3** o conjunto de dados reais é atualizado com um novo dado real para realizar uma nova previsão.

Todo esse processo é realizado **n - 24** vezes, sendo 24 o valor de amostras necessárias para o funcionamento do modelo. No final deste *loop* um conjunto de **n** amostras é retornado, sendo os 24 primeiros dados as primeiras amostras utilizadas no início do processo e o

restante são as amostras previstas.

Figura 26 – Lógica do Cenário de Próxima Hora.



Fonte: O próprio autor.

3.7.1.2 Resultados

A Figura 27 demonstra o resultado obtido neste cenário para cada uma das variáveis em estudo, sendo a linha em **azul os dados reais** e a linha em **laranja os dados previstos**. Foram realizadas 1000 iterações para esta simulação, e a partir dos gráficos já é possível confirmar a real eficiência do algoritmo LSTM.

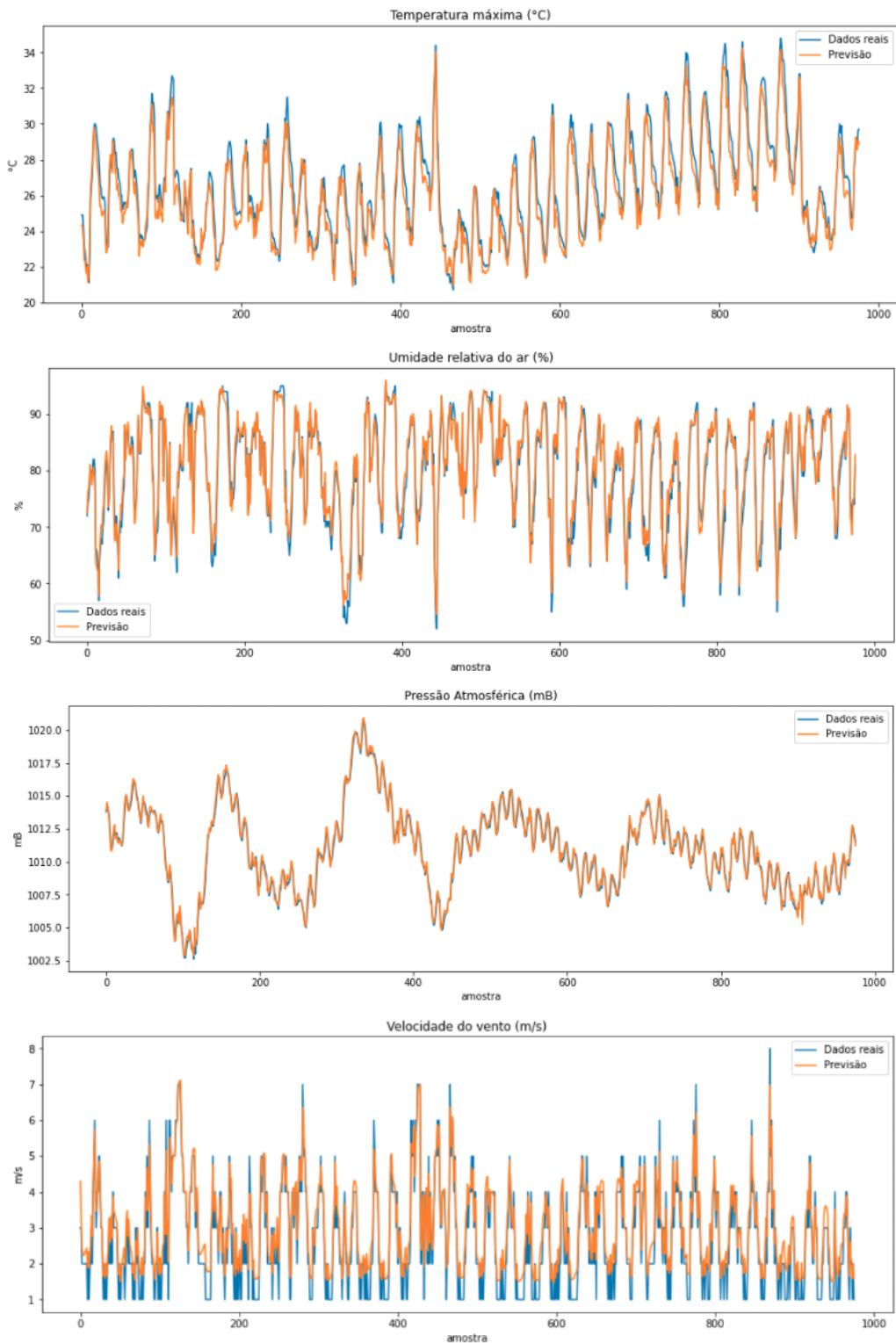
Para parâmetros com comportamentos bem definidos ao longo do tempo como **temperatura, umidade do ar e pressão atmosférica**, conforme vistos na Subseção 3.6.1, a acurácia da previsão (dada por MAPE) foi extremamente satisfeita, chegando em índices de erros máximos de 2,51%.

Por outro lado, para a variável **velocidade do vento**, a qual não possui um comportamento muito bem definido e que constantemente tem seu valor alterado ao longo das horas, ocorreu um erro percentual absoluto médio de 23.12%. Porém, mesmo apresentando este erro, ainda há uma grande margem de acerto para o usuário, garantindo-o já uma grande vantagem na realização de suas tarefas.

Tabela 5 – Métricas de erros para o Cenário de Próxima Hora.

	MAE	RMSE	MAPE
Temperatura	0.675	0.784	2.51
Umidade	1.53	1.93	2.0
Pressão atm.	0.303	0.356	0.03
Vel. vento	0.365	0.445	23.12

Figura 27 – Resultados Cenário 1.



Fonte: O próprio autor.

3.7.2 Cenário 2: Modelos em períodos do dia

Este cenário busca desenvolver um novo tipo de estratégia, sendo que diferentemente da técnica anterior onde há somente a criação de um único modelo LSTM, aqui se terá ao todo quatro modelos para cada variável climática.

Esses modelos irão se diferenciar pela sua base de dados de treinamento, cada uma contendo leituras dos dados em horários específicos do dia divididos entre os períodos da **madrugada**, **manhã**, **tarde** e **noite**. Com essa divisão, cada modelo gerado será responsável por prever **12 amostras futuras**, ou seja, 2 dias à frente com **24 amostras reais** de entrada (4 dias de dados reais). Essa estratégia foi criada pois ao realizar uma operação recursiva do LSTM, utilizando um dado previsto para prever o outro, o erro propagado tem grande influência após poucas amostras futuras. Dessa forma, prevendo apenas doze por modelo é possível ter um resultado sem ter interferência significativa do erro.

A Tabela 6 demonstra os horários que cada modelo será responsável por realizar a previsão, sendo que cada conjunto de 6 amostras para cada modelo é equivalente a previsão de 1 dia.

Tabela 6 – Modelos e seus respectivos horários de predição para o Cenário 2.

	Horários					
Modelo 1	0	1	2	3	4	5
Modelo 2	6	7	8	9	10	11
Modelo 3	12	13	14	15	16	17
Modelo 4	18	19	20	21	22	23

3.7.2.1 Modo de funcionamento

Como citado, esta técnica de previsão utiliza 24 dados de entrada reais para prever 12 amostras futuras para cada modelo. É importante destacar que 24 dados correspondem a 4 dias de captura de dados reais, pois ao dividir 24 por 6 (número de horários que cada modelo é responsável por prever), teremos esse valor em dias. Para uma melhor visualização gráfica e comprovação do funcionamento do algoritmo em grande escala, foi realizada uma simulação com uma alta quantidade de dados, a qual tem seu funcionamento descrito na Figura 28.

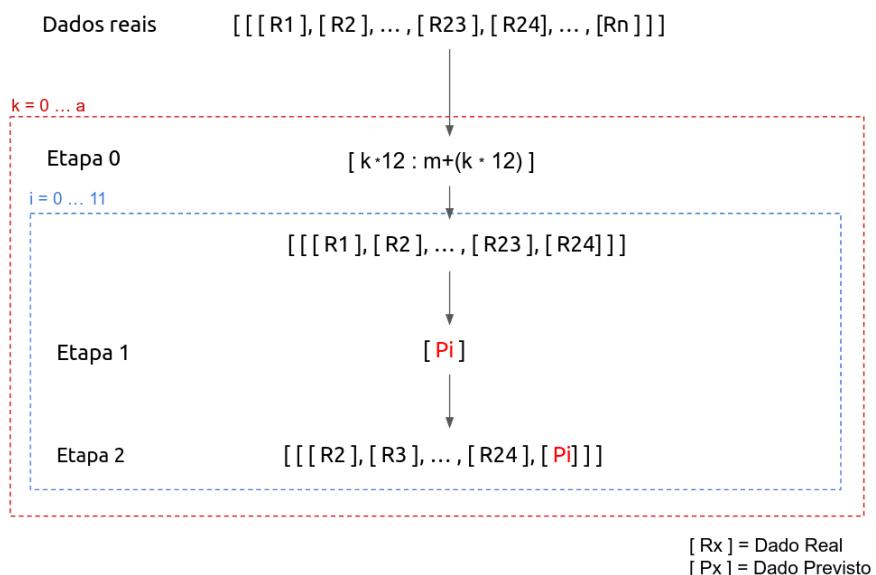
Ao selecionar os dados reais a serem utilizados na simulação, sendo **n** o número de amostras deste conjunto, serão realizadas **a** iterações representadas por **k**, onde cada uma irá prever um conjunto de 48 amostras, sendo 12 para cada modelo.

Na **Etapa 0** são selecionados **m** dados do conjunto de amostras reais para iniciar a simulação, sendo $m = 24$. A cada nova iteração o valor de **k** (retângulo em vermelho) é

incrementado em 1, fazendo com que o conjunto de dados reais nesta etapa seja sempre atualizado em 12 novas amostras reais, ou seja, os valores dos dois primeiros dias são atualizados com os novos dois dias.

Dentro do segundo laço de repetição (retângulo em azul) ocorrem 12 iterações a partir dos dados selecionados, sendo que cada uma gera uma amostra prevista como indicado na **Etapa 1**. Esta nova amostra irá compor o novo conjunto de dados para a próxima previsão, como indicado na **Etapa 2**.

Figura 28 – Lógica do Cenário de Modelos em períodos do dia



Fonte: O próprio autor.

Toda a lógica do bloco em vermelho ocorre **a** vezes para os quatro modelos LSTM gerados, alterando somente os dados de entrada na **Etapa 0** devido a diferença dos horários que cada um foi treinado. Ao final da simulação haverá um total ($a \times 48$) amostras futuras, pois 48 é a soma de 12 amostras dos 4 modelos, totalizando 2 dias de previsão para cada iteração.

Para finalizar a simulação é necessário interpolar os dados previstos em cada modelo para que os mesmos fiquem na sequência temporal correta. Feito isso, todo o processo de previsão foi realizado para a simulação.

3.7.2.2 Resultados

Como já citado na Sub-subseção 3.7.2.1, para este cenário foi realizada uma simulação com uma grande quantidade de dados a fim de se obter uma melhor visualização gráfica da previsão gerada. Porém, caso este mecanismo seja aplicado em um sistema SAD a técnica em questão irá necessitar apenas dos últimos quatro dias de amostras reais para conseguir prever os próximos dois dias.

É importante ressaltar que diferentemente do Cenário 1 que utiliza 24 dados reais para prever 1 amostra futura, cada modelo atual irá prever 12 amostras futuras com a mesma quantidade de amostras reais, utilizando a técnica de recursividade, ou seja, após prever 1 hora, utiliza este valor para prever as próximas. Com isso, uma taxa de erro maior já é esperada devido a propagação do erro assim que as amostras previstas são reutilizadas.

A partir da Figura 29 é possível visualizar os resultados gráficos obtidos na simulação realizada, a qual utilizou $a \times 48$ amostras, sendo $a = 20$. É interessante observar que as variáveis **temperatura** e **umidade** obtiveram bons índices de acurácia conforme demonstrado na Tabela 7, o que é um indicativo de que a divisão das amostras reais por períodos do dia foi uma boa estratégia.

Por outro lado, para as variáveis **pressão atmosférica** e **velocidade do vento** não foi possível, até o momento, concluir de forma óbvia os resultados obtidos para esses parâmetros, devido aos valores baixos das métricas de erros que não compactuam com o resultado gráfico. Isto pois, ao visualizar o comportamento da suas previsões, é fácil identificar que não se atingiu uma previsão satisfatória, porém seus cálculos de acertos demonstram o oposto, sendo que os valores MAE e RMSE da pressão atmosférica apresentaram valores melhores que os parâmetros temperatura e umidade, que graficamente apresentaram resultado superiores.

Dessa forma, espera-se que em um futuro próximo haja melhor entendimento da dinâmica do comportamento das variáveis para conclusões mais assertivas. Para ocorrer essa melhoria na previsão é necessário partir para uma outra estratégia de separação de dados, diferente desta apresentada, visando encontrar maneiras que respeitem a variação dos parâmetros em questão.

Tabela 7 – Métricas de erros para o Cenário de Modelos em períodos do dia ($a = 20$).

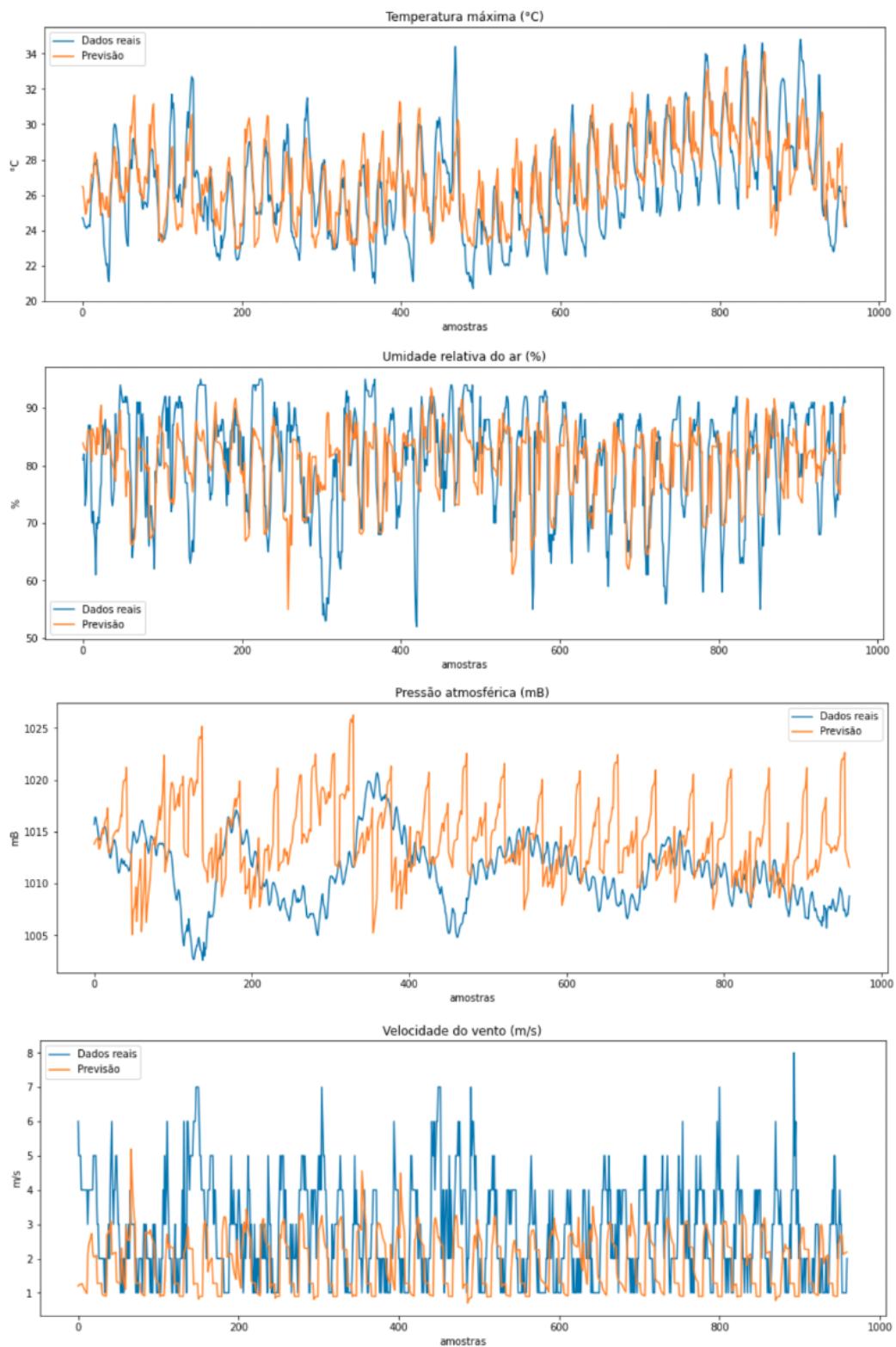
	MAE	RMSE	MAPE
Temperatura	1.58	1.98	6.00
Umidade	6.20	8.00	8.00
Pressão atm.	4.70	6.16	0.47
Vel. vento	1.48	1.89	55.58

A Tabela 8 demonstra as métricas de erros para o processo em que $a = 1$, ou seja, serão previstos apenas 2 dias. Conforme é possível visualizar, os valores estão muito próximos aos apresentados na simulação, concluindo o seu funcionamento para esta menor quantidade de dados.

Tabela 8 – Métricas de erros para o Cenário de Modelos em períodos do dia ($a = 1$).

	MAE	RMSE	MAPE
Temperatura	1.37	1.69	5.60
Umidade	6.55	8.73	8.88
Pressão atm.	2.33	3.44	0.23
Vel. vento	1.92	2.23	49.64

Figura 29 – Resultados Cenário 2.



Fonte: O próprio autor.

3.7.3 Cenário 3: Modelos Subamostrados

Neste cenário é aplicada uma estratégia parecida com a apresentada na Sub-subseção 3.7.2, a qual também irá utilizar quatro modelos para cada variável porém haverá uma outra divisão de horários.

Foi determinado que para este caso os horários serão intercalados em um período de 4 horas, conforme demonstrado pela Tabela 9. Essa subdivisão visa dar granularidade aos dados, visto que alguns parâmetros climáticos como **temperatura** e **umidade** não possuem uma variação tão expressiva de hora em hora, e com essa abordagem será possível analisar este comportamento através das previsões realizadas.

Tabela 9 – Modelos e seus respectivos horários de predição para o Cenário 3.

Horários						
Modelo 1	0	4	8	12	16	20
Modelo 2	1	5	9	13	17	21
Modelo 3	2	6	10	14	18	22
Modelo 4	3	7	11	15	19	23

3.7.3.1 Modo de funcionamento

A lógica de operação presente neste cenário iguala-se à apresentada na Sub-subseção 3.7.2.1, diferenciando apenas nos sub-conjuntos de dados reais utilizados, assim como no processo de interpolação dos dados finais. É importante relembrar que este processo de interpolação é necessário pois ao final da lógica de previsão haverão quatro conjuntos de dados separados, um para cada modelo, dessa forma será necessário juntá-los e organizá-los a fim de se obter a sequência correta de horários previstos.

Também é realizada uma simulação com uma grande quantidade de dados para se obter uma melhor visualização gráfica do resultado final, assim como tabelas de métricas de erros serão apresentadas tanto para esta simulação quanto para o processo de previsão de dois dias.

Apesar de apresentarem a mesma lógica, foi decidido que essas duas estratégias irão possuir implementações independentes no *back-end* do sistema, visando uma maior facilidade e escalabilidade quanto a melhorias e/ou correções futuras, porém claro, sempre será utilizado boas práticas de desenvolvimento de *software* a fim de tornar tudo mais prático e eficiente, ou seja, haverá compartilhamento de códigos entre as duas estratégias em certos pontos.

3.7.3.2 Resultados

Conforme apresentado na Figura 30 foram obtidos bons resultados para as variáveis **temperatura** e **umidade**, com valores de MAPE muito próximos ao do Cenário 2, como demonstra a Tabela 10 de métricas de erro. Analisando graficamente esses dois parâmetros climáticos, é possível perceber que o algoritmo teve dificuldade de prever valores com grande variação do valor médio, como por exemplo valores mais altos, próximos da amostra de número 500 e 900 para a temperatura, assim como para valores mais baixos para a umidade relativa do ar.

Porém mesmo apresentando esses índices de erro, o algoritmo pode ser considerado eficiente para o cenário, visto que em muitos casos o agricultor não possui nenhuma informação sobre o seu microclima, e com essa previsão sua atividade poderá se tornar muito mais produtiva.

Para os parâmetros **pressão atmosférica** e **velocidade do vento** também não foi possível concluir de forma sucinta assim como no Cenário 2. Isto devido a esta estratégia também não respeitar o padrão de variação das variáveis ao longo do tempo, necessitando um maior estudo na área de climatologia para identificar os seus comportamentos, a fim de encontrar uma forma correta para a utilização do LSTM.

Tabela 10 – Métricas de erros para o Cenário de Modelos subamostrados ($a = 20$).

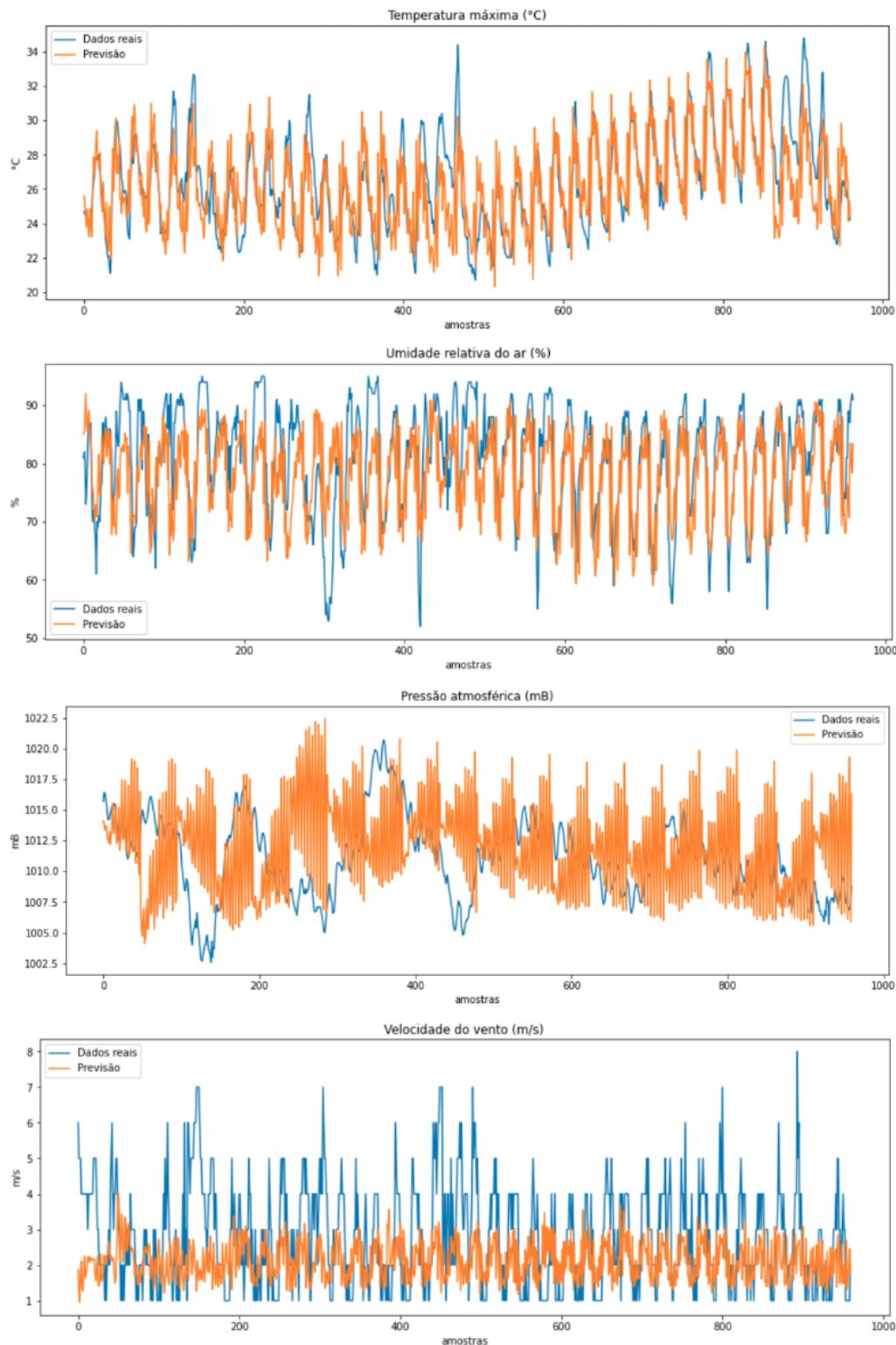
	MAE	RMSE	MAPE
Temperatura	1.64	2.12	6.16
Umidade	6.27	7.98	7.91
Pressão atm.	4.09	5.11	0.405
Vel. vento	1.19	1.55	46.67

A Tabela 11 demonstra as métricas obtidas para o cenário prático, ou seja, utilizando apenas 24 amostras reais para cada modelo. Seus valores ficaram muito próximos ao do processo de simulação, comprovando o funcionamento do algoritmo.

Tabela 11 – Métricas de erros para o Cenário de Modelos subamostrados ($a = 1$).

	MAE	RMSE	MAPE
Temperatura	0.99	1.32	3.87
Umidade	6.27	7.97	7.91
Pressão atm.	2.16	2.81	0.21
Vel. vento	1.82	2.15	47.05

Figura 30 – Resultados Cenário 3.



Fonte: O próprio autor.

3.8 Desenvolvimento da página web

A plataforma *web* visa dar liberdade ao usuário em analisar seus dados coletados pelas estações micrometeorológicas, visualizar previsões e gerenciar sua empresa e usuários. Dentre as estratégias de previsões apresentadas na Seção 3.7, o cenário de subamostras foi o escolhido para integrar ao sistema, o qual atualmente entrega ao usuário previsões de dois dias para as variáveis de temperatura e umidade do ar.

3.8.1 Tecnologias utilizadas

O projeto do sistema possui as três partes principais para uma aplicação, ou seja, é composto por *back-end*, *front-end* e *cloud*. Para o *back-end* decidiu-se utilizar o *Django rest framework*⁸ devido a sua alta popularidade e a grande capacidade de gerir seções e banco de dados. Nesta parte, também foi implementado toda lógica de previsão desenvolvida para a estratégia escolhida. O Apêndice A discute alguns pontos dessa parte do sistema.

Para o *front-end* foi utilizada a tecnologia *Angular*⁹, a fim de facilitar o desenvolvimento e também seguir padrões utilizados em diversas aplicações no mercado. Para visualizar o projeto *front* clique [aqui](#). E por fim, todos os componentes foram hospedados em serviços de *cloud* disponibilizados pela plataforma AWS¹⁰.

3.8.2 Páginas referentes ao monitoramento

Ao selecionar a opção **Monitoramento** na barra de opções lateral, o usuário será direcionado a uma página como a apresentada na Figura 31 e irá visualizar cada uma das suas estações, assim como algumas informações básicas referentes a ela. O bloco representa um equipamento, e indica em qual propriedade está localizado, o número de série do equipamento e o cultivo a ser monitorado.

Ao clicar sobre o ícone de *gráfico* situado na parte inferior esquerda do bloco será carregada a página de visualização dos dados reais, conforme demonstrada na Figura 32. O usuário poderá escolher o período de análise acionando o botão *Buscar dados*, e poderá trocar de variável pelos botões na cor roxa acima do gráfico. Também é possível passar o *mouse* sobre os pontos do gráfico para visualizar a informação mais detalhada.

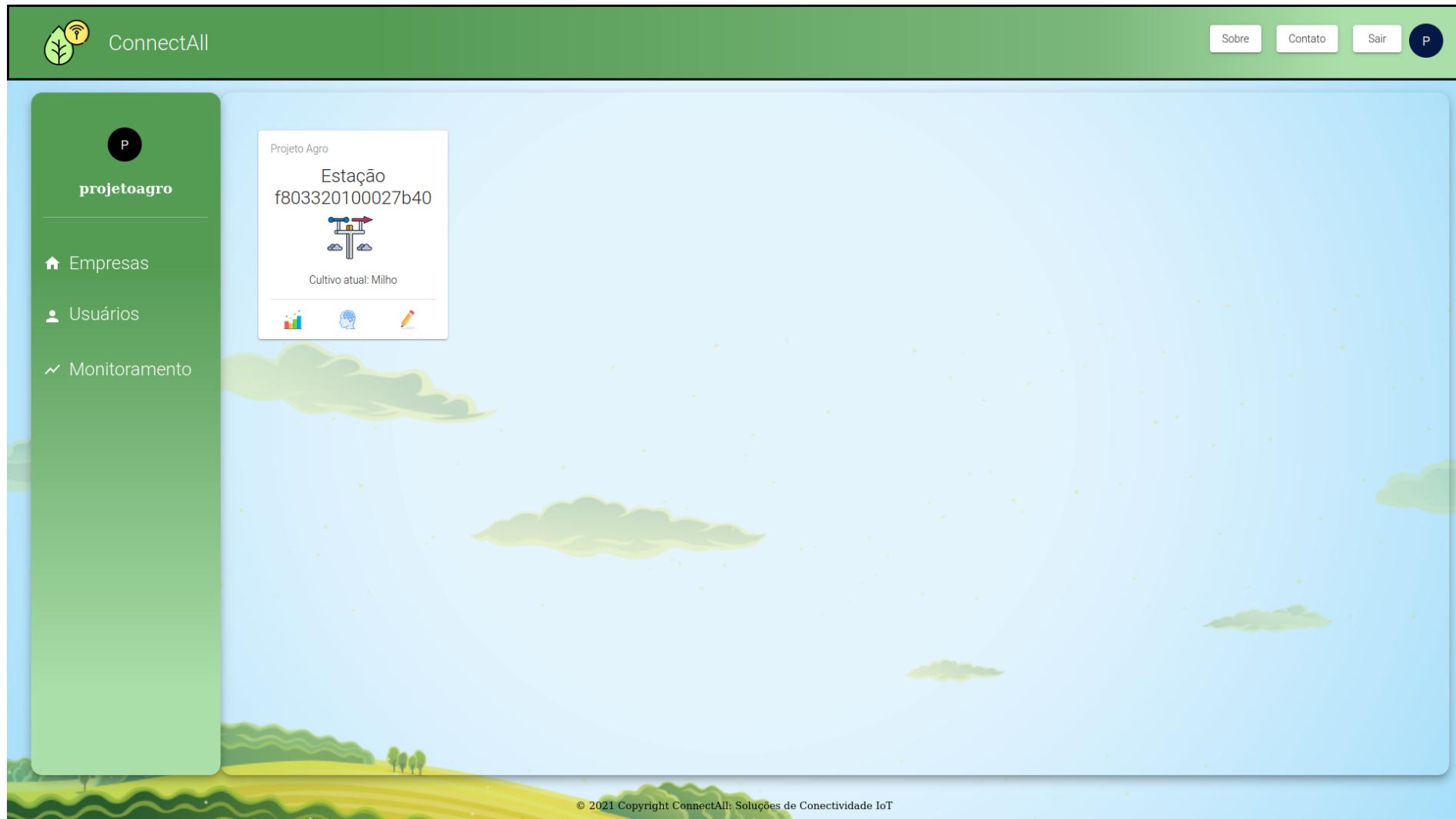
Por fim, no ícone de *inteligência artificial* localizado na parte central do bloco irá redirecionar para a página de previsão, que possuí a aparência demonstrada na Figura 33. Neste gráfico o usuário não tem a liberdade de selecionar intervalos de tempo para a previsão, a qual é fixa para dois dias a frente do dia atual de acesso à página.

⁸ Site oficial Django

⁹ Site oficial Angular

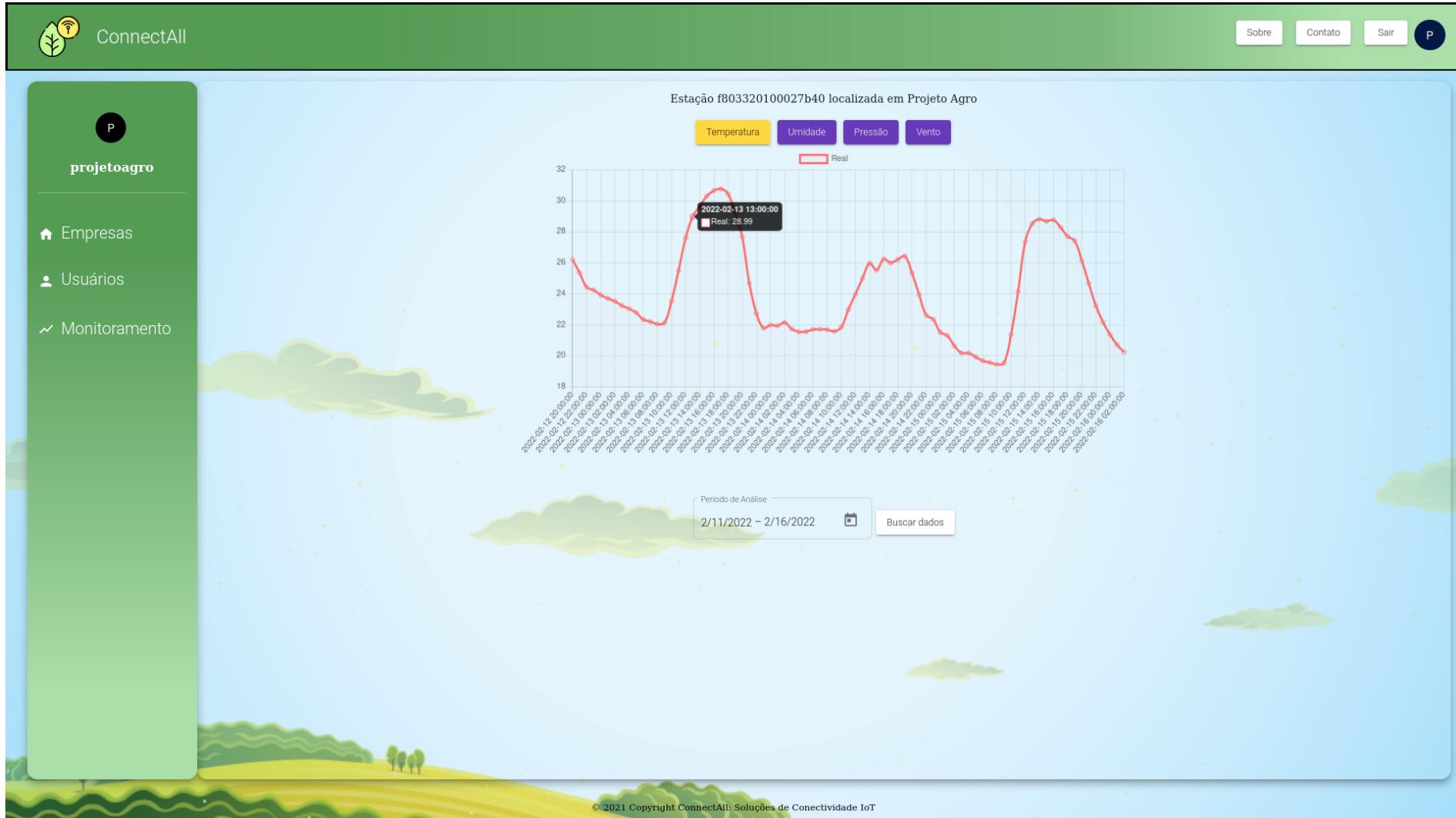
¹⁰ Site oficial AWS

Figura 31 – Página de monitoramento do sistema.



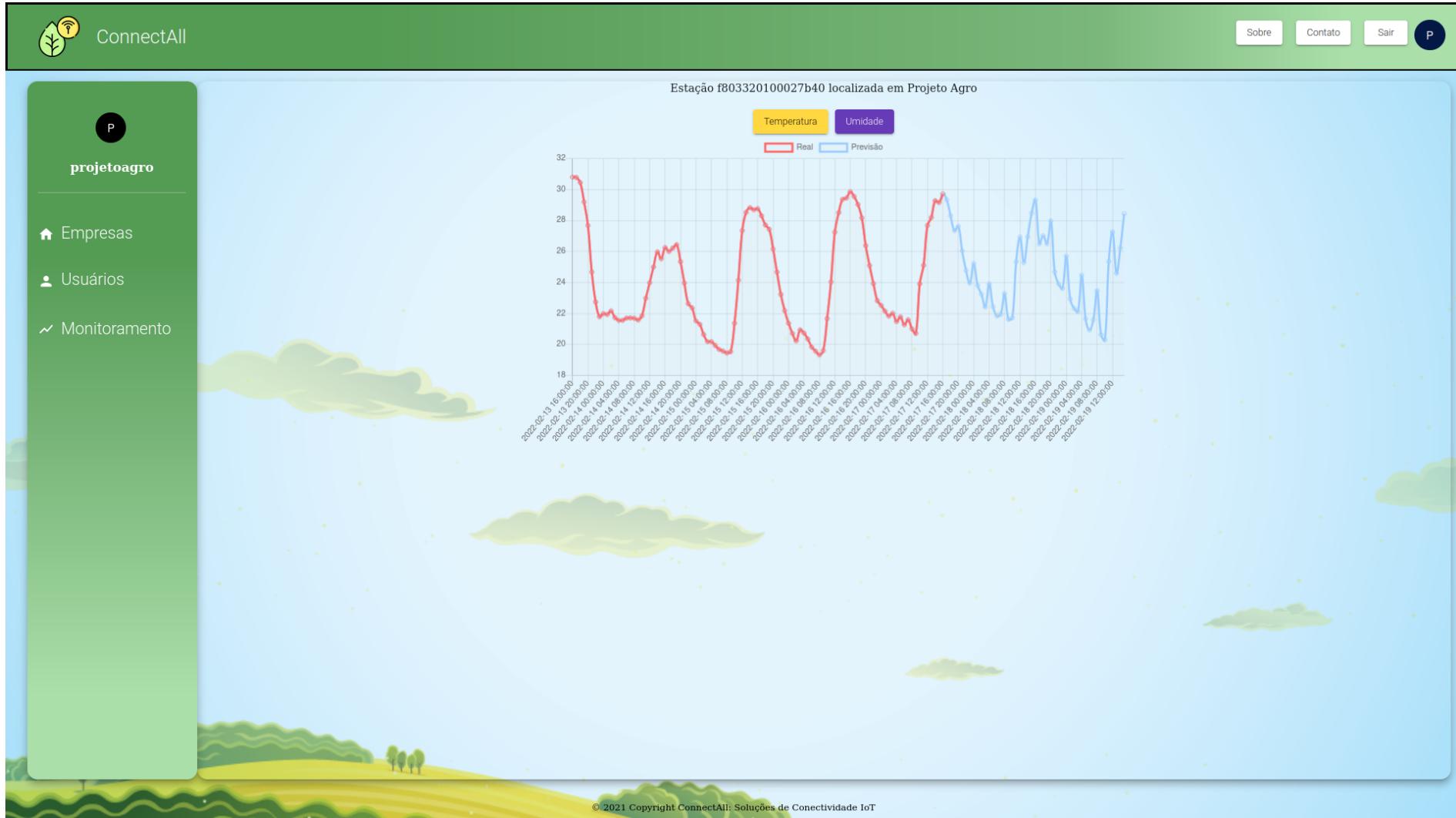
Fonte: O próprio autor.

Figura 32 – Página de gráfico dos dados reais.



Fonte: O próprio autor.

Figura 33 – Página de gráfico dos dados previstos.



Fonte: O próprio autor.

3.8.3 Exemplo de previsão utilizando a plataforma

Para se validar o funcionamento da plataforma em conjunto com a previsão dos parâmetros, foi realizada uma simulação para cada variável. A estratégia para prever os dados aplicada no *back-end* do sistema foi a de subamostras, a qual trabalha com quatro modelos que são responsáveis por horas específicas do dia, intercaladas em 4 horas.

É importante destacar, que para realizar esta simulação foi utilizada uma estação micrometeorológica real instalada para esta pesquisa no município de Jaguaruna, ao sul do estado de Santa Catarina. Seus maiores detalhes não irão entrar em discussão neste trabalho pois o foco da pesquisa está no algoritmo de previsão.

Os gráficos da Figura 34 demonstram as etapas da simulação, as quais o usuário terá acesso ao utilizar o sistema. No gráfico superior estão marcados em **vermelho** os dados reais capturados pela estação entre os dias 13/02/2022 16h até 17/02/2022 16h, e em **azul** estão os dados previstos do dia 17/02/2022 16h até 19/02/2022 16h, contabilizando assim os dois dias de previsão. No gráfico inferior há os dados reais capturados entre os dias 13/12/2022 16h até 19/02/2022 16h.

Visualmente já é possível perceber que a previsão conseguiu acompanhar o comportamento do parâmetro, ocorrendo pequenas variações ao longo da curva. E para uma análise mais detalhada, foram coletados esses dados previstos e reais a fim de calcular as métricas de erros apresentadas na Seção 2.9, as quais estão dispostas na Tabela 12.

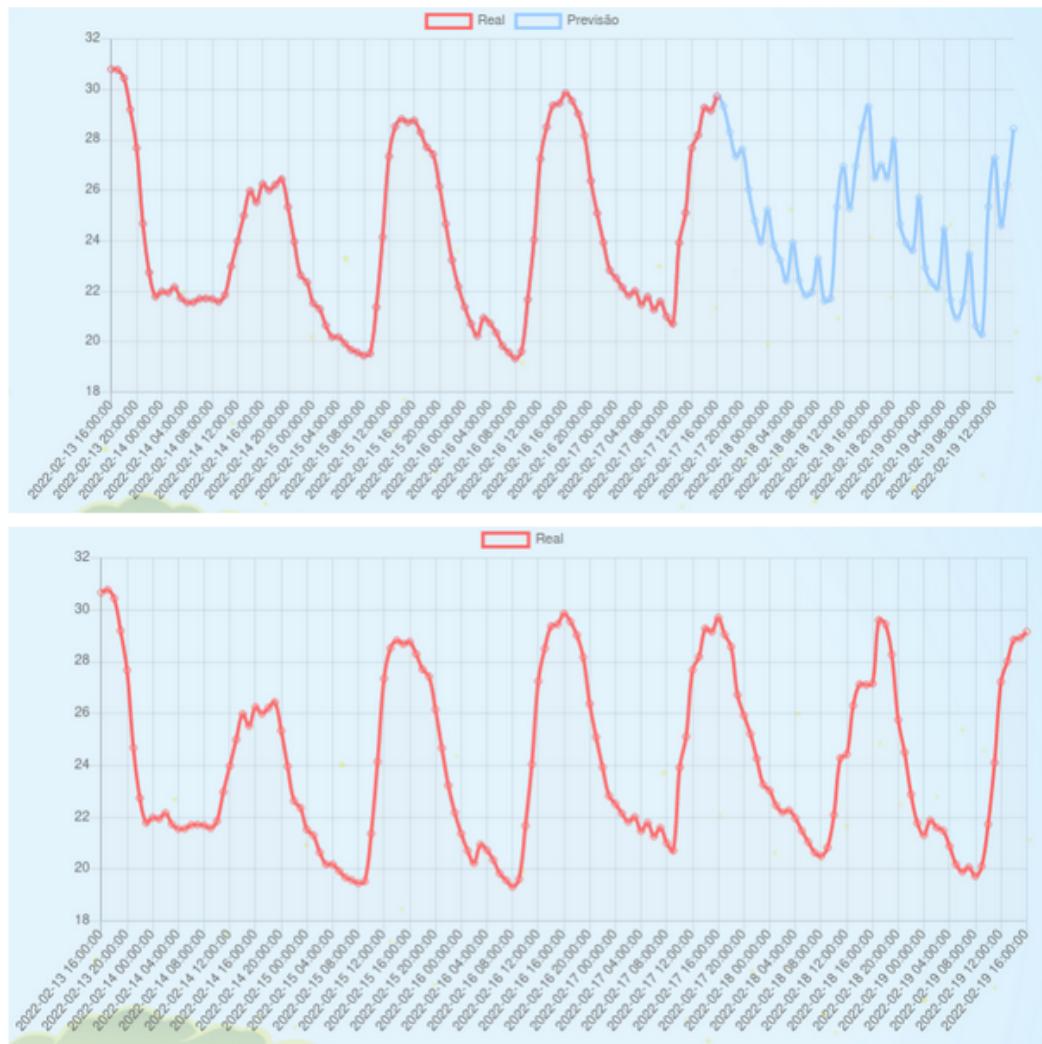
Com essas métricas para a variável em questão, conclui-se que foi obtido um ótimo resultado para os dados coletados pela estação, pois deve-se levar em consideração que o modelo LSTM utilizado foi treinado utilizando dados de uma estação localizada na cidade de Florianópolis. Porém, conforme a estação vai armazenando dados ao longo do tempo, se terá informações suficientes para treinar um modelo somente com as suas próprias capturas, tornando a previsão ainda mais eficiente.

Na Figura 35 há os resultados gráficos para o mesmo intervalo de tempo para a variável umidade. Através dessas imagens e da Tabela 12 também conclui-se que a previsão obteve resultados satisfatórios, gerando valor e conhecimento do clima em sua localidade para o usuário.

Tabela 12 – Métricas de erros para simulação de temperatura e umidade na plataforma.

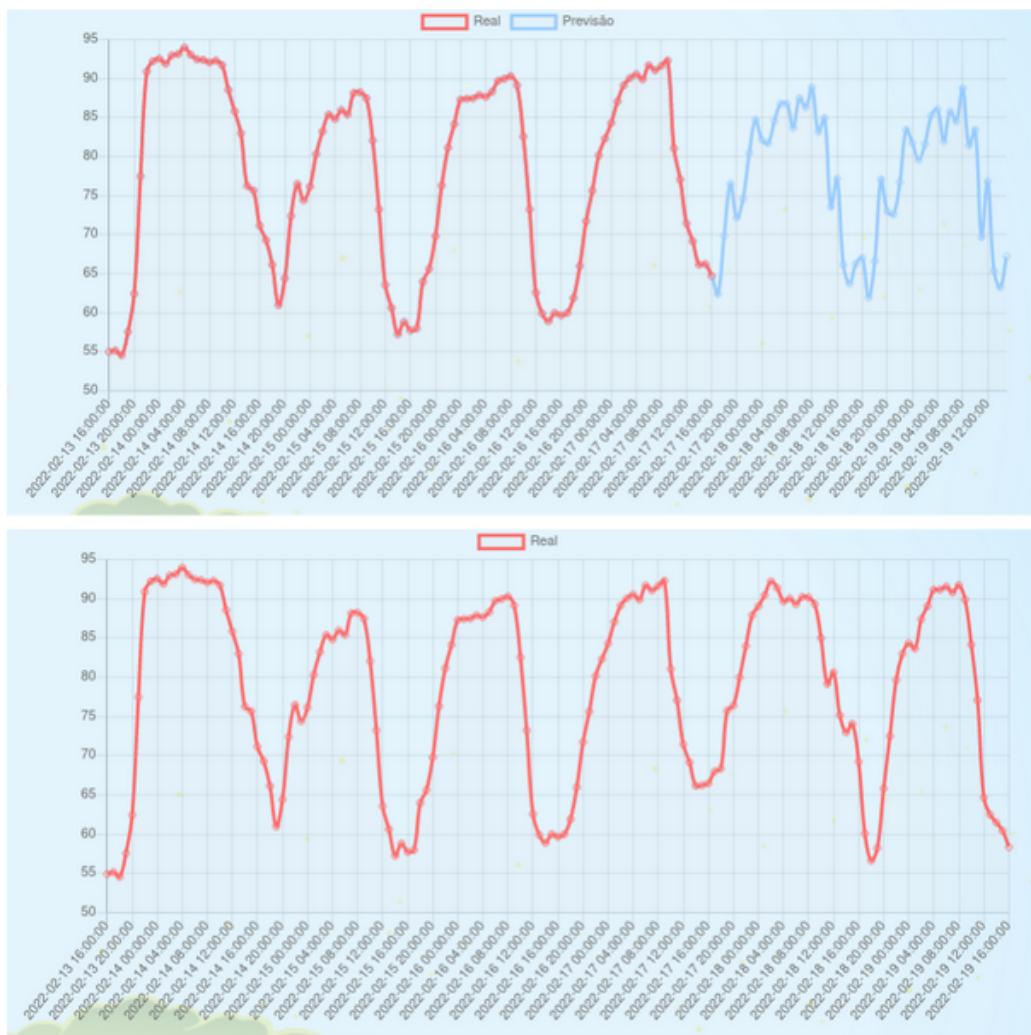
	MAE	RMSE	MAPE
Temperatura	1.82	2.52	7.79
Umidade	8.00	9.85	10.41

Figura 34 – Previsão de temperatura utilizando a plataforma.



Fonte: O próprio autor.

Figura 35 – Previsão de umidade utilizando a plataforma.



Fonte: O próprio autor.

4 CONCLUSÕES

Tendo como objetivo principal, este trabalho visou desenvolver uma Sistema de Apoio à Decisão (SAD) para o setor da agroindústria com foco no médio agricultor, a fim de proporcionar um melhor controle do microclima da sua região e tornar mais eficiente a realização de tarefas que dependem do fator climático.

A pesquisa tem como linha de estudo a análise de dados e Inteligência Artificial, onde através de informações reais obtidas de estação meteorológicas disponibilizadas pelo Instituto Nacional de Meteorologia (INMET) foi possível realizar estudos com uma longa sequência de amostras referentes a diversas variáveis climáticas.

Foi escolhido um método de trabalho procurando atingir um Mínimo Produto Viável, criando algumas funcionalidades que o setor necessita, como por exemplo a previsão de até dois dias do clima em sua propriedade.

Com a crescente popularidade de algoritmos de *Machine Learning* (ML) foi escolhido um que possuísse grande confiabilidade no mundo acadêmico e que apresentasse ferramentas íntegras para a linguagem de programação escolhida. Com isso, o algoritmo *Long-Short Term Memory* (LSTM) apresentou boas características de utilização e adequação ao projeto e foi selecionado como objeto de estudo.

As variáveis *temperatura máxima, umidade relativa do ar, pressão atmosférica e velocidade do vento* foram os parâmetros climáticos utilizados para o treinamento do modelo, onde foram desenvolvidos três estratégias para a aplicação no sistema SAD.

O primeiro cenário denominado de **Próxima Hora** consiste em utilizar 24 amostras reais (24 horas) para realizar a previsão de 1 amostra (hora) futura. Este método apresentou ótimos resultados, possuindo uma taxa de erro máxima muito baixa em torno de 2% para as variáveis temperatura, umidade e pressão. Para a velocidade do vento esse erro foi de 23.12%, porém deve-se levar em consideração que o padrão de comportamento deste parâmetro é muito irregular, e mesmo assim o algoritmo conseguiu identificar alguns padrões na curva.

Já para o segundo cenário, **Modelos em períodos do dia**, foi adotada a estratégia de dividir os dados em horários do dia e criar quatro modelos para cada variável, a fim de entender e visualizar a relação do comportamento das mesmas ao longo das horas. Esta técnica pode prever dois dias com apenas dados coletados de quatro dias anteriores, porém apresenta um erro maior do que o Cenário 1. Mas mesmo com erros maiores foi possível obter bons resultados gráficos e teóricos para a temperatura e umidade as quais são de extrema importância para o agricultor realizar diversas tarefas. Já os parâmetros pressão e

velocidade do vento não apresentaram resultados óbvios para sua conclusão, necessitando de um aprofundamento maior na área de climatologia a fim de se entender melhor o seus comportamentos e criar estratégias coerentes para se atingir uma previsão satisfatória.

Por fim, o último cenário desenvolvido é o chamado **Modelos subamostrados**, o qual subamostra os horários do dia em um período de quatro horas. O modo de funcionamento é o mesmo apresentado no segundo cenário, e também apresentou bons resultados conforme apresentado na Subseção 3.7.3 para temperatura e umidade, porém não obteve sucesso para pressão e velocidade do vento assim como na estratégia de modelos em períodos do dia.

Com o desenvolvimento da plataforma *web* o usuário poderá acessar de forma fácil e rápida os seus dados capturados pela estação, assim como realizar a previsão de dois dias para as variáveis temperatura e umidade, seguindo a estratégia os Modelos Subamostrados. Ao realizar simulações com dados coletados por um estação micrometeorológica real, foram obtidos ótimos resultados, comprovando a eficiência do algoritmo de previsão desses parâmetros e demonstrando a eficiência e facilidade de uso do *software*.

Desta forma, conclui-se que a pesquisa apresentou resultados satisfatórios para um produto inicial que aplica Inteligência Artificial em atividades da agroindústria. Através da grande quantidade de informações de treino e teste obtidas da base de dados foi possível realizar estratégias e técnicas que visaram o melhor resultado possível para previsões, entregando até dois dias à frente o comportamento de variáveis climáticas ao usuário.

4.1 Trabalhos futuros

Para melhorias e continuidade da pesquisa, sugere-se:

- Ajustar o modelo para um melhor desempenho na previsão dos parâmetros climáticos Pressão Atmosférica e Velocidade do Vento;
- Implementar modelos que utilizam mais de uma variável para a previsão, ou seja, utilizar em conjunto variáveis distintas que possam ter alguma relação de comportamento;
- Melhorias estéticas e funcionais na plataforma *web*, tornando-a mais autoconfigurável por parte do usuário.

REFERÊNCIAS

- ANTON-HARO, C.; DOHLER, M. *Machine-to-Machine (M2M) Communications*. [S.l.]: Woodhead Publishing Ltd, 2015. v. 69. 1-1 p. ISBN 9780857092656. Citado na página 18.
- ASA. *International: World Soybean Exports*. 2020. Disponível em: <<http://soystats.com/international-world-soybean-exports/>>. Acesso em: 20 jun 2021. Citado na página 15.
- BNDES. *Produto 7C: Aprofundamento de Verticais-Rural*. 2017. Disponível em: <<https://www.bnDES.gov.br/wps/wcm/connect/site/2fa8f7d1-9939-441d-b8ce-ed3459fcfd4d/relatorio-aprofundamento-das-verticais-rural-produto-7C.pdf?MOD=AJPERES&CVID=m3rPopG>>. Acesso em: 29 jul 2021. Citado 2 vezes nas páginas 15 e 19.
- CHAUDHARI, B.; ZENNARO, M. *LPWAN Technologies for IoT and M2M Applications*. [S.l.: s.n.], 2020. Citado na página 23.
- CHEN, H.; GUAN, M.; LI, H. Air quality prediction based on integrated dual lstm model. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 9, p. 93285–93297, 2021. ISSN 21693536. Citado na página 41.
- CLERCQ, M. de; VATS, A.; BIEL, A. Agriculture 4.0 – the future of farming technology. *OliverWyman*, p. 16–16, 2018. Disponível em: <<https://www.oliverwyman.com/our-expertise/insights/2018/feb/agriculture-4-0--the-future-of-farming-technology.html>>. Citado na página 16.
- DIEBOLD, F. X. *Elements of forecasting*. [S.l.: s.n.], 2006. 366 p. ISBN 9780324359046. Citado 2 vezes nas páginas 39 e 40.
- DUAN, Y.; EDWARDS, J. S.; DWIVEDI, Y. K. Artificial intelligence for decision making in the era of big data – evolution, challenges and research agenda. *International Journal of Information Management*, Elsevier Ltd, v. 48, p. 63–71, 10 2019. ISSN 02684012. Citado na página 28.
- FAO. *Food and Agriculture Organization of the United Nations: Pesticides indicators*. 2019. Disponível em: <<http://www.fao.org/faostat/en/#data/EP/visualize>>. Acesso em: 31 jul 2021. Citado na página 15.
- FOKEN, T. *Micrometeorology*. [S.l.]: Springer-Verlag Berlin Heidelberg, 2017. 1-362 p. ISBN 9783642254406. Citado na página 20.
- GRAUPE, D. *Principles of artificial neural networks*. [S.l.: s.n.], 2013. Citado 2 vezes nas páginas 31 e 32.
- HAYKIN, S. *Redes Neurais: princípios e prática*. [S.l.: s.n.], 2008. Citado 4 vezes nas páginas 31, 32, 33 e 34.
- HAYKIN, S. S. *Neural networks and learning machines*. [S.l.]: Prentice Hall/Pearson, 2009. 906 p. ISBN 9780131471399. Citado na página 33.

- HENNAYAKE, K. M.; DINALANKARA, R.; MUDUNKOTUWA, D. Y. Machine learning based weather prediction model for short term weather prediction in sri lanka. In: . [S.l.]: Institute of Electrical and Electronics Engineers Inc., 2021. p. 299–304. ISBN 9781665444293. Citado na página 37.
- HEUSER, C. A. *Projeto de Banco de Dados*. [S.l.: s.n.], 2009. v. 6. Citado na página 26.
- HUPFER, P. *Unsere Umwelt: Das Klima*. [S.l.: s.n.], 1996. Citado 2 vezes nas páginas 20 e 21.
- IBGE. *PIB do setor agropecuário cresce 1,3% em 2019*. 2020. Disponível em: <<https://www.gov.br/pt-br/noticias/financas-impostos-e-gestao-publica/2020/03/pib-do-setor-agropecuario-cresce-1-3-em-2019>>. Acesso em: 20 jun 2021. Citado na página 15.
- IBM. *An introduction to deep learning*. 2020. Disponível em: <<https://developer.ibm.com/articles/an-introduction-to-deep-learning/>>. Acesso em: 29 ago 2021. Citado na página 29.
- IBM. *Neural Networks*. 2020. Disponível em: <<https://www.ibm.com/cloud/learn/neural-networks>>. Acesso em: 29 ago 2021. Citado na página 29.
- IBM. *What is an Application Programming Interface (API)*. 2020. Disponível em: <<https://www.ibm.com/cloud/learn/api>>. Acesso em: 19 jun 2021. Citado na página 24.
- ISMANGIL, D. et al. Managing the microclimate. *Spate Irrigation Network Foundation*, v. 27, 12 2016. Citado na página 21.
- KHOMP. *Módulo extensão e sensores climáticos usados no ITS 3G e no Endpoint LoRa para monitoramento de ambientes*. 2020. Disponível em: <<https://www.khomp.com/pt/produto/endpoint-lora-estacao-meteorologica>>. Acesso em: 20 jun 2021. Citado na página 23.
- LANTZ, B. *Machine learning with R*. [S.l.]: Packt Publishing, 2015. ISBN 9781784393908. Citado 2 vezes nas páginas 29 e 30.
- LAZZERI, F. Machine learning for time series forecasting with python. 2020. Citado na página 39.
- MONGODB. *NoSQL vs SQL Databases*. 2021. Disponível em: <<https://www.mongodb.com/nosql-explained/nosql-vs-sql>>. Acesso em: 15 set 2021. Citado na página 27.
- MOZILLA. *HTTP request methods*. 2021. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>>. Acesso em: 19 jun 2021. Citado na página 24.
- NIELSEN, A. *Practical time series analysis : prediction with statistics and machine learning*. [S.l.: s.n.], 2019. ISBN 9781492041658. Citado na página 39.
- ORACLE. *MAPE (Mean Absolute Percentage Error)*. 2022. Disponível em: <https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusu/insights_metrics_MAPE.html>. Acesso em: 31 jan 2022. Citado na página 41.
- REIS, E. A.; REIS, I. A. *Análise Descritiva de Dados*. 2002. Disponível em: <www.est.ufmg.br>. Citado na página 56.

- ROSENBLATT, F. *THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN* 1. 1958. 19-27 p. Citado na página 32.
- STACKOVERFLOW. *Stack Overflow Trends*. 2022. Disponível em: <<https://insights.stackoverflow.com/trends?tags=>>. Acesso em: 02 fev 2022. Citado na página 48.
- SUTTON, R.; BARTO, A. *Reinforcement Learning: An Introduction*. The MIT Press, 2020. Disponível em: <<http://incompleteideas.net/book/RLbook2020trimmed.pdf>>. Citado na página 29.
- TAULLI, T. *Artificial Intelligence Basics*. [S.l.]: Apress, 2019. Citado na página 28.
- TIWARI, S. Professional nosql. 2011. Citado na página 27.
- YANG, J. 2018. Disponível em: <<https://github.com/Kulbear/deep-learning-coursera/blob/master/SequenceModels/BuildingaRecurrentNeuralNetwork-StepbyStep-v2.ipynb>>. Acesso em: 24 fev 2022. Citado 2 vezes nas páginas 36 e 38.
- YU, Y.; CAO, J.; ZHU, J. An lstm short-term solar irradiance forecasting under complicated weather conditions. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 7, p. 145651–145666, 2019. ISSN 21693536. Citado na página 41.

Apêndices

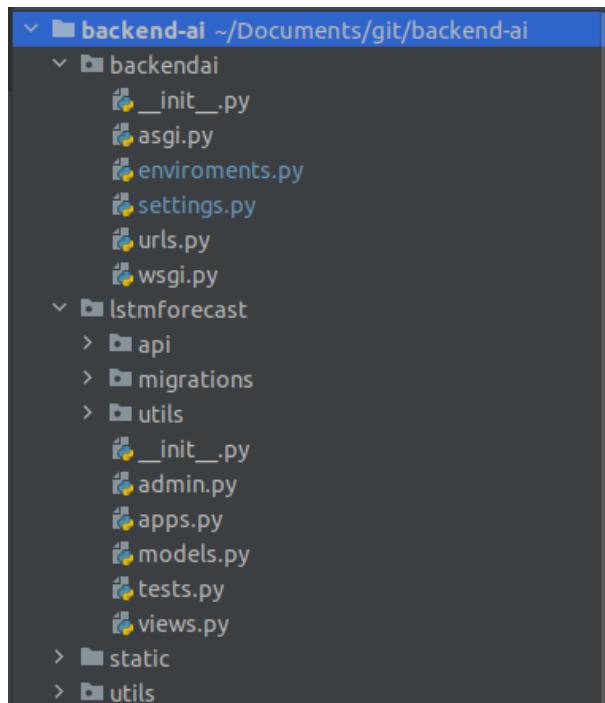
APÊNDICE A – TRECHO ENDPOINT

GET API PARA PREVISÃO LSTM

O *back-end* do sistema foi desenvolvido com Django *REST framework*, sendo que a estrutura do projeto seguiu o padrão estabelecido pela tecnologia conforme demonstrado na Figura 36. Todo o sistema *back* é composto por diversas classes e configurações complexas, para autenticação e acesso a banco de dados de forma segura confiável, porém a sua explanação detalhada não será realizada neste documento, pois este não é foco do trabalho. Por ser tratar de um futuro produto que poderá ser comercializado, não deve ultrapassar certo nível de transparência de explicação técnica, por outro lado, caso questionado seus métodos de implementação, o autor está completamente aberto para esclarecimento de dúvidas. Para acessar ao repositório privado do projeto basta clicar [aqui](#).

Na Código A.1 está descrito um trecho ao *endpoint* GET HTTP para a aquisição da previsão das variáveis, o qual será requerido pelo sistema *front-end*, ou seja, pelo próprio usuário.

Figura 36 – Estrutura *back-end* com Django *REST framework*



Fonte: O próprio autor.

Código A.1 – API *endpoint* para previsão

```

1 @action(detail=False, methods=['GET'])
2 def previsao_lstm(self, request, *args, **kwargs):
3
4     session = requests.Session()
5
6     # Requisição vinda do CORE
7     req_core = request.GET.dict()
8
9     end = req_core["timetoend"]
10    dev_id = req_core["dev_id"]
11    var = req_core["var"]
12    model = req_core["model"]
13    type_forecast = req_core["typeforecast"]
14
15    # Data inicio e final dos dados reais dependendo da previsão
16    n = 96
17    if type_forecast == "onehour":
18        n = 24
19
20    start = math.ceil(datetime.datetime.timestamp(datetime.datetime.
fromtimestamp(int(end)) - timedelta(hours=n)))
21
22    # Realiza requisição de dados reais na API IOT
23    lstm_service = LSTMservice()
24    start = str(start) + "000000"
25    end = str(end) + "000000"
26    uri = "?timetostart="+start+"&timetoend="+end+"&dev_id="+dev_id+"&var="+var
27    req_iot = session.get(API_IOT+uri)
28
29    req_iot_str = str(req_iot.json())
30    req_iot_str = req_iot_str.replace("\'", "\\"")
31
32    # Realiza a formatação dos dados para a previsão
33    real_data_format = lstm_service.format_real_data_to_forecast(req_iot_str,
var)
34
35    # Realiza previsão
36    lstm_forecast = LSTMforecast()
37    previsao = lstm_forecast.forecast(real_data_format, type_forecast, model,
var)
38
39    return Response(previsao, status=status.HTTP_200_OK)

```