

FACULDADE DE TECNOLOGIA DE SÃO PAULO
ITALO NASCIMENTO FREITAS

Internet of Things como Ferramenta de Gestão do Conhecimento: Exemplo com
Mineração de Dados

SÃO PAULO
2022

FACULDADE DE TECNOLOGIA DE SÃO PAULO
ITALO NASCIMENTO FREITAS

Internet of Things como Ferramenta de Gestão do Conhecimento: Exemplo com
Mineração de Dados

Trabalho submetido como exigência
parcial para a obtenção do Grau de
Tecnólogo em Análise e Desenvolvimento
de Sistemas
Orientador: Prof. Me. Valter Yogui

SÃO PAULO
2022

FACULDADE DE TECNOLOGIA DE SÃO PAULO

ITALO NASCIMENTO FREITAS

Internet of Things como Ferramenta de Gestão do Conhecimento: Exemplo com
Mineração de Dados

Trabalho submetido como exigência parcial para a obtenção do Grau de
Tecnólogo em Análise e Desenvolvimento de Sistemas.

Parecer do Professor Orientador **aprovado**

Conceito/Nota Final: 9,8

**Atesto o conteúdo contido na postagem do ambiente TEAMS pelo aluno e
assinada por mim para avaliação do TCC.**


Orientador: Prof. Me. Valter Yogui

SÃO PAULO, 11 de dezembro de 2022.

Assinatura do Orientador


Assinatura do aluno

Agradecimentos

Em primeiro lugar, agradeço a Deus por me ter me criado e me dado forças para chegar até este momento.

Agradeço a minha família pelo suporte material e emocional dado até aqui. Um agradecimento especial para meu irmão Gleyson, que me auxiliou bastante na montagem da parte física do meu projeto.

Agradeço ao professor Valter Yogui pelas orientações de como conduzir o tema do meu TCC e de se eu estava seguindo no caminho certo.

Agradeço também aos demais professores da Fatec pela contribuição que deram para eu alcançar o nível de conhecimento que possuo hoje.

Resumo

Este trabalho trata da Internet of Things como ferramenta de gestão do conhecimento, especialmente na criação de conhecimento. Ainda há lacunas de pesquisa sobre os *insights* nas características da criação de conhecimento dentro dessas aplicações e esse trabalho visa dar alguma contribuição para esclarecê-las. Para isso, escolheu-se a Mineração de Dados como metodologia e foi criado um cenário para extrair conhecimento a partir de dados coletados de Internet of Things: Prever a umidade do solo 12 horas no futuro de um conjunto de três plantas a partir da umidade do solo atual e de outras características ambientais. Ao fim, conseguiu-se essa previsão com uma taxa relativamente baixa de erros e alguns *insights* na criação de conhecimento dentro das aplicações de Internet of Things, embora haja várias limitações quanto aos modelos de previsão obtidos.

Palavras-chave: Gestão do conhecimento; Criação de conhecimento; Internet of Things; Internet das Coisas; Internet of Everything; Internet de Tudo; Mineração de Dados; Data Mining.

Abstract

This work deals with the Internet of Things as a knowledge management tool, especially in knowledge creation. There is still a research gap on the insights in the characteristics of knowledge creation within these applications and this work aims to make some contribution to clarify it. To achieve this, Data Mining was chosen as a methodology and a scenario was created to extract knowledge from data collected from Internet of Things: Predicting soil moisture 12 hours in the future of a set of three plants from current soil moisture and other environmental characteristics. This prediction was achieved with a relatively low rate of errors and some insights in knowledge creation from Data Mining in Internet of Things applications, although there are several limitations regarding the prediction models obtained.

Keywords: Knowledge Management; Knowledge Creation; Internet of Things; Internet of Everything; Data Mining.

Sumário

Introdução	11
1 Gestão e Criação do Conhecimento	12
1.1 Conceito de Gestão do Conhecimento	12
1.2 Processos da Gestão do Conhecimento	13
1.3 Conhecimento, dado e informação	14
1.4 Criação do Conhecimento.....	15
2 Internet of Things e Internet of Everything.....	18
2.1 Conceito de Internet of Things	18
2.2 Conceito de Internet of Everything	19
2.3 Sensores e controladores	20
2.4 Big Data	21
2.5 <i>Cloud, Fog e Edge Computing</i>	21
3 Criação de Conhecimento usando IoT e IoE.....	23
3.1 Contexto.....	23
3.2 As cinco categorias de análise de Internet of Things	23
3.3 Dimensão do conhecimento relativa aos sensores de Internet of Everything	24
3.4 Modelo de Gestão do Conhecimento integrado à Internet of Everything	27
4 Mineração de Dados	31
4.1 O que é	31
4.2 Processos	32
4.3 Preparação dos dados	33
4.4 Técnicas.....	33
4.5 Algoritmos de Regressão	35
4.5.1 Regressão Linear.....	35
4.5.2 Árvores de Decisão.....	35
4.5.3 Processos Gaussianos	37
4.5.4 Redes Neurais Artificiais	37
4.5.5 <i>Support Vector Machines</i>	38
4.6 Métricas estatísticas de erro	39
4.7 Mineração de Dados, Criação do Conhecimento e Internet of Everything	40
5 Projeto.....	41
5.1 Descrição Física.....	41
5.1.1 Coisas	42
5.1.2 Sensores.....	43
5.1.3 Controlador	45
5.1.4 Computador	46
5.1.5 Nuvem.....	47
5.2 Mineração de Dados	48
5.2.1 Preparação dos dados	48
5.2.2 Modelagem	54
5.3 Resultados	55
5.3.1 Caso A	55
5.3.2 Caso B	56
5.3.3 Caso C	59
5.4 Discussão.....	59
5.5 Limitações	61
Conclusão	63
Referências	64
Anexos	70

Anexo 1 - Script usado no Arduino.....	71
Anexo 2 - Script Python para gravar dados enviados pelo Arduino	74
Anexo 3 - Script Python para enviar os dados para a nuvem	76
Anexo 4 - Script Python para agregar arquivos com dados do mesmo dia.....	77
Anexo 5 - Script Python para detectar erros nos arquivos	79
Anexo 6 - Script Python para remover atributos de controle	81
Anexo 7 - Script Python para reduzir leituras de segundos para leituras de minuto	82
Anexo 8 - Script Python para reduzir leituras de minutos para leituras de hora....	85
Anexo 9 - Script Python para unificar as instâncias em um único <i>dataset</i>	88
Anexo 10 - Script Python para adicionar médias e desvios-padrão das 12 horas anteriores	89
Anexo 11 - Script para adicionar a classe	93
Anexo 12 - Amostra do <i>dataset</i> de treino da planta 1	96
Anexo 13 - Amostra do <i>dataset</i> de testes da planta 1	97
Anexo 14 - Amostra do <i>dataset</i> de treino da planta 2	98
Anexo 15 - Amostra do <i>dataset</i> de testes da planta 2.....	99
Anexo 16 - Amostra do <i>dataset</i> de treino da planta 3	100
Anexo 17 - Amostra do <i>dataset</i> de teste da planta 3.....	101

Lista de Ilustrações

Figuras

Figura 1 - Espiral do Modelo SECI	16
Figura 2 - Representação da Internet of Everything	20
Figura 3 - Dimensão do conhecimento relativa aos sensores de Internet of Everything	25
Figura 4 - Modelo de Gestão do Conhecimento integrado à Internet of Everything ..	28
Figura 5 - Representação de uma regressão linear	35
Figura 6 - Exemplo de árvore de decisão	36
Figura 7 - Exemplos de processos gaussianos	37
Figura 8 - Exemplo de Rede Neural Artificial.....	37
Figura 9 - Exemplo de <i>Support Vector Machine</i>	38
Figura 10 - Representação do projeto	41
Figura 11 - Planta 1	42
Figura 12 - Planta 2	42
Figura 13 - Planta 3	43
Figura 14 - Árvore de Decisão criada pelo algoritmo <i>M5P</i> para a planta 2	57

Quadros

Quadro 1 - Categorias de análise de Internet of Things aplicadas às técnicas de Mineração de Dados	40
Quadro 2 - Características dos sensores	43
Quadro 3 - Cabeçalhos após a execução da sétima etapa	52
Quadro 4 - Cabeçalhos após a execução da oitava etapa	53
Quadro 5 - Características dos <i>datasets</i> de treinamento e de teste.....	53
Quadro 6 - Regressões lineares de cada folha da árvore de decisão do algoritmo <i>M5P</i>	58

Lista de Tabelas

Tabela 1 - Intervalos aceitáveis para cada grandeza	44
Tabela 2 - Aplicação dos algoritmos para escolha do melhor e definição da linha de base do Caso A	55
Tabela 3 - Desempate para escolher qual algoritmo otimizar para o Caso A	55
Tabela 4 - Otimização do algoritmo <i>M5P</i> para o Caso A	55
Tabela 5 - Aplicação dos algoritmos para escolha do melhor e definição da linha de base do Caso B	56
Tabela 6 - Otimização do algoritmo <i>M5P</i> para o Caso B	56
Tabela 7 - Aplicação dos algoritmos para escolha do melhor e definição da linha de base do Caso C	59
Tabela 8 - Otimização do algoritmo <i>RandomForest</i> para o Caso C	59

Introdução

No mundo atual, em constante mutação, a gestão do conhecimento se mostra fundamental para que a empresa responda adequadamente às mudanças no ambiente. Entre as diversas ferramentas capazes de auxiliar esse processo, estão a Internet of Things e a Internet of Everything, que contribuem principalmente na etapa de criação do conhecimento.

Viviane Cunha Farias da Costa identificou uma lacuna de pesquisa na criação de conhecimento dentro da Internet of Everything (COSTA, 2022, p. 141), sobre os *insights* que dizem respeito às características da criação do conhecimento nesse tipo de aplicação. Dando sua contribuição para solucionar essa lacuna, Costa desenvolveu alguns artefatos (2022, p. 11), entre eles, uma taxonomia para classificar os objetos de Internet of Everything e um modelo que integra Gestão do Conhecimento e Internet of Everything.

Este trabalho visa dar uma contribuição na solução da lacuna identificada por Costa (2022, p. 141), buscando *insights* na criação de conhecimento dentro da Internet of Everything quando se usa o processo de Mineração de Dados.

Para isso, os artefatos desenvolvidos por Costa (2022, p. 77-101) foram aplicados à Mineração de Dados e foi desenvolvido um projeto prático. O projeto consiste em alguns sensores que coletam a umidade do solo de algumas plantas e outros dados sobre o ambiente. Usando a Mineração de Dados, busca-se prever, a partir desses dados a umidade do solo, de cada planta após 12 horas, com uma baixa taxa de erros.

Esse trabalho está estruturado da seguinte forma: O primeiro capítulo apresenta os conceitos de Gestão do Conhecimento e Criação do Conhecimento. O segundo apresenta os conceitos de Internet of Things e Internet of Everything. O terceiro mostra como Gestão do Conhecimento e Internet of Everything podem ser combinados. O quarto explica a Mineração de Dados e contextualiza seu uso dentro da criação do conhecimento e Internet of Everything. O quinto capítulo é o projeto que explora, de forma prática, os conceitos apresentados no trabalho.

1 Gestão e Criação do Conhecimento

1.1 Conceito de Gestão do Conhecimento

O mundo atual caracteriza-se pela mudança constante, em razão dos contínuos avanços tecnológicos, da constante pressão do mercado por inovação, dos conflitos geopolíticos e das mudanças na sociedade. Para sobreviver nesse cenário, a organização precisa usar o conhecimento de forma estratégica. Segundo Nonaka e Takeuchi (2008a, p. IX), o conhecimento perpetua as mudanças dentro da organização quando incorporado aos produtos, serviços, tecnologias e processos. Da necessidade de gerir esse recurso, surge a gestão do conhecimento.

A gestão do conhecimento é o conjunto dos processos usados por uma organização para garantir que o conhecimento gere valor. Envolve todos os processos de criar, reter, capturar, organizar e disseminar o conhecimento (SAITO, 2017).

Entre os benefícios que a Gestão do Conhecimento consegue oferecer, pode-se citar (MARQUES JÚNIOR, 2017):

- Aprendizagem com os próprios erros para não sejam repetidos no futuro;
- Compartilhamento de ideias;
- Condução de melhores relações com os clientes;
- Operações mais efetivas;
- Maior capacidade de inovação.

A gestão do conhecimento é uma área recente. Foi criada em 1995, com o lançamento do livro “A Empresa Criadora de Conhecimento”, de Nonaka e Takeuchi. Outros livros, como “Capital intelectual: a nova riqueza das organizações”, também alimentaram a discussão (SAITO, 2017).

Durante o início da década de 2000, a gestão do conhecimento passou por um período bastante aquecido. Uma grande quantidade de programas internos, consultorias e plataformas foram criados com foco na gestão do conhecimento. Porém, com a dificuldade de mostrar o retorno do investimento, muitos dos programas foram descontinuados (SAITO, 2017).

No início da década de 2010, houve uma retomada nesses programas. As empresas que perseveraram e amadureceram na gestão do conhecimento se tornaram referência para as que desistiram ou que não haviam iniciado seus programas de implantação (SAITO, 2017).

Hoje, há duas tendências na gestão do conhecimento, uma na tecnologia e outra nas pessoas. A tendência na tecnologia é o desenvolvimento da inteligência artificial, que provoca muito impacto em como o conhecimento é criado e usado. A tendência nas pessoas são as redes sociais e a colaboração em rede, que permitem que as pessoas se comuniquem e compartilhem conhecimento em uma escala antes impossível (SAITO, 2017).

1.2 Processos da Gestão do Conhecimento

Com base em Prat (2011 apud COSTA, 2022, p. 89), é possível dividir a Gestão do Conhecimento em duas categorias de processo: Estratégicos e Operacionais.

Os processos estratégicos são:

- **Identificação:** Reconhecimento, mapeamento e modelagem do conhecimento atual ou necessário para alcançar objetivos. (COSTA, 2022, p. 97)
- **Avaliação:** Verificação da adequação do conhecimento às necessidades da organização. Pode ser realizada ao nível do conhecimento, ao nível dos projetos e sistemas de gestão do conhecimento e ao nível de estratégia de gestão do conhecimento (COSTA, 2022, p. 97)
- **Atualização:** Modificação do conhecimento para que ele permaneça adequado às necessidades presentes da organização. (COSTA, 2022, p. 98)
- **Proteção:** Resguardo do conhecimento para que não seja usado pelos concorrentes. Pode envolver o uso de patentes e mecanismos de segurança da informação, como firewalls. (COSTA, 2022, p. 98)

Os processos operacionais são:

- **Aquisição:** Atividades que aumentam a quantidade de conhecimento existente na organização. (COSTA, 2022, p. 95)
- **Armazenamento:** Retenção do conhecimento na memória individual e coletiva. (COSTA, 2022, p. 99)
- **Transferência:** Compartilhamento do conhecimento entre os indivíduos, grupos e organizações. (COSTA, 2022, p. 99)
- **Utilização:** Aplicação do conhecimento. (COSTA, 2022, p. 95)

1.3 Conhecimento, dado e informação

O conhecimento é uma crença verdadeira justificada (NONAKA; TAKEUCHI, 2008c, p. 56): “crença”, pois o indivíduo acredita no modelo mental que possui; “verdadeira”, pois corresponde à realidade e/ou é coerente com outros modelos mentais que o indivíduo possui; “justificada”, pois o indivíduo tem razões para acreditar que é verdadeira. Além disso, o conhecimento está sempre relacionado com a ação humana e é específico ao contexto e à situação (NONAKA; TAKEUCHI, 2008c, p. 57).

O conhecimento não se confunde com dados nem com informação, embora esteja em estrita relação com eles.

Dados são sequências quantificáveis de símbolos. “Quantificável” significa as sequências que podem ser reproduzidas sem que haja distinção perceptível entre o original e o reproduzido. Os dados são entidades matemáticas, sintáticas e destituídas de significado próprio. Textos, figuras, áudios e vídeos são exemplos de dados (SETZER, 2015).

Informações são abstrações informais (ou seja, que não podem ser formalizadas por uma linguagem lógica ou matemática) da mente de uma pessoa e que carregam significado com relação a algo (SETZER, 2015). “O céu é azul” é um exemplo de informação, caso o indivíduo que leia ou escute essa frase relacione “céu” com o “espaço onde se observam os astros” e “azul” com a cor correspondente a essa palavra.

Dado se distingue de informação porque o primeiro é puramente sintático, enquanto a segunda contém necessariamente semântica; por isso, embora a

informação pode ser representada por meio de dados, não pode ser processada, já que o computador é uma máquina puramente sintática (SETZER, 2015). A informação também se distingue do conhecimento porque, enquanto a primeira envolve apenas o aspecto semântico, a segunda acrescenta a noção de crença do indivíduo e de ser voltada para uma ação. (NONAKA; TAKEUCHI, 2008c, p. 56)

1.4 Criação do Conhecimento

Nonaka e Takeuchi (2008c, p. 55) desenvolveram uma teoria da criação do conhecimento com base em duas dimensões que este possui: a ontológica e a epistemológica.

A dimensão ontológica diz respeito ao nível de distribuição que o conhecimento atinge: individual, grupal, organizacional e interorganizacional (NONAKA; TAKEUCHI, 2008c, p. 57). O conhecimento é criado apenas pelos indivíduos, mas a criação de conhecimento se torna mais intensa ao avançar nos níveis ontológicos.

Quanto à dimensão epistemológica, o conhecimento possui dois tipos (NONAKA; TAKEUCHI, 2008b, p. 19):

- **Conhecimento Explícito:** É o conhecimento que pode ser expresso em palavras, números ou sons e transmitido aos indivíduos de maneira formal e sistemática.
- **Conhecimento Tácito:** É o conhecimento pessoal, que está dentro da mente do indivíduo e, por isso, é difícil de expressar. Tem dois aspectos: O técnico, relacionado a como fazer as coisas (know-how); e o cognitivo, ligado às crenças, percepções, valores e emoções.

A chave para criação do conhecimento é a interação contínua e dinâmica entre o conhecimento explícito e tácito (NONAKA; TAKEUCHI, 2008c, p. 69). Essa interação acontece de quatro modos, que formam um ciclo conhecido como Modelo SECI (NONAKA; TAKEUCHI, 2008b, p. 23): Socialização, Externalização, Combinação e Internalização. Esse ciclo está representado na Figura 1.

Figura 1 - Espiral do Modelo SECI



Fonte: Adaptado de Nonaka e Takeuchi (2008c, p. 69)

A socialização é a conversão de conhecimento tácito para tácito. Ocorre através da troca de experiências pela observação, imitação e prática. Um exemplo é quando o aprendiz aprende com seu mestre: Ele não aprende apenas pela linguagem, mas também por observar e imitar o comportamento de seu mestre (NONAKA; TAKEUCHI, 2008c, p. 60-61).

A externalização é a conversão de conhecimento tácito para explícito. Ocorre através do diálogo e da reflexão, pelos quais o indivíduo conceitua suas imagens internas servindo-se de induções, deduções, metáforas e analogias. Um exemplo é da força-tarefa da minicopiadora da Canon, que fez uma analogia de uma lata de cerveja com um tambor de impressora; eles buscavam um material barato e confiável para os tambores da minicopiadora e, com base nessa analogia, escolheram o alumínio. É a mais importante para a criação do conhecimento, pois durante esse processo, conceitos são criados para descrever o conhecimento já obtido de forma tácita (NONAKA; TAKEUCHI, 2008c, p. 62-64).

A combinação é a conversão de conhecimento explícito para explícito. Ocorre através da sistematização do conhecimento, quando há troca de conhecimento por documentos, conversas, reuniões, palestras e análises de dados, especialmente entre profissionais de áreas distintas. O principal exemplo é a educação formal realizada nas escolas (NONAKA; TAKEUCHI, 2008c, p. 65-66).

A internalização é a conversão de conhecimento explícito para tácito. Ocorre por meio da aquisição do conhecimento, quando o indivíduo “aprende fazendo” ou revive mentalmente experiências dos outros. Um exemplo é quando o indivíduo absorve o conteúdo de documentos ou manuais (NONAKA; TAKEUCHI, 2008c, p. 67-68).

Esses quatro modos progridem nos níveis ontológicos, formando um ciclo (NONAKA; TAKEUCHI, 2008b, p. 23):

- **Socialização:** de indivíduo para indivíduo;
- **Externalização:** do indivíduo para o grupo;
- **Combinação:** do grupo para a organização;
- **Internalização:** da organização para o indivíduo.

A socialização cria um campo de interação, que permite aos indivíduos compartilharem suas experiências e modelos mentais. Em segundo lugar, a externalização se sucede pelo diálogo e pela reflexão entre os membros do grupo. Em seguida, a combinação é desencadeada pela “rede” do conhecimento recentemente criado e do conhecimento de outros departamentos da organização. Por fim, a internalização ocorre pelo “aprender fazendo” e o ciclo se repete (NONAKA; TAKEUCHI, 2008c, p. 69).

2 Internet of Things e Internet of Everything

2.1 Conceito de Internet of Things

A internet mostrou-se uma tecnologia revolucionária desde que se popularizou. Graças a essa tecnologia, computadores de qualquer lugar do mundo podem se conectar e transmitir dados. Mas é possível conectar à internet não apenas computadores, mas qualquer tipo de objeto?

A Internet of Things (IoT), ou Internet das Coisas, em português, é uma resposta afirmativa para essa pergunta. Ela pode ser conceituada como uma “rede” de coisas conectadas ao redor do mundo (COSTA, 2022, p. 1). Essas coisas vão além dos meros dispositivos eletrônicos: incluem comida, roupas, móveis, monumentos, obras de arte e até seres vivos, como plantas e animais. Praticamente qualquer objeto físico distinguível no mundo real pode ser conectado à internet (KOSMATOS; TSELIKAS; BOUCOUVALAS, 2011, p. 6).

O conceito de Internet of Things foi proposto pela primeira vez por Kevin Ashton em 1999. Contudo, estudos relacionados ocorriam desde os anos 1980. Um dos primeiros exemplos é uma máquina de vender Coca-Cola que ficava a alguns andares de distância dos programadores. Os programadores conectaram a máquina à internet e, por meio de alguns programas instalados nela, conseguiam descobrir remotamente seu estado. Em 2003, o conceito se tornou bastante popular, graças aos esforços do Auto-ID Center (MADAKAM; RAMASWAMY; TRIPATHI, 2015, p. 166).

A Internet of Things tem aplicações em diversas áreas. Uma delas são as *Smart Homes* ou Casas Inteligentes, que são sistemas e dispositivos que automatizam várias tarefas domésticas, oferecendo conforto, segurança, saúde e economia de energia aos moradores (SALMERÓN-MANZANO et al., 2021, p. 2). Por exemplo, um aquecedor ajusta automaticamente a temperatura para manter o ambiente confortável nos dias de frio; uma câmera de segurança envia um alerta no celular do dono da casa quando percebe uma movimentação estranha; uma geladeira envia um e-mail quando percebe a falta de algum produto; uma lâmpada ajusta automaticamente a luminosidade com base na luz do sol recebida por uma janela.

Uma outra área de aplicação são as indústrias, onde é usado o termo *IloT* (*Industrial Internet of Things*). A *IloT* mecaniza objetos inteligentes para detectar, coletar, sentir, manipular e comunicar eventos em tempo real dentro dos sistemas

industriais, visando alcançar alta produtividade e uma gestão otimizada dos recursos (AHMED et al., 2022). Um exemplo é o uso de sensores para medir a “saúde” de um motor e enviar alertas quando o motor não está funcionando bem ou prever quando será necessário realizar sua manutenção ou substituí-lo.

Uma última área, entre muitas outras, é a *Smart Farming*. Ela usa Internet of Things e outras tecnologias para vencer o desafio de alimentar uma população mundial em crescimento, minimizando o desperdício e aumentando a produtividade. Um exemplo é a utilização de sensores para controlar o uso de pesticidas, monitorar a colheita e gerir o solo (NAVARRO; COSTA; PEREIRA, 2020).

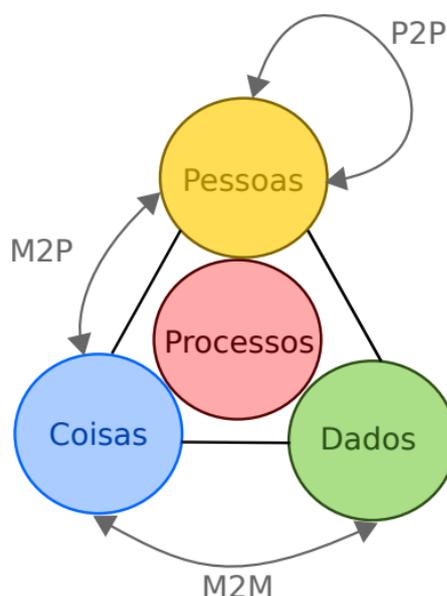
Desse modo, a Internet of Things segue crescendo. Entre 2008 e 2009, o número de objetos conectados ultrapassou a população mundial (EVANS, 2011, p. 3). Em 2021, havia cerca de 12,2 bilhões de dispositivos conectados. A expectativa é de que, em 2025, haja cerca de 25 bilhões de dispositivos ativos (HASAN, 2022),

2.2 Conceito de Internet of Everything

A Internet of Everything (IoE), ou Internet de Tudo em português, é uma expansão do conceito da Internet of Things e considera não apenas as “coisas” como participantes da internet, mas também as pessoas, os processos e os dados (EVANS, 2012). As pessoas são os seres humanos que utilizam a internet. Os dados são as representações de informações trocadas entre pessoas e coisas. Os processos são todas as trocas de dados entre pessoas e coisas, podendo ser de Máquina a Máquina (M2M), de Máquina a Pessoa (M2P ou P2M) ou de Pessoa a Pessoa (P2P). Uma representação da Internet of Everything é feita na Figura 2.

O conceito de Internet of Things foca na camada de rede, onde as coisas se conectam e se comunicam para entregar serviços aos usuários finais (COSTA, 2022, p. 15), limitando-se às conexões Máquina a Máquina (M2M). Já a Internet of Everything representa todas as conexões de rede e fluxos de dados em tempo real entre as pessoas e coisas; o resultado é inteligência e *insights* em tempo real, bem além do contexto de IoT (COSTA, 2022, p. 19).

Figura 2 - Representação da Internet of Everything



Fonte: Adaptado de Evans (2012)

Graças à Internet of Everything, a Internet se transforma de uma rede destinada apenas à comunicação para uma rede de controle enxertada diretamente no mundo físico (COSTA, 2022, p. 4).

2.3 Sensores e controladores

Entre as tecnologias que viabilizam a Internet of Things e a Internet of Everything, estão os sensores e os controladores.

Os sensores são objetos que convertem sinais do mundo físico em dados que podem ser processados por um computador. Eles fazem a ponte entre os objetos do mundo real e os objetos do mundo virtual (RAYES; SALAM, 2017 apud COSTA, 2022, p. 29). É por meio de um sensor conhecido como RFID, por exemplo, que objetos não eletrônicos podem ser identificados e conectados à Internet of Things (AGGARWAL; LAL DAS, 2012 apud MADAKAM; RAMASWAMY; TRIPATHI, 2015, p. 169). Outros exemplos de sensores incluem sensores de temperatura, umidade, movimento, luz infravermelha etc.

Os controladores são dispositivos responsáveis por receber os dados dos sensores e oferecer uma conexão de internet. Podem ter a capacidade de tomar decisões de forma autônoma e de enviar os dados para um computador para que este realize a análise. Um exemplo de controlador é o Arduino, que é uma plataforma eletrônica *open-source* que visa facilitar ao máximo a construção de soluções que envolvem hardware e software. O Arduino pode receber como input os dados de

diversos tipos de sensores e transformar isso em output, que pode ser uma ação como ligar um LED ou emitir um som. O Arduino aceita uma série de instruções de programação em uma linguagem semelhante à linguagem C (WHAT IS ARDUINO, 2018).

2.4 Big Data

Uma das consequências do uso de Internet of Things e Internet of Everything é o grande volume de dados gerados, graças aos bilhões de objetos conectados em tempo real. Um problema que surge disso é a “Big Data”.

Big Data são todos os dados que são “grandes demais”, “rápidos demais” ou “difíceis demais” para serem processados por ferramentas comuns. “Grandes demais” significa dados em uma escala gigantesca, que ultrapassam os *petabytes*. “Rápidos demais” significa a necessidade de processar muito rapidamente, como quando uma fraude do cartão de crédito está sendo verificada. “Difícil demais” significa que os dados têm características demasiado específicas que não podem ser processadas pelas ferramentas existentes. (MADDEN, 2012, p. 4)

2.5 Cloud, Fog e Edge Computing

Outras tecnologias que viabilizam a Internet of Things e a Internet of Everything são os diferentes níveis de computação: *Cloud Computing*, *Fog Computing* e *Edge Computing*.

A *Cloud Computing* (ou Computação em Nuvem, em português) é uma forma de computação onde os recursos de computação como processamento, armazenamento, aplicações etc. são obtidos pela internet sob demanda, também conhecida como *pay-as-you-go* (MEI; BLAKE, 2010 apud QI; TAO, 2019). Por oferecer serviços de alta qualidade a baixo custo, atraiu enorme atenção da indústria e da academia e muitas empresas como Amazon, Google e Microsoft, IBM e Alibaba passaram a vender esse tipo de serviço.

Contudo, enviar e receber dados pela internet causa uma latência que pode ser problemática quando é necessário ter respostas em tempo real. Além disso, há o custo pela grande quantidade de dados enviados e o risco de queda quando há falhas no acesso à internet. Por isso, surgiram as soluções intermediárias de *Fog Computing* e *Edge Computing* (QI; TAO, 2019).

Na *Fog Computing* (ou Computação em Névoa, em português), os dados são processados em um computador na rede local, próximo das fontes dos dados. Já na *Edge Computing* (ou Computação de Borda, em português), os dados são processados já no sensor ou dispositivo, sem realizar transferências (BOUACHIR et. al., 2020). Assim, os problemas apontados na computação em nuvem são mitigados.

3 Criação de Conhecimento usando IoT e IoE

3.1 Contexto

Com a expansão da Internet of Things e da Internet of Everything, o mundo se torna cada vez mais conectado e inteligente. Isso oferece uma quantidade imensa de dados que a gestão de conhecimento pode usar na criação de conhecimento (SHEN; NEWSHAM; GUNAY, 2017 apud COSTA, 2022, p. 32). Além disso, a Internet of Things e a Internet of Everything expandem enormemente a capacidade de reagir às mudanças, que é um dos objetivos centrais da gestão do conhecimento. Por isso, a Internet of Things e a Internet of Everything mostram-se ferramentas valiosas para a gestão do conhecimento.

Há várias pesquisas que relacionam Internet of Things, Internet of Everything e Gestão do Conhecimento, como mostram Ferreira et al. (2018, p. 104) e Costa (2022, p. 32). Siow, Tiropanis e Hall (2018, p. 13) também apresentam cinco categorias para as análises possíveis nos dados de IoT e que podem participar da gestão do conhecimento. Entretanto, como mencionado na introdução, Costa notou uma lacuna de pesquisa sobre os *insights* que dizem respeito às características da criação do conhecimento (2022, p. 141) e para contribuir na solução dessa lacuna, desenvolveu alguns artefatos, entre eles, uma taxonomia para classificar os objetos de Internet of Everything e um modelo que integra Gestão do Conhecimento e Internet of Everything.

As seções seguintes apresentarão as categorias descritas por Siow, Tiropanis e Hall e alguns artefatos desenvolvidos por Costa.

3.2 As cinco categorias de análise de Internet of Things

Segundo Siow, Tiropanis e Hall (2018, p. 2) a análise pode ser descrita como o processo de derivar conhecimento e *insights* a partir dos dados. Esses autores apresentam, no contexto da Internet of Things, cinco categorias de análise: Descrição, Diagnóstico, Descoberta, Predição e Prescrição. (SIOW; TIROPANIS; HALL, 2018, p. 13)

A Descrição resume e apresenta os dados brutos de Internet of Things que foram coletados. Ela responde à pergunta “O que está acontecendo?”. Isso normalmente é feito por meio de gráficos, relatórios e estatísticas.

O Diagnóstico explica os dados brutos coletados. Ela responde à pergunta “Por que aconteceu?”, indo um passo além da descrição.

A Descoberta encontra informações não triviais dentro dos dados. Ela responde à pergunta “Que coisas interessantes aconteceram e não sabemos?”. Ela diferencia-se das anteriores porque tenta descobrir coisas novas, em vez de apenas descrever ou diagnosticar o que aconteceu.

A Predição prevê resultados futuros com suas probabilidades. Ela responde à pergunta “O que provavelmente acontecerá?”.

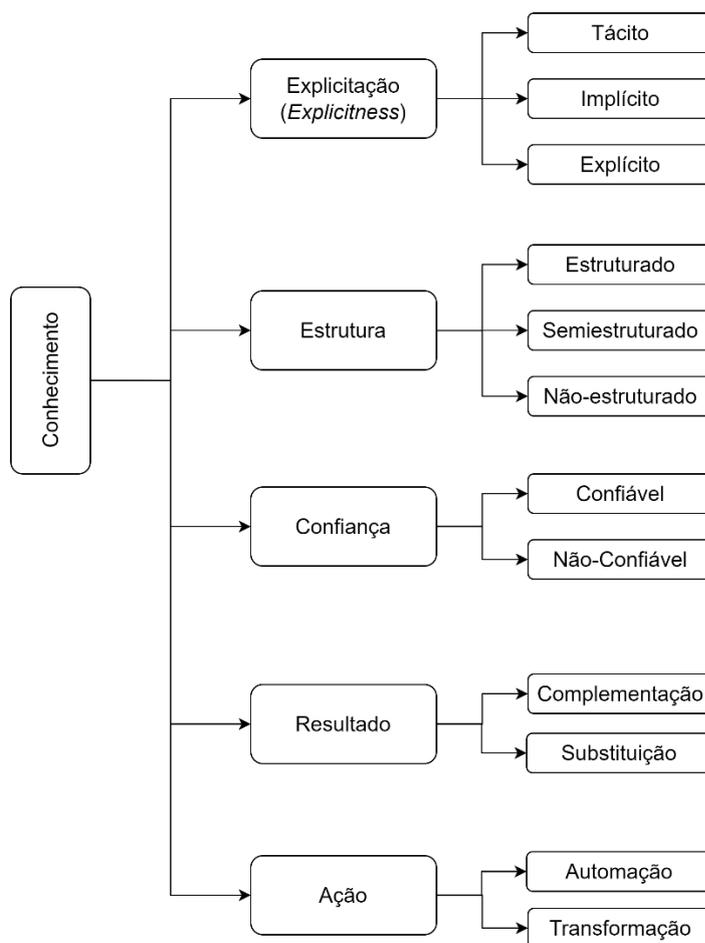
A Prescrição sugere as melhores ações a serem tomadas com base nos resultados previstos. Ela responde à pergunta “O que devemos fazer?”.

3.3 Dimensão do conhecimento relativa aos sensores de Internet of Everything

Costa (2022, p. 77-83) desenvolveu uma taxonomia para a classificação dos sensores em Internet of Everything, com quatro dimensões principais: Conhecimento, Tipo, Observação e Potencialidade. Para o propósito deste trabalho, que enfatiza a criação de conhecimento, apenas a dimensão relativa ao conhecimento será apresentada. Sua representação está na Figura 3.

Em sua classificação, Costa (2022, p. 32) considera como sensores não apenas os dispositivos eletrônicos, mas também os seres humanos, já que participam da Internet of Everything e podem extrair dados do mundo físico.

Figura 3 - Dimensão do conhecimento relativa aos sensores de Internet of Everything



Fonte: Adaptado de Costa (2022, p. 62)

A dimensão relativa ao conhecimento se subdivide em:

- **Explicitação (*Explicitness*):** Diz respeito às características do conhecimento. Equivale à dimensão epistemológica do conhecimento de Nonaka e Takeuchi (2008b, p. 19), mas com uma categoria adicional de “Implícito”.
 - *Tácito*: É o conhecimento interno das pessoas, difícil de ser comunicado. Pertence apenas aos sensores humanos.
 - *Implícito*: É o “conhecimento” interno dos objetos, as informações que estão entrelaçadas nos sistemas e fontes de dados. Ele emerge por meio da Inteligência Artificial.

- *Explícito*: É o conhecimento codificado e articulado, após ser externalizado do conhecimento tácito ou extraído do conhecimento implícito.
- **Estrutura**: Refere-se à organização dos dados que compõem o conhecimento.
 - *Estruturado*: São os dados organizados em um formato bem definido. Normalmente, são organizados em uma estrutura relacional similar a uma tabela, de forma a serem consultados por meio de SQL. Exemplos incluem números, datas e cadeias de caracteres (*strings*, em inglês).
 - *Semiestruturado*: São os dados que ainda são organizados, porém de forma menos rígida que os dados estruturados. Normalmente, podem ser consultados por meio de NoSQL. Exemplos incluem arquivos JSON e XML.
 - *Não-Estruturado*: Dados sem nenhuma organização definida. Exemplos incluem imagens, áudio, vídeo etc.
- **Confiança**: Diz respeito à veracidade dos dados.
 - *Confiável*: Dados certos e completos.
 - *Não-Confiável*: Dados com erros, incertos e/ou incompletos.
- **Resultado**: Diz o valor do conhecimento que os dados do sensor permitem trazer.
 - *Complementação*: Dados compartilhados que apenas ajudam a adicionar novas informações. Um exemplo é a combinação de sensores em uma *Smart Home* para descrever o estado da casa: Em quais cômodos há pessoas, em quais a luz está acesa, qual a temperatura de cada cômodo etc.
 - *Substituição*: Dados que permitem gerar informação disruptiva. Um exemplo são os sensores utilizados no corpo humano e cujos dados permitem detectar alguma doença,

- **Ação:** Diz qual o objetivo a ser alcançado por meio do sensor.
 - *Automação:* Realização de processos que reduzem a necessidade de participação humana. Um exemplo é um sensor de luz ultravioleta que detecta defeitos nas peças de uma linha de montagem e aciona uma máquina para removê-las da linha.
 - *Transformação:* Uso dos sensores para mudar o estado do ambiente servindo-se de pouca ou nenhuma participação humana. Um exemplo são os sensores de uma casa que, detectando a presença de uma pessoa, já acendem a luz e ativam um ar-condicionado para regular a temperatura do local; detectando a ausência, enviam dados para que a luz seja apagada e para que o ar-condicionado entre no estado de economia de energia.

3.4 Modelo de Gestão do Conhecimento integrado à Internet of Everything

Costa (2022, p. 89-101) desenvolveu um modelo de Gestão do Conhecimento integrado à Internet of Everything com base na taxonomia dos processos de Gestão do Conhecimento de Prat (PRAT, 2011 apud COSTA, 2022, p. 89), no modelo SECI (já apresentado na seção 1.4 deste trabalho) e o modelo ERIS. Aqui, será apresentada uma adaptação do modelo de Costa.

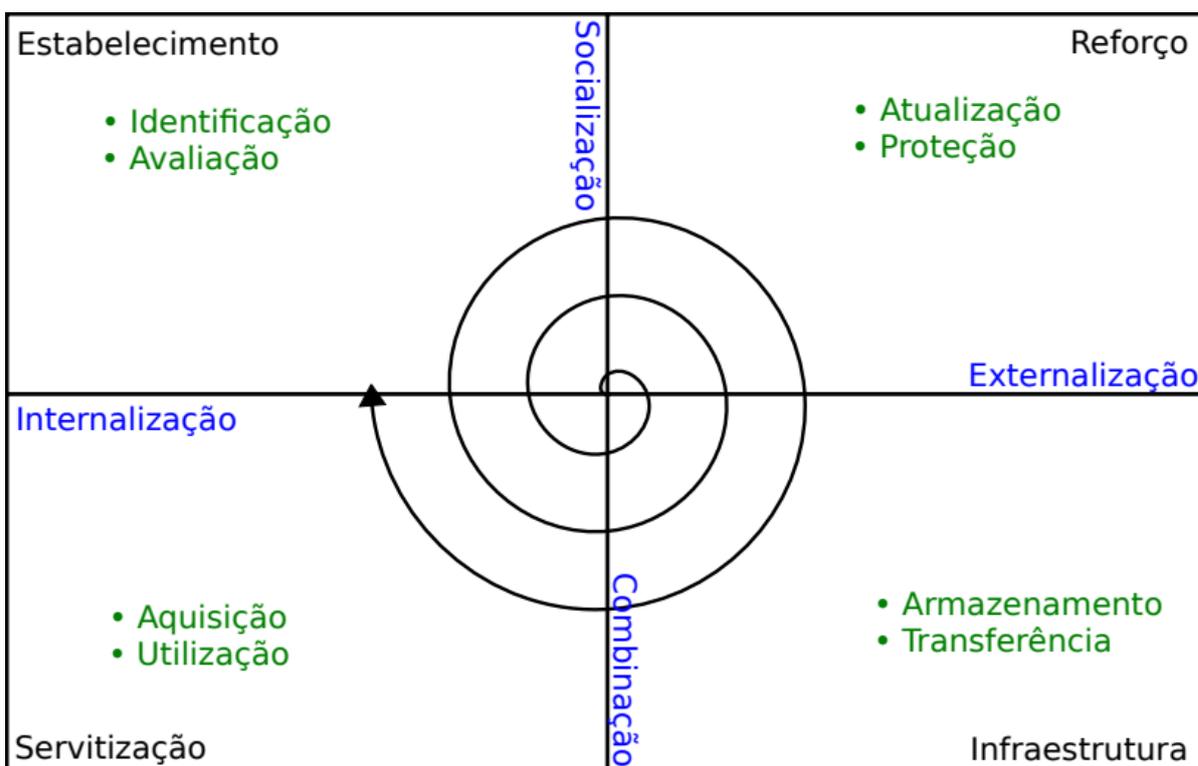
O modelo ERIS (ou SERI) é um modelo de ciclo de evolução de serviços desenvolvido por Hyunsoo Kim (COSTA, 2022, p. 93). É dividido em quatro fases: Estabelecimento (*Establishment*), Reforço (*Reinforcement*), Infraestrutura (*Infrastructure*) e Servitização (*Servitization*). Normalmente, se um serviço é centrado em elementos intangíveis (como no domínio humano), o ciclo começa no Estabelecimento; se centrado é em elementos tangíveis (como no domínio de produtos), começa na Servitização:

- No Estabelecimento, identifica-se oportunidades para criar valores ao cliente.
- No Reforço, a existência das oportunidades é confirmada e prepara-se ou atualiza-se o conceito do serviço.
- Na Infraestrutura, adquirem-se os recursos necessários para concretizar o novo conceito do serviço.

- Na Servitização, o novo serviço é concretizado.

Integrando os três artefatos, surge o modelo a seguir, representado na Figura 4. Cada processo do modelo ERIS possui dois processos do modelo SECI associados, sendo um principal e outro secundário. Além disso, cada processo também está associado a dois processos de Gestão do Conhecimento e a um grupo de sensores.

Figura 4 - Modelo de Gestão do Conhecimento integrado à Internet of Everything



Fonte: Adaptado de Costa (2022, p. 92)

- Estabelecimento
 - Processos do Modelo SECI
 - *Principal*: Socialização - A identificação das oportunidades ocorre pela observação atenta do comportamento humano, permitindo descobrir uma necessidade ainda não atendida.
 - *Secundário*: Internalização - A identificação das oportunidades pode ser auxiliada pela aplicação de um conhecimento ou pelo aprendizado das informações extraídas dos dados coletados.

- Processos de Gestão do Conhecimento
 - Identificação
 - Avaliação
- Sensores da Internet of Everything participantes
 - Aqueles que possuem conhecimento tácito e implícito
- Reforço
 - Processos do Modelo SECI
 - *Principal*: Externalização – Criação e identificação de conceitos por meio da reflexão e do diálogo, permitindo aproveitar a oportunidade e criar ou modificar um serviço.
 - *Secundário*: Socialização - Observação atenta para confirmar a existência das demandas.
 - Processos de Gestão do Conhecimento
 - Atualização
 - Proteção
 - Sensores da Internet of Everything participantes
 - Aqueles que possuem conhecimento explícito e tácito

- Infraestrutura
 - Processos do Modelo SECI
 - *Principal*: Combinação - Sistematização do conhecimento criado para usá-lo na concretização do serviço.
 - *Secundário*: Externalização - Durante o processo de sistematização do conhecimento, pode ser percebida a necessidade de formular mais conceitos.
 - Processos de Gestão do Conhecimento
 - Armazenamento
 - Transferência
 - Sensores de Internet of Everything participantes
 - Aqueles que possuem conhecimento explícito
- Servitização
 - Processo do Modelo SECI
 - *Principal*: Internalização - Aplicação dos conhecimentos na concretização do serviço
 - *Secundário*: Combinação - Complemento da sistematização, quando se percebe na prática a ausência de sistematização de certos conhecimentos
 - Processos de Gestão do Conhecimento
 - Aquisição
 - Utilização
 - Sensores de Internet of Everything participantes
 - Todos

4 Mineração de Dados

4.1 O que é

No início da computação, o custo para armazenar dados era muito alto, porém reduziu significativamente com o passar do tempo. Aproveitando-se desse baixo custo, as organizações começaram a armazenar uma grande quantidade de dados. Contudo, muitos desses dados não tinham utilidade. Então, como aproveitá-los? Assim, surgiu a mineração de dados (ou Data Mining, em inglês) (AMO, 2004, p. 2).

A mineração de dados é uma técnica automatizada para extrair padrões e informações relevantes de um conjunto de dados (AMO, p. 2, 2004). O termo surgiu na década de 1990, no contexto do processo de Descoberta de Conhecimento em Bases de Dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, p. 39, 1996). Possui aplicação em diversas áreas de negócios e da ciência, como (CAMILO; SILVA, 2009, p. 2):

- Nas compras com cartão de crédito, para a identificação de segmentos do mercado e detecção de fraudes.
- Na medicina, para diagnósticos mais precisos.
- Nos negócios, para identificação de perfis de clientes.
- Na tomada de decisão, para filtrar informações relevantes.

Uma lenda famosa sobre a mineração de dados é o Walmart que buscava associações entre os dados de venda e descobriu que, na noite de sexta-feira, havia uma grande venda associada de fraldas e cervejas, porque jovens pais vinham comprar fraldas para seus filhos e se recompensavam pela compra de cerveja. Percebendo isso, o Walmart teria colocado esses dois produtos próximos um do outro e alavancado as vendas de fralda e cerveja. Não há, entretanto, confirmação disso. O fato mais próximo é de que Thomas Blischok, executivo da NCR, fez um estudo em 1992 para a American Stores' Osco Drugs e descobriu várias associações nos dados de vendas, entre elas, a conexão entre a venda de cervejas e fraldas durante as 17 e 19 horas. A Osco não modificou a disposição dos produtos, mas a história se tornou famosa depois que Blischok a contou em um discurso para outros consultores (BIRTH, 1998).

No futuro, a mineração de dados terá cada vez mais aplicações. Uma tendência é a Mineração de Dados Distribuída, que busca possibilitar que a Mineração ocorra em diferentes locais e bancos de dados, em vez de centralizá-la em um único local (MALLADI et al., 2020, p. 206). Outra tendência é a Mineração de Dados em Multimídia, que analisa imagens, áudios e vídeos na busca de informações relevantes; ela ainda é desafiadora, por conta da dificuldade de que os dados em multimídia são não estruturados (MALLADI et al., 2020, p. 210).

4.2 Processos

Existem vários modelos de processos da mineração de dados (CAMILO; SILVA, 2009, p. 3). Um dos mais usados é o *CRISP-DM* (*Cross Industry Standard Process for Data Mining*) que é um padrão dividido em seis etapas iterativas (SCHRÖER; KRUSE; GÓMEZ, 2021, p. 527):

- **Entendimento do Negócio:** Etapa em que se entende qual o contexto do negócio, quais os recursos disponíveis e necessários e qual objetivo será buscado com a mineração de dados.
- **Entendimento dos Dados:** Etapa em que os dados são coletados, explorados, descritos e compreendidos.
- **Preparação dos Dados:** Etapa em que os dados são limpos, integrados, transformados e reduzidos para que possam passar pela Mineração de Dados.
- **Modelagem:** Etapa em que ocorre a Mineração de Dados propriamente dita. Aqui se escolhem as técnicas a serem usadas, os casos de treino e os casos de teste.
- **Avaliação:** Nessa etapa, compara-se os resultados obtidos com os objetivos definidos na etapa de Entendimento do Negócio, além de fazer uma revisão geral do processo.
- **Implantação:** Etapa em que os resultados obtidos são aplicados. Exemplos de aplicação incluem um componente de software ou apenas um relatório final.

As etapas de preparação e as técnicas de modelagem serão apresentadas a seguir.

4.3 Preparação dos dados

Antes de passarem pela mineração de dados propriamente dita, os dados precisam ser preparados por alguns processos (HAN; KAMBER, 2006 apud CAMILO; SILVA, 2009, p. 6-7):

- **Limpeza:** Muitas vezes, os dados vêm incompletos, inconsistentes ou errôneos. Nessa etapa, ocorre a correção desses dados, seja eliminando os incorretos, seja os substituindo por um valor adequado.
- **Integração:** Nessa etapa, ocorre a unificação dos dados de diversas fontes em um único repositório. Isso exige a análise de redundância, de valores conflitantes e de dependências entre os atributos.
- **Transformação:** Nessa etapa, ocorre a criação ou a conversão de atributos numéricos em nominais ou vice-versa. É necessária porque alguns algoritmos só trabalham com um tipo de atributo.
- **Redução:** Nessa etapa, ocorre a diminuição da quantidade de dados a ser analisada, pois a quantidade tende a ser muito alta e tornar a mineração impraticável. A redução ocorre de modo a não fazer os dados perderem sua representatividade original.

4.4 Técnicas

Existem algumas técnicas de mineração de dados. Alguns conceitos são necessários para entender essas técnicas:

- **Atributo:** O registro de uma característica de um objeto. Pode ser numérico ou nominal (WITTEN; FRANK; HALL, 2011, p. 49).
- **Instância:** O conjunto pré-determinado de atributos que descreve as características de um objeto (WITTEN; FRANK; HALL, 2011, p. 39).
- **Classe:** Atributo usado de forma especial para categorizar as instâncias, sendo um resultado da avaliação dos outros atributos da instância. Pode ser numérico ou nominal (WITTEN; FRANK; HALL, 2011, p. 40).

- **Dataset:** Conjunto de instâncias.
- **Modelo:** Resultado da aplicação de uma técnica em um *dataset*.
- **Dataset de Treinamento:** *Dataset* usado para desenvolver um modelo. (CAMILO; SILVA, 2009, p. 7)
- **Dataset de Testes:** *Dataset* usado para testar o modelo desenvolvido, verificando sua taxa de erros (CAMILO; SILVA, 2009, p. 7).

Entre as tarefas usadas pela mineração de dados, estão (LAROSE, 2005 apud CAMILO; SILVA, 2009, p. 8-10):

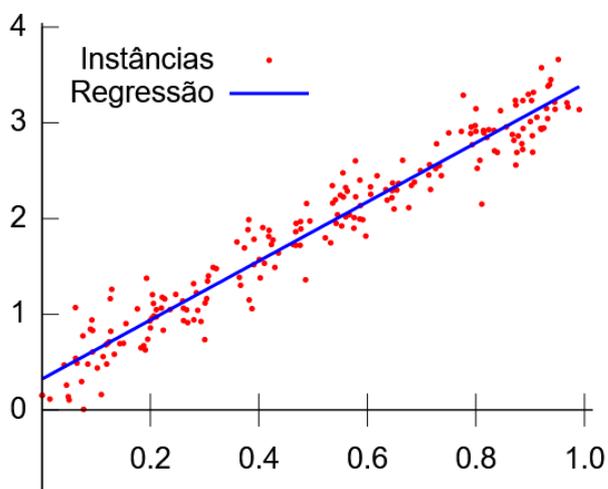
- **Descrição:** Tarefa que tenta expressar padrões e informações relevantes encontrados nos dados.
- **Classificação:** Tarefa que, dada uma instância, tenta identificar a que classe ela pertence. É usada quando a classe a ser identificada é nominal.
- **Estimação ou Regressão:** Similar à classificação, mas usada quando a classe a ser identificada é numérica. Seus algoritmos serão apresentados na próxima seção.
- **Agrupamento ou Clusterização:** Busca agrupar as instâncias do *dataset* de forma que as semelhantes estejam em um mesmo grupo e as diferentes estejam em grupos distintos. Diferente da classificação e da regressão, não utiliza classes; os grupos são criados a partir do exame dos atributos.
- **Associação:** Tenta descobrir associações entre os atributos na forma de regras. Regras são afirmações no modelo “SE atributo X ENTÃO atributo Y”, de modo que se a premissa for verdadeira, a conclusão é verdadeira.

Independente da tarefa adotada, é importante definir uma *baseline* (ou linha de base, em português), que é o desempenho mínimo esperado do algoritmo usado na técnica. Qualquer algoritmo que esteja abaixo ou muito próximo dessa linha de base é considerado ineficiente.

4.5 Algoritmos de Regressão

4.5.1 Regressão Linear

Figura 5 - Representação de uma regressão linear



Fonte: Adaptado de Berland (2007)

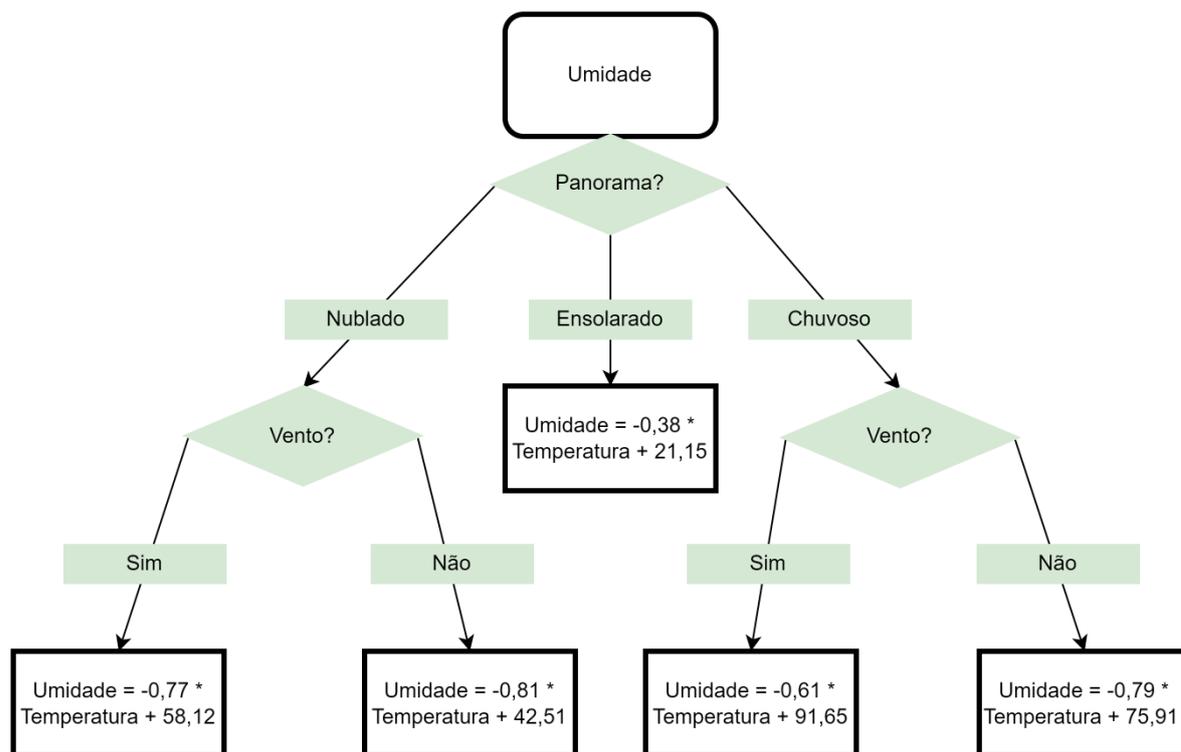
Regressão Linear é usada quando os atributos e a classe possuem uma relação matemática que pode ser descrita por meio de uma função linear, do tipo $y = ax + b$ (CAMILO; SILVA, 2009, p. 17). A Figura 5 mostra uma representação de uma regressão linear.

4.5.2 Árvores de Decisão

As árvores de decisão são fluxogramas na forma de árvore onde cada nó indica um teste de valor a ser feito no valor de um atributo (como $\text{massa} > 50$). As folhas indicam um valor ou intervalo de valores para a classe. Sua vantagem vem de sua simplicidade (CAMILO; SILVA, 2009, p. 12), facilitando a compreensão mesmo para quem não tem conhecimentos de Mineração de Dados. A Figura 6 mostra um exemplo de árvore de decisão, que se divide nos atributos "Panorama" e "Vento" e em que cada folha possui um modelo de regressão linear para a classe de Umidade.

Figura 6 - Exemplo de árvore de decisão

Fonte: Elaborado pelo autor (2022)



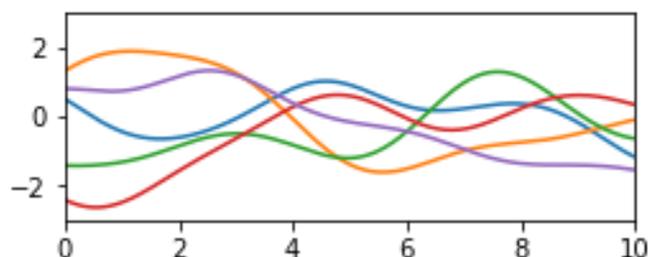
Entre os algoritmos de árvore de decisão para regressão, estão:

M5: Algoritmo desenvolvido por Quinlan em 1992 e uma das árvores de decisão pioneiras a lidar com classes numéricas. Ele constrói a árvore de modo a reduzir gradativamente a variação entre as instâncias que se encaixam naquele teste. Cada folha possui um modelo de regressão linear (WANG; WITTEN, 1996, p. 2).

- **Random Forest:** Algoritmo que reúne várias árvores de decisão cujas estruturas dependem de um vetor aleatório, sendo um para cada árvore. Cada árvore de decisão “vota” com uma classe no momento de definir uma classe para a instância (BREIMAN, 2001, p. 5-6). A média das classes atribuída é a classe final para aquela instância (BREIMAN, 2001, p. 26).

4.5.3 Processos Gaussianos

Figura 7 - Exemplos de processos gaussianos

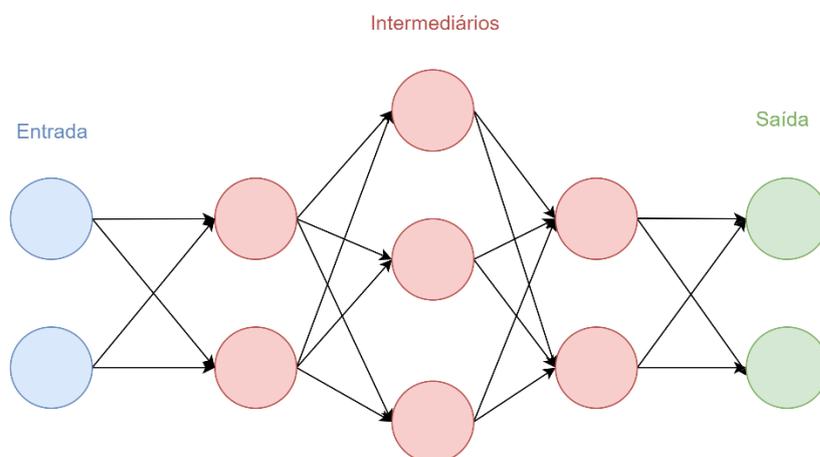


Fonte: PHYSIKINGER, 2017

Os processos gaussianos são modelos estatísticos que permitem realizar previsões com incerteza, ou seja, que representam um intervalo em vez de um valor fixo. Esses processos partem da pressuposição de que os valores a serem previstos seguem uma distribuição gaussiana (YI, 2019). A Figura 7 ilustra alguns exemplos de processos gaussianos.

4.5.4 Redes Neurais Artificiais

Figura 8 - Exemplo de Rede Neural Artificial



Fonte: Elaborado pelo autor (2022)

As redes neurais artificiais são algoritmos que buscam imitar o funcionamento do cérebro humano. Essas redes são compostas por várias unidades de processamento chamadas de neurônios, podendo receber inputs e retornar outputs. Os neurônios são ligados entre si por caminhos, que, quanto mais usados, terão maior peso e serão escolhidos com mais frequência. Conforme o treinamento, esses pesos são ajustados para realizar as classificações corretamente. (CAMILO; SILVA, 2009, p. 14).

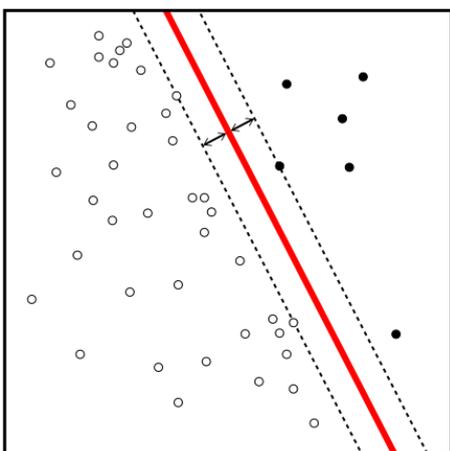
A Figura 8 apresenta um exemplo de rede neural, com a camada de entrada, três camadas intermediárias e a camada de saída.

Rauber (2005, p. 3) lista algumas vantagens relacionadas a esse tipo de algoritmo:

- **Tolerância a falhas:** Falhas em alguns neurônios não prejudicam a funcionalidade global.
- **Capacidade de aprendizagem:** Os neurônios são capazes de lidar com situações não vistas antes.
- **Processamento de informação incerta:** Mesmo que a informação seja ruidosa ou incompleta, é possível chegar a um raciocínio correto.
- **Paralelismo:** É possível trabalhar com vários neurônios ao mesmo tempo, sem a limitação do processador de executar apenas uma instrução por vez.

4.5.5 Support Vector Machines

Figura 9 - Exemplo de *Support Vector Machine*



Fonte: Adaptado de Alisneaky (2017)

Support Vector Machines (ou Máquinas de Vetores de Suporte, em português) são um algoritmo que busca criar a melhor divisão entre classes por meio da maximização da margem entre os pontos mais próximos das classes. As instâncias que estão mais próximas da linha de separação são chamadas de “Vetores de Suporte” (WITTEN; FRANK; HALL, 2011, p. 224-225). A figura 9 representa um exemplo de *Support Vector Machine*.

4.6 Métricas estatísticas de erro

Algumas métricas podem ser usadas para medir a precisão dos algoritmos de regressão. (NASER; ALAVI, 2021, p. 4-5) Elas são listadas abaixo. Em todas elas, “ E_i ” é um erro i medido pela diferença entre obtido i e esperado i , “ A_i ” é um valor real i e “ $A_{média}$ ” é a média dos valores reais.

Mean Absolute Error (MAE), ou Erro Médio Absoluto, é dado pela seguinte fórmula:

$$\frac{\sum_{i=1}^n |E_i|}{n}$$

Root Mean Squared Error (RMSE), ou raiz do erro médio quadrado, é dado pela seguinte fórmula:

$$\sqrt{\frac{\sum_{i=1}^n E_i^2}{n}}$$

Relative Absolute Error (RAE), ou erro relativo absoluto, expressa a razão do erro médio para os erros produzidos por um modelo trivial. É dado pela seguinte fórmula:

$$\sum_{i=1}^n \frac{|E_i|}{|A_i - A_{média}|}$$

Root Relative Squared Error (RRSE), ou raiz do erro quadrado relativo, é dado pela fórmula a seguir. Quanto mais próximo de zero, melhor:

$$\sqrt{\sum_{i=1}^n \frac{E_i^2}{(A_i - A_{média})^2}}$$

4.7 Mineração de Dados, Criação do Conhecimento e Internet of Everything

A Mineração de Dados é um dos diversos métodos que podem ser usados para extrair conhecimento dos dados coletados pela Internet of Things e pela Internet of Everything.

Segundo Siow, Tiropanis e Hall (2018, p. 14), a Mineração de Dados se encaixa nas categorias de Descrição, Diagnóstico, Descoberta e Predição. Ela não se encaixa na categoria de Prescrição. No Quadro 1, é realizada uma associação entre as categorias dos autores e as técnicas de Mineração de Dados.

Quadro 1 - Categorias de análise de Internet of Things aplicadas às técnicas de Mineração de Dados

	Descrição	Diagnóstico	Descoberta	Predição
Descrição	X			
Classificação	X		X	X
Regressão	X		X	X
Agrupamento	X		X	X
Associação		X	X	X

Fonte: Adaptado de Siow, Tiropanis e Hall (2018, p. 14)

No modelo de gestão do conhecimento integrado à Internet of Everything, a Mineração de Dados se encaixa principalmente na etapa de “Estabelecimento” e secundariamente na etapa de “Infraestrutura”. É por meio dos *insights* descobertos por essa metodologia que é possível identificar as novas oportunidades que são buscadas durante o Estabelecimento. As descrições feitas pela Mineração de Dados podem auxiliar na sistematização do conhecimento desejada na etapa de Infraestrutura.

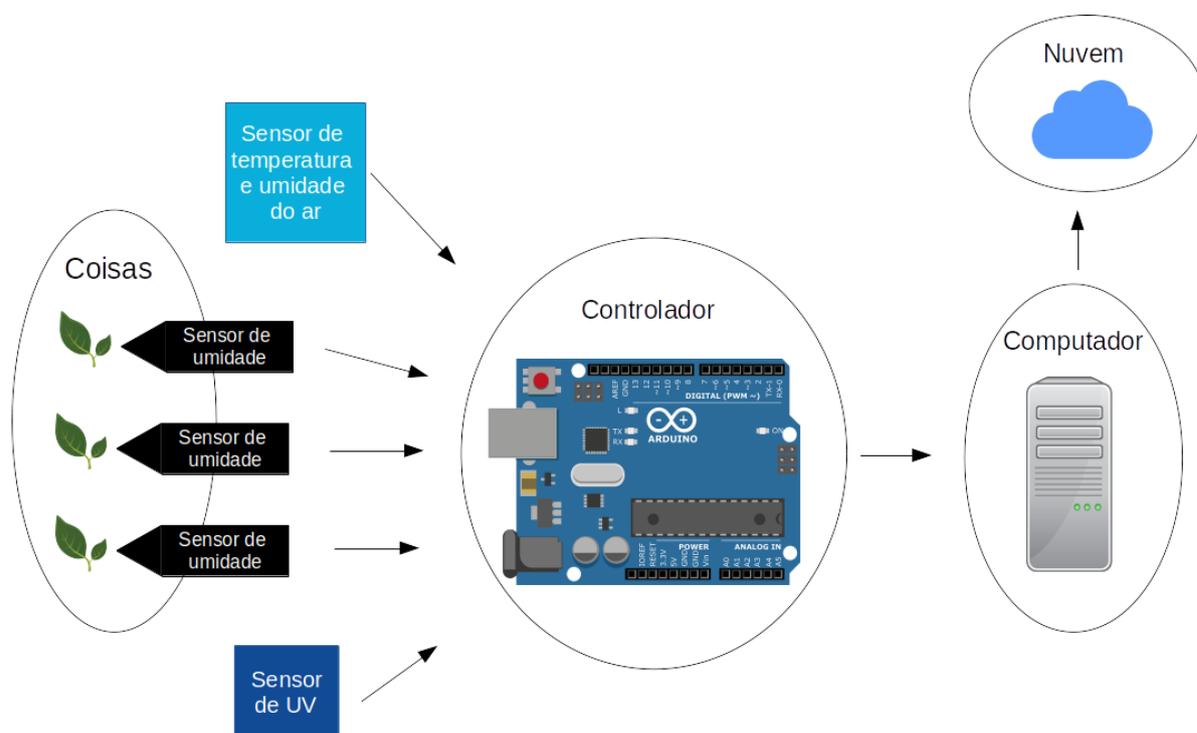
5 Projeto

Para fins de exploração e exemplificação de como a Internet of Things pode ser usada na criação de conhecimento, foi desenvolvido um projeto, que a partir de dados coletados de três plantas e de três parâmetros coletados ambiente, temperatura, umidade do ar e raios UV, tenta descobrir qual será a umidade 12 horas depois, auxiliando a determinar se a planta necessitará de água ou não.

Como esse projeto não envolve a participação de usuários, não pôde ser considerado uma aplicação de Internet of Everything. Por isso, ficou limitado à Internet of Things.

5.1 Descrição Física

Figura 10 - Representação do projeto



Fonte: Elaborado pelo autor (2022)

Os sensores extraem os dados das "coisas" e do ambiente. O Arduino capta esses dados por meio do script em sua memória interna e envia para o computador. Por meio de um script Python, o computador registra essa leitura em um arquivo. Por meio de um outro script Python, o computador sincroniza os dados no seu armazenamento local com os dados no armazenamento da nuvem.

5.1.1 Coisas

Há três plantas cuja umidade está sendo coletada em tempo real. Essas plantas têm portes diferentes, a fim de gerar uma variabilidade nas umidades coletadas. As Figuras 11, 12 e 13 mostram as plantas usadas.

Figura 11 - Planta 1



Fonte: Elaborado pelo autor (2022)

Figura 12 - Planta 2



Fonte: Elaborado pelo autor (2022)

Figura 13 - Planta 3

Fonte: Elaborado pelo autor (2022)

5.1.2 Sensores

Há três sensores para umidade, um para temperatura e umidade do ar e outro para raios UV. As características dos sensores estão descritas no Quadro 2.

Quadro 2 - Características dos sensores

Sensor	Grandeza Medida	Output	Tipo de Output	Alimentação	Quantidade
Capacitive Soil Moisture Sensor V 1.2	Umidade do Solo	0 a 1023	Analógico	5V	3
DHT11	Temperatura e Umidade do Ar	Temperatura em °C e Umidade em %	Digital	5V	1
GYML8511	Raios UV	0 a 1023 (Medição), 0 a 1023 (Referência)	Analógico	3,3V	1

Fonte: Elaborado pelo autor (2022)

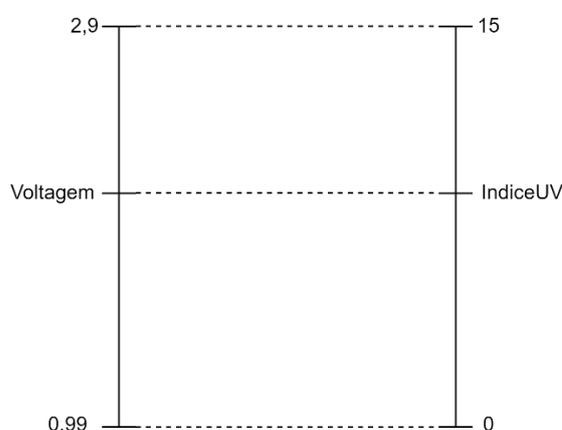
Os sensores de umidade retornam a voltagem em um valor de 0 a 1023. Quanto maior essa voltagem, mais úmidos os sensores estão.

O sensor DHT11 retorna dois valores: a temperatura, em °C e a umidade do ar, em porcentagem.

O sensor de UV retorna um valor em voltagem. Quanto maior esse valor, maior a radiação UV captada. Para garantir mais confiança à medição, o sensor também emite um valor de referência, que indica quanta alimentação ele está recebendo. No final, a voltagem final do sensor UV é dada por:

$$\text{Voltagem Final} = \frac{3,3}{\text{Referência}} \times \text{Valor Coletado}$$

E essa voltagem pode ser convertida para o índice UV (em mW/cm²) por meio da fórmula abaixo (SUNFOUNDER, 2019), que deriva do Teorema de Tales:



$$\frac{\text{Voltagem} - 0,99}{\text{Índice UV} - 0} = \frac{2,9 - 0,99}{15 - 0}$$

$$\frac{\text{Voltagem} - 0,99}{\text{Índice UV}} = \frac{1,91}{15}$$

$$\text{Voltagem} - 0,99 = \frac{1,91}{15} \times \text{Índice UV}$$

$$(\text{Voltagem} - 0,99) \times \frac{15}{1,91} = \text{Índice UV}$$

$$\text{Índice UV} = (\text{Voltagem} - 0,99) \times \frac{15}{1,91}$$

Na Tabela 1, estão os intervalos aceitáveis para cada grandeza. Leituras fora desses intervalos são consideradas errôneas.

Tabela 1 - Intervalos aceitáveis para cada grandeza

Grandeza	Mínimo	Máximo
Umidade do Solo	300	1000
Temperatura	0	50
Umidade do Ar	0%	99%
Índice UV	-0,26	15

Fonte: Elaborado pelo autor (2022)

Seguindo as categorias de dimensão do conhecimento da Internet of Things apresentadas por Costa (2022, p. 77-83), é possível classificar cada sensor da seguinte forma:

- Quanto à explicitação, todos os sensores são do tipo “conhecimento implícito”, pois são dados que ainda necessitam de tratamento para que sejam considerados conhecimento propriamente dito.
- Quanto à estrutura, todos os sensores são do tipo “estruturado”, pois devolvem um valor ou um par de valores.
- Quanto à confiança, os Capacitive Soil Moisture Sensor V 1.2 e o DHT11 podem ser considerados confiáveis pelo baixo índice de leituras errôneas. Já o GYML8511 é considerado não-confiável porque as leituras neste projeto foram muito instáveis, variando grandemente mesmo com intervalo de segundos entre uma e outra; apenas calculando-se as médias para as leituras de uma hora, foi possível obter um valor confiável, e por isso -0,26 foi tolerado como mínimo.
- Quanto ao resultado, todos obedecem ao tipo “complementação”, pois os dados se combinam uns aos outros na descrição do estado das plantas e do ambiente.
- Quanto à ação, todos são do tipo automação, pois apenas visam diminuir a necessidade humana de verificar se as plantas precisam ser regadas e quando deverão ser regadas.

5.1.3 Controlador

O controlador é um Arduino, na sua versão “Mega”. Ele possui 54 pinos para outputs digitais e 16 pinos para outputs analógicos.

No Arduino, foi instalado um script que faz uma leitura a cada 900 milissegundos. Esse limite foi escolhido para garantir que houvesse pelo menos uma leitura a cada segundo. Se fosse usado como valor 1000 milissegundos, caso a execução dos passos não fosse rápida o suficiente, faltaria uma leitura naquele determinado segundo.

O script do Arduino grava os dados na seguinte ordem: “Umidade1_V, Umidade2_V, Umidade3_V, Temperatura, UmidadeAr, IndiceUV_V, IndiceUV_Ref, outputVoltage, indiceUV, TempoMs, DataHora”, onde:

- **Umidade1_V** é a umidade do solo da primeira planta;
- **Umidade2_V** é a umidade do solo da segunda planta;
- **Umidade3_V** é a umidade do solo da terceira planta;
- **Temperatura** é a temperatura ambiente;
- **UmidadeAr** é a umidade do ar;
- **IndiceUV_V** é o output de medição do sensor GYML8511;
- **IndiceUV_Ref** é o output de referência do sensor GYML8511;
- **outputVoltage** é a voltagem final do sensor UV, calculado a partir do IndiceUV_V e do IndiceUV_Ref;
- **indiceUV** é o índice UV calculado a partir da outputVoltage;
- **TempoMs** é a quantidade de milissegundos transcorrida desde que o Arduino foi ativado;
- **DataHora** é o dia, hora, minuto e segundo em que a leitura ocorreu. Como o Arduino não tem suporte a essa função nativamente, ele retorna apenas um valor em branco nessa parte e o computador fica responsável por preencher o valor em branco.

É possível ver o script no anexo “Anexo 1 - Script usado no Arduino”.

5.1.4 Computador

O computador recebe os dados do Arduino em uma porta USB e os grava em um arquivo CSV. Quando o arquivo completa 700 leituras, o arquivo é fechado e um novo arquivo é aberto para armazenar as leituras seguintes. Esse limite foi estabelecido para que o arquivo não ficasse excessivamente grande.

Os nomes dos arquivos obedecem ao formato: “leituraArduino_YYYY-MM-DD_hh-mm-ss”

A gravação ocorre por meio de um script Python (que pode ser visto no anexo “Anexo 2 - Script Python para gravar dados enviados pelo Arduino”), que segue os seguintes passos:

- Definição das variáveis
- Abertura do arquivo em que as leituras ficarão salvas
- Enquanto a porta do Arduino estiver aberta:
 - Se os cabeçalhos do arquivo CSV não foram escritos, realiza essa escrita
 - Escreve a linha obtida da porta do Arduino, acrescentando a Data e Hora no final (já que esse recurso não está disponível no Arduino)
 - Se atingiu o limite de 700 leituras por arquivo:
 - Fecha o arquivo atual
 - Abre um novo arquivo

Para enviar os dados para o backup na nuvem, é usado o programa Rclone, que permite sincronizar facilmente um repositório local com o repositório em nuvem (RCLONE, 2022). O script Python usado para executar esse programa está no anexo “Anexo 3 - Script Python para enviar os dados para a nuvem” e é executado automaticamente a cada 10 minutos.

5.1.5 Nuvem

Nesse projeto, a nuvem desempenha apenas o papel de backup dos dados. A plataforma de nuvem utilizada foi o OneDrive.

A categoria de computação mais adequada para esse projeto é a *Fog Computing*. Ela não se encaixa na *Edge Computing*, pois os sensores não realizam nenhum processamento, nem na *Cloud Computing*, pois todo o processamento ocorre no computador.

5.2 Mineração de Dados

5.2.1 Preparação dos dados

A coleta dos dados começou em 22 de outubro de 2022. Após cerca de um mês, foi iniciada a mineração dos dados. Por falhas dos sensores e do Arduino, as leituras nem sempre são contínuas: Há, por exemplo, um salto de 31/10/2022 para 09/11/2022.

O processo de preparação dos dados seguiu oito etapas:

1. Agregação
2. Limpeza
3. Redução apenas aos atributos úteis
4. Redução de horas, minutos e segundos para horas e minutos
5. Redução de horas e minutos para horas
6. Unificação dos arquivos
7. Adição de atributos com as Médias e Desvio-Padrão das 12 horas passadas
8. Adição da classe

Na primeira etapa, todas as leituras de um mesmo dia (no período de 00:00:00 a 23:59:59) são condensadas em um único arquivo. Assim, há um arquivo para cada dia, em vez de um arquivo para cada 700 leituras. O script Python pode ser visto no anexo “Anexo 4 - Script Python para agregar arquivos com dados do mesmo dia” e segue o seguinte funcionamento:

- Definir as variáveis, particularmente a lista de datas cujos arquivos serão criados
- Listar todos os arquivos da pasta e ordená-los em ordem alfabética
- Para cada data definida:
 - Recolher o último arquivo de leitura do Arduino do dia anterior
 - Recolher todos os arquivos referentes ao dia buscado

- Abrir o arquivo que guardará as leituras do dia
- Escrever os cabeçalhos
- Escrever as leituras do dia buscado que estão presentes no último arquivo do dia anterior
- Escrever as leituras do dia buscado que estão presentes nos arquivos do dia buscado

Na segunda etapa, todas as leituras errôneas foram corrigidas ou eliminadas. O script Python para identificar as leituras errôneas pode ser visto no anexo “Anexo 5 - Script Python para detectar erros nos arquivos” e segue o seguinte funcionamento:

- Para cada linha do arquivo CSV:
 - Somar 1 ao número de linhas totais
 - Para cada atributo da linha:
 - Verificar se está dentro do intervalo para aquele atributo. Caso contrário, imprimir a linha errônea na tela (destacando o atributo errôneo), somar 1 ao número de erros encontrados e pular para a próxima linha

A maior parte dos erros foi corrigida manualmente após sua identificação. Dois erros se repetiram com certa frequência:

- Uma sequência de três leituras errôneas para a temperatura e umidade do ar, com valores de -1. Os valores foram corrigidos com base nas leituras adjacentes.
- Sequências de leituras errôneas para a umidade do solo de alguma planta. Essas sequências de leituras errôneas foram eliminadas.

Na terceira etapa, foram eliminados os atributos de controle e mantidos apenas os atributos que seriam úteis para a mineração de dados. Os atributos mantidos foram: Umidade1_V, Umidade2_V, Umidade3_V, Temperatura, UmidadeAr, indiceUV e DataHora. O script que fez a eliminação está em anexo como “Anexo 6 - Script Python para remover atributos de controle”. Ele segue os seguintes passos:

- Para cada linha do arquivo CSV:
 - Recalcular o índiceUV, para garantir que está dentro do padrão (já que, como explicado na parte de sensores, o sensor de UV é instável)
 - Eliminar as colunas inúteis
 - Escrever em um novo arquivo a linha com o índiceUV recalculado e sem as colunas inúteis

Na quarta etapa, foi feita a redução das leituras de hora, minuto e segundo para apenas leituras de hora e minuto. O script Python que realizou essa etapa está no anexo “Anexo 7 - Script Python para reduzir leituras de segundos para leituras de minuto” e segue as seguintes etapas:

- Inicializar um vetor vazio para armazenar as leituras por minuto
- Definir uma variável tendo como a primeira linha do arquivo. Essa variável será responsável por armazenar o valor da linha anterior à atual sendo lida
- Para cada linha do arquivo CSV depois da primeira linha:
 - Verificar se a linha anterior e a linha atual pertencem ao mesmo minuto. Se sim, simplesmente adicionar a leitura ao vetor com as leituras do minuto. Se não, calcular as médias das leituras
 - A média não é calculada se o número de leituras para um mesmo minuto for menor que 45 ou maior que 75, já que o esperado é em torno de 60 leituras por minuto
 - Valores inteiros antes de serem convertidos para a média são arredondados para continuarem inteiros. Valores decimais continuam decimais após o cálculo da média, com duas casas após a vírgula
 - Se a média foi calculada normalmente, escrever as leituras e esvaziar o vetor
 - Substituir o valor da linha atual pelo valor da linha anterior

Na quinta etapa, foi feita a redução das leituras de hora e minuto para apenas leituras de hora. O script Python que realizou essa etapa está no anexo “Anexo 8 - Script Python para reduzir leituras de minutos para leituras de hora” e funciona de forma idêntica ao script da quarta etapa.

Na sexta etapa, com o número de instâncias reduzido significativamente, as leituras foram todas unificadas em um único arquivo. O script Python que realizou essa etapa está no anexo “Anexo 9 - Script Python para unificar as instâncias em um único *dataset*” e funciona da seguinte forma:

- Recuperar e listar todos os arquivos do diretório em ordem alfabética
- Recolher todos os arquivos que serão leituras do Arduino
- Abrir o arquivo da *dataset* unificado e escrever os cabeçalhos
- Para cada arquivo:
 - Abrir um leitor CSV para o arquivo
 - Para cada linha do arquivo:
 - Escrever essa linha no arquivo do *dataset* unificado

Na sétima etapa, foram acrescentados as médias e os desvios-padrão de cada atributo para as 12 horas anteriores. O script Python que realizou essa etapa está no anexo “Anexo 10 - Script Python para adicionar médias e desvios-padrão das 12 horas anteriores”. O funcionamento é da seguinte forma:

- Inicializar um vetor vazio para armazenar as linhas anteriores
- Para cada linha do arquivo:
 - Verificar se o tamanho do vetor é maior que 12, o que é sinal de erro grave. Caso isso aconteça, interromper a execução do programa
 - Se há 12 linhas no vetor e essas 12 linhas correspondem às 12 horas anteriores à linha sendo lida, escrever a linha com os novos atributos no novo *dataset* e remover o último elemento do vetor das linhas anteriores

- Se há 12 linhas no vetor, mas essas 12 linhas não correspondem às 12 horas anteriores, simplesmente esvaziar o vetor das linhas anteriores e não escrever nenhuma nova linha
- Inserir a linha sendo lida no início do vetor de linhas anteriores

Desse modo, os cabeçalhos ficaram como descritos no Quadro 3. Os cabeçalhos com “_MEDIA12H” representam os valores das médias das últimas 12 horas e os cabeçalhos com “_DP12H” representam os valores dos desvios-padrão das últimas 12 horas.

Quadro 3 - Cabeçalhos após a execução da sétima etapa

Umidade1_V, Umidade2_V, Umidade3_V, Temperatura, UmidadeAr, indiceUV,
 Umidade1_V_MEDIA12H, Umidade2_V_MEDIA12H, Umidade3_V_MEDIA12H,
 Temperatura_MEDIA12H, UmidadeAr_MEDIA12H, indiceUV_MEDIA12H,
 Umidade1_V_DP12H, Umidade2_V_DP12H, Umidade3_V_DP12H,
 Temperatura_DP12H, UmidadeAr_DP12H, indiceUV_DP12H, Data, Hora

Fonte: Elaborado pelo autor (2022)

Na oitava etapa, é feita a adição da classe, que é o valor da umidade de uma das plantas a partir de 12 horas daquela leitura. O script Python que realizou essa etapa está no anexo “Anexo 11 - Script para adicionar a classe”. Esse algoritmo descarta as linhas sem um correspondente de 12 horas atrás, garantindo que classe adicionada está certa, mesmo sem continuidade das leituras. O funcionamento é da seguinte forma:

- Definir as variáveis, especialmente a partir de qual umidade será criada a classe
- Inicializar um vetor com as 12 primeiras linhas do *dataset*.
- Para cada linha depois das 12 primeiras:
 - Verificar se há uma linha que possui o correspondente de 12 horas atrás no vetor de 12 linhas. Se sim, escrever a linha do vetor no arquivo, adicionando como valor da classe a umidade da linha que está sendo lida

- Acrescentar a linha que acabou de ser lida no vetor de 12 linhas e remover a última linha do vetor.
- Se por algum motivo o vetor de 12 linhas ultrapassou o tamanho de 12 linhas, interromper a execução do programa

Os cabeçalhos ficaram como descritos no Quadro 4, substituindo “X” no último cabeçalho pelo número da planta sendo analisada no *dataset*.

Quadro 4 - Cabeçalhos após a execução da oitava etapa

Umidade1_V, Umidade2_V, Umidade3_V, Temperatura, UmidadeAr, indiceUV, Umidade1_V_MEDIA12H, Umidade2_V_MEDIA12H, Umidade3_V_MEDIA12H, Temperatura_MEDIA12H, UmidadeAr_MEDIA12H, indiceUV_MEDIA12H, Umidade1_V_DP12H, Umidade2_V_DP12H, Umidade3_V_DP12H, Temperatura_DP12H, UmidadeAr_DP12H, indiceUV_DP12H, Data, Hora, UmidadeX_V

Fonte: Elaborado pelo autor (2022)

Criou-se, no total, 6 *datasets*, sendo 1 de treinamento e 1 de teste para cada planta. Isso exigiu 6 execuções do oitavo passo. O Caso A representa os *datasets* da planta 1, o Caso B os *datasets* da planta 2 e o Caso C os *datasets* da planta 3. É possível observar uma amostra de cada *dataset* nos anexos 12 a 17.

O Quadro 5 mostra algumas das características dos *datasets* de treinamento e de teste. A única diferença entre os *datasets* de treinamento de um caso e o *dataset* de treinamento de outro caso é o valor da classe. A mesma lógica se aplica aos *datasets* de teste.

Quadro 5 - Características dos *datasets* de treinamento e de teste

<i>Dataset</i>	Datas cobertas	Total de Instâncias
Treinamento	24/10/2022, 25/10/2022, 26/10/2022, 28/10/2022, 30/10/2022, 31/10/2022, 09/11/2022, 10/11/2022	109
Teste	13/11/2022, 14/11/2022, 15/11,2022, 16/11/2022, 17/11/2022, 18/11/2022	98

Fonte: Elaborado pelo autor (2022)

5.2.2 Modelagem

Após a preparação dos dados, segue-se a fase da modelagem, onde a partir do *dataset*, tenta-se prever qual será a umidade da planta 12 horas depois.

Os *datasets* foram modelados no Weka, um software de mineração de dados desenvolvido pela Universidade de Waikato (WEKA, 2022).

O Weka disponibiliza vários algoritmos para as diversas técnicas de mineração de dados. Para realizar a regressão, os seguintes algoritmos foram usados:

- **GaussianProcess:** Algoritmo de processos gaussianos;
- **LinearRegression:** Algoritmo de regressão linear;
- **MultilayerPerceptron:** Algoritmo de redes neurais artificiais;
- **SMOreg:** Algoritmo de *Support Vector Machines*;
- **M5P:** Implementação do algoritmo *M5*, uma árvore de decisão;
- **RandomForest:** Implementação do algoritmo *Random Forest*.

Os passos seguidos foram dois:

1. **Escolha do algoritmo e da linha de base:** Escolhe-se o algoritmo com melhor desempenho na tarefa trivial de descobrir qual a umidade futura usando apenas a umidade atual da planta. O melhor algoritmo segue para a próxima fase e seu desempenho é usado como linha de base. Em caso de empate, aplica-se os algoritmos empatados ao *dataset* bruto (com todos os atributos, exceto o atributo *Data*, que gera problema em alguns algoritmos) e aquele que obtém o melhor desempenho é escolhido;
2. **Otimização:** São realizados diversos experimentos removendo atributos do *dataset* até chegar a uma seleção de atributos que dá a menor taxa de erros.

5.3 Resultados

5.3.1 Caso A

Aqui, é aplicada a modelagem para a planta 1. O modelo é treinado com um *dataset* e testado com outro.

Na Tabela 2, está a aplicação dos algoritmos para escolher qual o melhor e definir a linha de base. Foi usado como atributo apenas “Umidade1_V”. As linhas em amarelo são as que possuem menores taxas de erro.

Tabela 2 - Aplicação dos algoritmos para escolha do melhor e definição da linha de base do Caso A

Algoritmo	MAE	RMSE	RAE	RRSE
<i>GaussianProcess</i>	14,533	17,1523	134,4392%	138,9045%
<i>LinearRegression</i>	3,583	4,5025	33,1454%	36,4622%
<i>MultilayerPerceptron</i>	9,3513	9,9859	86,5053%	80,8685%
<i>SMOreg</i>	3,8261	4,8528	35,3937%	39,2997%
<i>M5P</i>	3,583	4,5025	33,1454%	36,4622%
<i>RandomForest</i>	5,5735	6,8311	51,5585%	55,3205%

Fonte: Elaborado pelo autor (2022)

Como a *LinearRegression* e o *M5P* deram o mesmo resultado, foi feito desempate com o *dataset* bruto (apenas sem o atributo “Data”) para decidir qual otimizar, que está representado na Tabela 3. O algoritmo *M5P* foi o escolhido.

Tabela 3 - Desempate para escolher qual algoritmo otimizar para o Caso A

Algoritmo	MAE	RMSE	RAE	RRSE
<i>LinearRegression</i>	6,2514	8,1799	57,8294%	66,2435%
<i>M5P</i>	4,5275	6,4223	41,8822%	52,0099%

Fonte: Elaborado pelo autor (2022)

Na Tabela 4, está representada a seleção de atributos que traz a menor taxa de erros para o algoritmo *M5P*.

Tabela 4 - Otimização do algoritmo *M5P* para o Caso A

Atributos escolhidos	MAE	RMSE	RAE	RRSE
Umidade1_V, Temperatura	2,1276	2,5182	19,6814%	20,3932%

Fonte: Elaborado pelo autor (2022)

No entanto, aplicando o Algoritmo *LinearRegression* com os mesmos atributos escolhidos para o algoritmo *M5P*, obteve-se os mesmos resultados da Tabela 4. Assim, é possível representar o resultado desse caso pela função: $Umidade1_V_Classe = 0,8424 * Umidade1_V - 1,005 * Temperatura + 128,892$.

5.3.2 Caso B

Aqui, é aplicada a modelagem para a planta 2. Assim como no Caso A, o modelo é treinado com um *dataset* e testado com outro.

Na Tabela 5, está a aplicação dos algoritmos para escolher o algoritmo e definir a linha de base. Foi usado como atributo apenas “Umidade2_V”. A linha em amarelo representa a menor taxa de erro e, portanto, o algoritmo escolhido.

Tabela 5 - Aplicação dos algoritmos para escolha do melhor e definição da linha de base do Caso B

Algoritmo	MAE	RMSE	RAE	RRSE
<i>GaussianProcess</i>	47,8637	48,9811	113,9929%	112,6173%
<i>LinearRegression</i>	30,744	31,9602	73,2203%	73,4829%
<i>MultilayerPerceptron</i>	67,9229	68,0559	161,7663%	156,474%
<i>SMOreg</i>	38,8776	41,0841%	94,9732%	94,4606%
<i>M5P</i>	27,113	29,0401	64,5728%	66,7689%
<i>RandomForest</i>	32,5649	38,6411	77,5571%	88,8435%

Fonte: Elaborado pelo autor (2022)

Na Tabela 6, está representada a seleção de atributos que traz a menor taxa de erros para o algoritmo *M5P*.

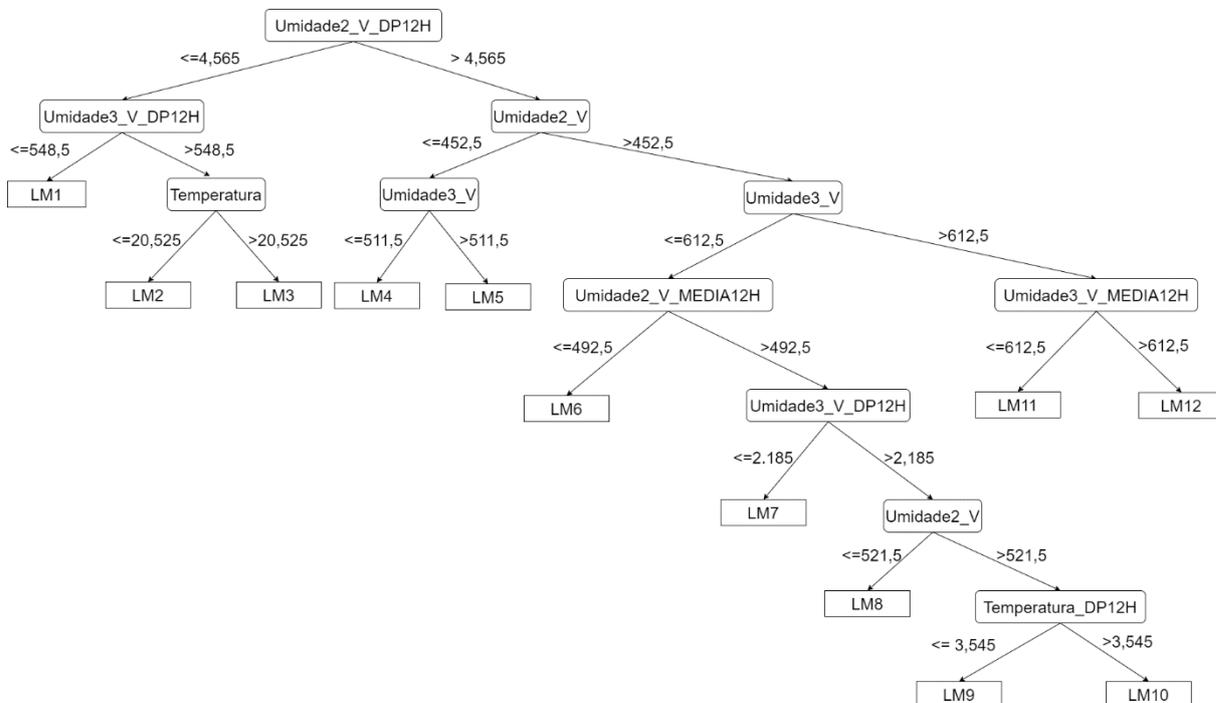
Tabela 6 - Otimização do algoritmo *M5P* para o Caso B

Atributos escolhidos	MAE	RMSE	RAE	RRSE
Umidade2_V, Umidade3_V, Temperatura, indiceUV, Umidade2_V_MEDIA12H, Umidade3_V_MEDIA12H, Temperatura_MEDIA12H, indiceUV_MEDIA12H, Umidade2_V_DP12H, Umidade3_V_DP12H, Temperatura_DP12H, indiceUV_DP12H	7,413	8,5481	17,655%	19,6538%

Fonte: Elaborado pelo autor (2022)

O *M5P* produziu uma árvore, que pode ser visualizada na Figura 14. O modelo de regressão linear que está em cada folha pode ser visualizado no Quadro 6.

Figura 14 - Árvore de Decisão criada pelo algoritmo *M5P* para a planta 2



Fonte: Elaborado pelo autor (2022)

Quadro 6 - Regressões lineares de cada folha da árvore de decisão do algoritmo M5P

Folha	Equação para o valor de UMIDADE2_V_CLASSE
LM1	$-0.0193 * Umidade2_V + 0.087 * Umidade3_V + 1.0257 * Temperatura - 80.8467 * indiceUV + 0.0848 * umidade2_V_MEDIA12H + 0.3531 * Umidade3_V_MEDIA12H - 28.3216 * indiceUV_MEDIA12H - 0.4448 * Umidade3_V_DP12H - 1.1309 * Temperatura_DP12H + 233.0324$
LM2	$-0.0193 * Umidade2_V + 0.087 * Umidade3_V + 1.0257 * Temperatura - 80.8467 * indiceUV + 0.0848 * Umidade2_V_MEDIA12H + 0.6872 * Umidade3_V_MEDIA12H - 83.573 * indiceUV_MEDIA12H - 0.4448 * Umidade3_V_DP12H - 1.1309 * Temperatura_DP12H + 54.8302$
LM3	$-0.0193 * Umidade2_V + 0.087 * Umidade3_V + 1.0257 * Temperatura - 80.8467 * indiceUV + 0.0848 * Umidade2_V_MEDIA12H + 0.4779 * Umidade3_V_MEDIA12H - 48.8569 * indiceUV_MEDIA12H - 0.4448 * Umidade3_V_DP12H - 1.1309 * Temperatura_DP12H + 167.9306$
LM4	$-0.0791 * Umidade2_V + 0.3097 * Umidade3_V + 0.8259 * Temperatura - 65.0973 * indiceUV + 0.2441 * Umidade2_V_MEDIA12H + 0.3008 * Umidade3_V_MEDIA12H - 1.215 * Umidade3_V_DP12H - 0.9106 * Temperatura_DP12H + 65.5928$
LM5	$-0.0791 * Umidade2_V + 0.3223 * Umidade3_V + 0.8259 * Temperatura - 30.589 * indiceUV + 0.2441 * Umidade2_V_MEDIA12H + 0.3008 * Umidade3_V_MEDIA12H - 1.215 * Umidade3_V_DP12H - 0.9106 * Temperatura_DP12H + 62.2766$
LM6	$-0.0463 * Umidade2_V + 0.2201 * Umidade3_V + 3.1814 * Temperatura - 180.5555 * indiceUV + 0.2348 * Umidade2_V_MEDIA12H + 0.1879 * Umidade3_V_MEDIA12H - 1.1324 * Umidade3_V_DP12H - 3.0935 * Temperatura_DP12H + 156.6983$
LM7	$-0.0252 * Umidade2_V + 0.2201 * Umidade3_V + 3.0886 * Temperatura - 180.5555 * indiceUV + 0.2248 * Umidade2_V_MEDIA12H + 0.1879 * Umidade3_V_MEDIA12H - 1.1018 * Umidade3_V_DP12H - 2.3802 * Temperatura_DP12H + 150.1777$
LM8	$-0.0207 * Umidade2_V + 0.2201 * Umidade3_V + 3.4205 * Temperatura - 180.5555 * indiceUV + 0.2248 * Umidade2_V_MEDIA12H + 0.1879 * Umidade3_V_MEDIA12H - 1.1018 * Umidade3_V_DP12H - 2.5331 * Temperatura_DP12H + 138.0072$
LM9	$-0.0217 * Umidade2_V + 0.2201 * Umidade3_V + 3.3276 * Temperatura - 180.5555 * indiceUV + 0.2248 * Umidade2_V_MEDIA12H + 0.1879 * Umidade3_V_MEDIA12H - 1.1018 * Umidade3_V_DP12H - 2.5331 * Temperatura_DP12H + 142.379$
LM10	$-0.0215 * Umidade2_V + 0.2201 * Umidade3_V + 3.377 * Temperatura - 180.5555 * indiceUV + 0.2248 * Umidade2_V_MEDIA12H + 0.1879 * Umidade3_V_MEDIA12H - 1.1018 * Umidade3_V_DP12H - 2.5331 * Temperatura_DP12H + 140.5797$
LM11	$0.411 * Umidade2_V + 0.2634 * Umidade3_V + 3.8799 * Temperatura - 420.8334 * indiceUV + 0.2124 * Umidade2_V_MEDIA12H + 10.0102 * Umidade3_V_MEDIA12H - 1.1263 * Umidade3_V_DP12H - 4.3305 * Temperatura_DP12H - 6064.6153$
LM12	$-0.0821 * Umidade2_V + 0.2634 * Umidade3_V + 3.8799 * Temperatura - 438.3362 * indiceUV + 0.2124 * Umidade2_V_MEDIA12H + 0.1879 * Umidade3_V_MEDIA12H - 1.1263 * Umidade3_V_DP12H - 4.3305 * Temperatura_DP12H + 192.7471$

Fonte: Elaborado pelo autor (2022)

5.3.3 Caso C

Aqui, é aplicada a modelagem para a planta 3. Assim como nos outros casos, o modelo é treinado com um *dataset* e testado com outro.

Na Tabela 7, está a aplicação dos algoritmos para escolher o algoritmo e definir a linha de base. Foi usado como atributo apenas “Umidade3_V”. A linha em amarelo representa a menor taxa de erro e, portanto, o algoritmo escolhido.

Tabela 7 - Aplicação dos algoritmos para escolha do melhor e definição da linha de base do Caso C

Algoritmo	MAE	RMSE	RAE	RRSE
<i>GaussianProcess</i>	115,918	116,2349	99,5023%	99,4717%
<i>LinearRegression</i>	57,6557	58,1084	49,4908%	49,7281%
<i>MultilayerPerceptron</i>	49,0953	49,8513	42,1427%	42,6618%
<i>SMOreg</i>	25,7838	26,6058	22,1324%	22,7688%
<i>M5P</i>	18,0102	19,2088	15,4597%	16,4386%
<i>RandomForest</i>	16,2425	17,7126	13,9423%	15,1581%

Fonte: Elaborado pelo autor (2022)

Na Tabela 8, está representada a seleção de atributos que traz o melhor resultado para o algoritmo *RandomForest*.

Tabela 8 - Otimização do algoritmo *RandomForest* para o Caso C

Atributos escolhidos	MAE	RMSE	RAE	RRSE
Umidade3_V, Temperatura, UmidadeAr, Umidade3_V_MEDIA12H	15,9372	17,3849	13,6802%	14,8777%

Fonte: Elaborado pelo autor (2022)

O algoritmo *RandomForest* não retornou nenhum resultado na forma de equação ou visual.

5.4 Discussão

Nas categorias de análise de dados em Internet of Things, o projeto se encaixa nos papéis descritivo e preditivo. Descritivo, porque os dados foram usados para descrever a umidade das plantas e as condições ambientais. Preditivo, porque foram usados para prever a umidade da planta depois de 12 horas.

O conhecimento que pode extraído dos dados coletados pelos sensores é implícito, mas meu conhecimento tácito de Mineração de Dados também participou do processo.

No modelo de Gestão do Conhecimento integrado à Internet of Everything, o projeto se encaixa no Estabelecimento e na Servitização. No Estabelecimento, por conta da tentativa de identificar padrões e informações relevantes nos dados, que envolveu Socialização e Internalização. Na Servitização, porque o objetivo da identificação dos padrões e das informações seria a concretização de um serviço para o usuário, envolvendo Internalização e Combinação.

O objetivo de prever a umidade da planta 12 horas depois foi parcialmente atingido, pois os algoritmos apresentaram métricas relativamente baixas de erro; contudo, como testes não foram feitos com *datasets* com mais variedade no tempo meteorológico, não é possível estabelecer a confiança total nas previsões do modelo.

Entre as descobertas desse projeto, é possível citar:

- Enquanto a planta 1 necessitou apenas da temperatura, a planta 2 e a planta 3 precisaram de outros parâmetros para prever com mais precisão a sua futura umidade.
- A planta 2 foi a única que obteve um resultado melhor usando atributos de umidade de outra planta. Além disso, foi a que possuiu mais atributos selecionados.
- Nas plantas 1 e 2, a distinção entre a linha de base e o modelo é bastante sensível. Já na planta 3, a otimização teve efeito mínimo.
- A temperatura foi um atributo útil em todas as otimizações, enquanto a umidade do ar foi útil apenas na otimização da planta 3 e o desvio-padrão do índice UV apenas foi útil para a planta 2.
- Os atributos Umidade1_V_MEDIA12H, UmidadeAr_MEDIA12H, Umidade1_V_DP12H e UmidadeAr_DP12H não foram selecionados em nenhum dos casos.
- Os processos gaussianos foram o pior resultado das plantas 1 e 3, enquanto a *MultilayerPerceptron* foi o pior resultado da planta 2.
- Houve uma igualdade de resultados entre a *LinearRegression* e a *M5P* para o Caso A.

- O processo de otimização dos algoritmos envolveu muitos testes manuais, mas provavelmente há formas de automatizar isso.

Desse modo, é possível citar como alguns *insights* no processo de criação de conhecimento dentro da Internet of Things:

- A adequação da Mineração de Dados para a tarefa de criação de conhecimento.
- A possibilidade de casos em que um mesmo *dataset* gera um mesmo resultado ou muito parecidos para dois algoritmos diferentes.
- A possibilidade de automatizar a otimização dos algoritmos de Mineração de Dados.

5.5 Limitações

Esse projeto possui várias limitações. Uma delas é a falta de recursos financeiros e de tempo para analisar outros estudos. Que outros estudos aplicaram a mesma técnica e que resultados obtiveram? Que variáveis eles coletaram das plantas e do ambiente? Que instrumentos eles usaram na medição e no processamento? Quais os intervalos entre uma leitura e outra?

Uma segunda limitação é a qualidade dos modelos de previsão criados. Embora o modelo tenha produzido bons resultados com um *dataset* de treino e um *dataset* de teste distintos, a hipótese de que os modelos ficaram muito acoplados a esses *datasets* não está descartada. Seriam necessárias outras verificações com *datasets* mais variados (com dados coletados durante mais tempo e com maior variação nas condições ambientais) para certificar que o modelo faz boas previsões.

Uma terceira limitação é a espécie das plantas. Se o estudo fosse feito com outras plantas, o resultado seria diferente? As espécies das plantas influenciaram em algum fator nos atributos que foram selecionados? Além disso, também não foi possível correlacionar o porte das plantas com as umidades obtidas, nem descobrir a causa do *M5P* e da *LinearRegression* terem o mesmo resultado no Caso A.

Uma quarta limitação na simulação das condições para criação de conhecimento na Internet of Things é que ele cria o conhecimento a partir dos dados armazenados, não dos dados em trânsito, como é a regra.

Uma última limitação foi o processo. A otimização foi feita manualmente e envolveu juízo pessoal, então um resultado melhor pode ter sido perdido.

Conclusão

Esse trabalho buscou encontrar *insights* na criação de conhecimento dentro das aplicações de Internet of Things e Internet of Everything, que foi o pressuposto de uma lacuna de pesquisa. Para isso, buscou relacionar Mineração de Dados com essa área e desenvolver um projeto prático que integrasse Criação do Conhecimento, Internet of Things e Mineração de Dados.

Os objetivos foram parcialmente atingidos. Confirmou-se a aplicabilidade de Mineração de Dados para a criação de conhecimento dentro da Internet of Everything, pois foi possível encontrar informações e padrões relevantes nos dados, além de alguns *insights* nas características da criação de conhecimento dentro das aplicações de Internet of Things e Internet of Everything. No entanto, embora os modelos de previsão encontrados tenham baixa taxa de erros, eles não são totalmente confiáveis por não levarem em conta dados mais variados.

Estudos futuros serão necessários para mitigar uma ou mais das limitações desse projeto, especialmente quanto à primeira (comparação com outros estudos) e à segunda (qualidade dos modelos de previsão).

Referências

AHMED, Hanan et al. Introduction to Industrial Internet of Things (IIoT). *In*: JHA, Sudan et al. (org.) **Industrial Internet of Things: Technologies, Design, and Applications**. Boca Raton (Flórida), 2022, n.p. Disponível parcialmente em: [https://www.google.com.br/books/edition/Industrial Internet of Things/ruNhEAAAQB_AJ?hl=pt-BR](https://www.google.com.br/books/edition/Industrial_Internet_of_Things/ruNhEAAAQB_AJ?hl=pt-BR). Acesso em: 1 dez. 2022

ALISNEAKY. Kernel machines are used to compute a non-linearly separable functions into a higher dimension linearly separable function. **Wikimedia Commons**, [S. l.], 17 abr. 2011. Disponível em: [https://commons.wikimedia.org/wiki/File:Kernel Machine.png](https://commons.wikimedia.org/wiki/File:Kernel_Machine.png). Acesso em: 7 dez. 2022.

AMO, Sandra de. **Técnicas de Mineração de Dados**, [S. l.], 2004. Disponível em: [https://www.researchgate.net/publication/260300816 Tecnicas de Mineracao de D ados](https://www.researchgate.net/publication/260300816_Tecnicas_de_Mineracao_de_Dados). Acesso em: 4 dez. 2022.

BERLAND. Illustration of linear regression on a data set. **Wikimedia Commons**, [S. l.], 24 jun. 2007. Disponível em: <https://commons.wikimedia.org/wiki/File:LinearRegression.svg>. Acesso em 7 dez. 2022.

BIRTH of a legend. Forbes, [S. l.], 6 abr. 1998. Disponível em: www.forbes.com/forbes/1998/0406/6107128s1.html?sh=1a554e3048cf. Acesso em: 12 nov. 2022.

BOUACHIR, Ouns et al. Blockchain based solutions for achieving secure storage in fog computing. *In*: REHAN, Muhammad Maaz; REHMANI, Mubashir Husain. **Blockchain-enabled Fog and Edge Computing: Concepts, Architectures and Applications**. Boca Raton (Flórida), 2020, n.p. Disponível parcialmente em: [https://www.google.com.br/books/edition/Secure Edge Computing/fX01EAAAQBAJ?hl=pt-BR](https://www.google.com.br/books/edition/Secure_Edge_Computing/fX01EAAAQBAJ?hl=pt-BR). Acesso em: 2 dez. 2022.

BREIMAN, Leo. Random Forests. **Machine Learning**, [S. l.], v. 45, p. 5-32, out. 2001. Disponível em:

<https://link.springer.com/article/10.1023/a:1010933404324#citeas>. Acesso em: 5 dez. 2022.

CAMILO, Cássio Oliveira; SILVA, João Carlos da. **Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas**, [S. l.], ago. 2009. Disponível em: <https://ww2.inf.ufg.br/node/355>. Acesso em: 5 dez. 2022.

COSTA, Viviane Cunha Farias da. **A Knowledge-based approach to support Internet of Everything (IoE) lifestyle**. Orientador: Dr. Jano Moreira de Souza. 2022. 201 p. Tese (doutorado) – Curso de Ciências em Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2022. Disponível em: <https://cos.ufrj.br/index.php/pt-BR/publicacoes-pesquisa/details/15/3039>. Acesso em: 13 nov. 2022.

EVANS, Dave. The Internet of Things - How the Next Evolution of the Internet Is Changing Everything. **Cisco Internet Business Solutions Group (IBSG)**, [S. l.], abr. 2011. Disponível em: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf. Acesso em: 1 dez. 2022.

EVANS, Dave. How the Internet of Everything Will Change the World...for the Better #IoE [Infographic]. **Cisco Blog**, [S. l.], 7 nov. 2012. Disponível em: <https://web.archive.org/web/20121108200300/http://blogs.cisco.com/news/how-the-internet-of-everything-will-change-the-worldfor-the-better-infographic/>. Acesso em: 26 nov. 2022.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. From Data Mining to Knowledge Discovery in Databases. **AI Magazine**, [S. l.], v. 17, n. 3, p. 37-54, mar. 1996. Disponível em: <https://ojs.aaai.org/index.php/aimagazine/article/view/1230>. Acesso em: 5 dez. 2022.

FERREIRA, Eric de Paula et al. Gestão do conhecimento, internet das coisas e inovação: a relação dos temas e a intensidade de pesquisas realizadas. **Navus**, Florianópolis, v. 8, n. 3, p. 99-112, jul./set. 2018. Disponível em: <https://dialnet.unirioja.es/servlet/articulo?codigo=6570269>. Acesso em: 15 jun. 2022.

HASAN, Mohammad. State of IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally. **IoT Analytics**, [S. l.], 18 maio 2022. Disponível em: <https://iot-analytics.com/number-connected-iot-devices/>. Acesso em 1 dez. 2022.

KOSMATOS, Evangelos A.; Tselikas, Nikolaos D.; Boucouvalas, Anthony C. Integrating RFIDs and Smart Objects into a Unified Internet of Things Architecture. **Advances in Internet of Things**, [S. l.], v. 1, n. 1, p. 5-12, abr. 2011. Disponível em: <https://www.scirp.org/journal/paperinformation.aspx?paperid=4696>. Acesso em: 1 dez. 2022.

MADAKAM, Sommaya; RAMASWAMY, R.; TRIPATHI, Siddharth. Internet of Things (IoT): A Literature Review. **Journal of Computer and Communications**, Mumbai, v. 3, n. 5, maio, 2015. Disponível em: https://www.scirp.org/html/56616_56616.htm. Acesso em: 30 nov. 2022.

MADDEN, Sam. From Databases to Big Data. **IEEE Internet Computing**, [S. l.], v. 16, n. 3, p. 4-6, maio-junho, 2012. Disponível em: <https://ieeexplore.ieee.org/document/6188576>. Acesso em 29 nov. 2022

MALLADI, Ravisankar et al. **Data Mining: A Future Scope**. Bhopal (Índia): Nitya Publications, 2020. Disponível parcialmente em: https://www.google.com.br/books/edition/Data_Mining_A_Future_Scope/RrE0EAAAQBAJ?hl=pt-BR. Acesso em: 5 dez. 2022.

MARQUES JÚNIOR, Euro. Gestão do Conhecimento – Aula 01 - Fundamentos da Gestão do Conhecimento. [S. l.: s. n.], 2017, 1 vídeo (17 min). Publicado pelo canal UNIVESP. Disponível em: <https://www.youtube.com/watch?v=sriEhPFwKj0>. Acesso em 15 jun. 2022.

NAVARRO, Emerson; COSTA, Nuno; PEREIRA, António. A Systematic Review of IoT Solutions for Smart Farming. **Sensors**, [S. l.], v. 20, n. 15, n.p., jul. 2020. Disponível em: <https://www.mdpi.com/1424-8220/20/15/4231>. Acesso em 1 dez. 2022.

NASER, M. Z.; ALAVI, Amir H. Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences. **Architecture, Structures and Construction**, 19 p., nov. 2021. Disponível em: <https://link.springer.com/article/10.1007/s44150-021-00015-8>. Acesso em: 5 dez. 2022.

NONAKA, Ikujiro; TAKEUCHI, Hirotaka. Prefácio. *In*: _____. (org.) **Gestão do conhecimento**. Bookman, Porto Alegre, 2008a, p. IX-XIII.

NONAKA, Ikujiro; TAKEUCHI, Hirotaka. Criação e Dialética do Conhecimento. *In*: _____. (org.) **Gestão do conhecimento**. Bookman, Porto Alegre, 2008b, p. 17-38.

NONAKA, Ikujiro; TAKEUCHI, Hirotaka. Teoria da Criação do Conhecimento Organizacional. *In*: _____. (org.) **Gestão do conhecimento**. Bookman, Porto Alegre, 2008c, p. 54-90.

PHYSIKINGER. Gaussian process with smooth signal. **Wikimedia Commons**, [S. l.], 5 fev. 2017. Disponível em: https://commons.wikimedia.org/wiki/File:Gaussianprocess_smooth.png. Acesso em: 7 dez. 2022.

QI, Qinglin; TAO, Fei. A Smart Manufacturing Service System Based on Edge Computing, Fog Computing, and Cloud Computing. **IEEE Access**, [S. l.], v. 7, p. 86769-86777, 19 jun. 2019. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8740963>. Acesso em: 2 dez. 2022.

RAUBER, Thomas Walter. Redes Neurais Artificiais. [S. l.], 2005. Disponível em: https://www.researchgate.net/publication/228686464_Redets_neurais_artificiais. Acesso em 6 dez. 2022

RCLONE. [Site institucional]. Disponível em: <https://rclone.org/>. Acesso em: 23 out. 2022.

SAITO, André. Gestão do Conhecimento - Aula 02 - Aprendizagem nas Organizações. [Entrevista concedida a] Euro Marques Júnior. [S. l.: s. n.], 2017. 1 vídeo (24 min), trecho entre 11:43 e 22:52. Publicado pelo canal UNIVESP. Disponível em: <https://www.youtube.com/watch?v=Ui3sCyngS7g>. Acesso em 15 jun. 2022.

SALMERÓN-MANZANO et al. Worldwide Research Trends on Smart Homes. *In*: RAHMANI-ANDEBILI, Mehdi (org.). **Operation of Smart Homes**. New York: Springer, 2021, p. 1-32. Disponível parcialmente em: https://www.google.com.br/books/edition/Operation_of_Smart_Homes/0VlzEAAAQB_AJ?hl=pt-BR. Acesso em: 1 dez. 2022.

SCHRÖER, Christoph; KRUSE, Felix; GÓMEZ, Jorge Marx. A Systematic Literature Review on Applying CRISP-DM Process Model. **Procedia Computer Science**, [S. l.], v. 181, p. 526-534, 2021. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050921002416>. Acesso em: 5 dez. 2022.

SETZER, Valdemar Waingort. **Dado, Informação, Conhecimento e Competência**. [S. l.], 25 maio 2015. Disponível em: <https://www.ime.usp.br/~vwsetzer/dado-info.html>. Acesso em 30 nov. 2022.

SLOW, Eugene; TIROPANIS, Thanassis; HALL, Wendy. Analytics for the Internet of Things: A Survey. **ACM Computing Surveys**, [S. l.], v. 51, n. 4, 36 p., jul. 2018. Disponível em: <https://dl.acm.org/doi/10.1145/3204947>. Acesso em 26 nov. 2022.

SUNFOUNDER. **GYML8511 UV Sensor**. [S. l.], 13 nov. 2019. Disponível em: http://wiki.sunfounder.cc/index.php?title=GYML8511_UV_Sensor. Acesso em: 30 jul. 2022.

WHAT IS ARDUINO? **Arduino.cc**. [S. l.], 5 fev. 2018. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em 2 dez. 2022.

WANG, Yong; WITTEN, Ian Hugh. **Induction of model trees for predicting continuous classes**. [S. l.], 1996. Disponível em: https://www.researchgate.net/publication/33051395_Induction_of_model_trees_for_predicting_continuous_classes. Acesso em 5 dez. 2022.

WEKA. [Site institucional]. Disponível em: <https://www.cs.waikato.ac.nz/ml/weka/>. Acesso em: 3 nov. 2022.

WITTEN, Ian Hugh; FRANK, Eibe; HALL, Mark. **Data Mining: Practical Machine Learning Tools and Techniques**. Burlington (Massachusetts): Elsevier, 2011. Disponível parcialmente em: https://www.google.com.br/books/edition/Data_Mining/bDtLM8CODsQC?hl=pt-BR. Acesso em 5 dez. 2022.

YI, Wei. Understanding Gaussian Process, the Socratic Way. **Towards Data Science**. [S. l.], 1 dez. 2019. Disponível em:

<https://towardsdatascience.com/understanding-gaussian-process-the-socratic-way-ba02369d804>. Acesso em: 6 dez. 2022.

Anexos

Observação: Todos os códigos em anexo foram escritos pelo autor e podem ser usados segundo a MIT License (<https://opensource.org/licenses/MIT>), sem necessidade de mencionar o autor.

Anexo 1 - Script usado no Arduino

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define pinoUmidade1 A0
#define pinoUmidade2 A1
#define pinoUmidade3 A2
#define pinoDHT11 2
#define pinoUV A3
#define pinoUV_REF A4
#define delayPadrao 900

int umidade1_V = 0;
int umidade2_V = 0;
int umidade3_V = 0;
float indiceUV_REF = 0;
float indiceUV_V = 0;
float outputVoltage = 0;
float indiceUV = 0;

uint32_t tempoMS = 0;

DHT_Unified dht(pinoDHT11, DHT11);

void setup() {
  Serial.begin(9600);

  dht.begin();
}

void loop() {
  int umidade1_V = analogRead(pinoUmidade1);
  Serial.print(umidade1_V);
  Serial.print(",");
```

```
int umidade2_V = analogRead(pinoUmidade2);
Serial.print(umidade2_V);
Serial.print(",");

int umidade3_V = analogRead(pinoUmidade3);
Serial.print(umidade3_V);
Serial.print(",");

sensors_event_t event;
dht.temperature().getEvent(&event);
if (isnan(event.temperature)) {
    Serial.print("-1,");
} else {
    Serial.print(event.temperature);
    Serial.print(",");
}

dht.humidity().getEvent(&event);
if (isnan(event.relative_humidity)) {
    Serial.print("-1,");
} else {
    Serial.print(event.relative_humidity);
    Serial.print(",");
}

indiceUV_V = analogRead(pinoUV);
indiceUV_REF = analogRead(pinoUV_REF);

outputVoltage = 3.3 / indiceUV_REF * indiceUV_V;

indiceUV = (outputVoltage - 0.99) * (15.0 - 0.0) / (2.8 - 0.99) +
0.0;

Serial.print(indiceUV_V);
Serial.print(",");
```

```
Serial.print(indiceUV_REF);  
Serial.print(",");  
  
Serial.print(outputVoltage);  
Serial.print(",");  
  
Serial.print(indiceUV);  
Serial.print(",");  
  
Serial.print(millis());  
Serial.print(",");  
  
Serial.println();  
  
delay(delayPadrao);  
}
```

Anexo 2 - Script Python para gravar dados enviados pelo Arduino

```

import serial

from datetime import datetime

portaArduino = '/dev/ttyACM0'

baud = 9600

dataHoraAtual = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")

caminhoArquivo = '/root/leiturasLocais/'

nomeArquivo = 'leituraArduino_' + dataHoraAtual

dadosPorArquivo = 700

labels=
"Umidade1_V,Umidade2_V,Umidade3_V, Temperatura,UmidadeAr,IndiceUV_V,I
ndiceUV_Ref,outputVoltage,indiceUV,TempoMs,DataHora\n"

printedLabels = False

counter = 0

ser = serial.Serial(portaArduino, baud, timeout=10)

arquivo = open(caminhoArquivo + nomeArquivo, "a")

while ser.isOpen():
    if not printedLabels:
        arquivo.write(labels)
        printedLabels = True

    dataHoraAtual = datetime.now().strftime("%Y/%m/%d_%H:%M:%S")

    dadosComQuebraDeLinha= str(ser.readline(), 'utf-8')

    dados = dadosComQuebraDeLinha.replace('\r',
    '').replace('\n','')

    arquivo.write(dados + dataHoraAtual + "\n")

    counter += 1

```

```
if (counter >= dadosPorArquivo):  
    arquivo.close()  
    counter = 0  
    dataHoraAtual =  
datetime.now().strftime("%Y-%m-%d_%H-%M-%S")  
    nomeArquivo = 'leituraArduino_' + dataHoraAtual  
    arquivo = open(caminhoArquivo + nomeArquivo, "a")  
    printedLabels = False
```

Anexo 3 - Script Python para enviar os dados para a nuvem

```
import os
import re
import subprocess
from datetime import datetime

path = "/root/leiturasLocais/"
pathLog = "/root/Logs/"
remoteRClone = "Arduino"
pathRemoto = "/leiturasRemotas"

log_rclone = open(pathLog + "logRClone" + ".txt", "a")

try:
    subprocess.run(["rclone", "sync", path, remoteRClone + ":" +
pathRemoto])
    hora = datetime.now()
    hora_formatada = hora.strftime('%d/%m/%Y %H:%M:%S')
    log_rclone.write("Copiado com sucesso as " + hora_formatada +
"\n")
except:
    hora = datetime.now()
    hora_formatada = hora.strftime('%d/%m/%Y %H:%M:%S')
    log_rclone.write("Erro ao fazer a copia as " + hora_formatada +
"\n")

log_rclone.close()
```

Anexo 4 - Script Python para agregar arquivos com dados do mesmo dia

```
from os import listdir
from os.path import isfile, join
import csv
#import argparse

def salvaLinhasDiaArquivo(diaParametro, arquivoSerLidoPath,
arquivoSerEscritoPath):
    arquivoSerLido = open(arquivoSerLidoPath, "r")
    arquivoSerEscrito = open(arquivoSerEscritoPath, "a")

    csv_arquivoSerLido = csv.reader(arquivoSerLido)
    csv_arquivoSerEscrito = csv.writer(arquivoSerEscrito)

    for linha in csv_arquivoSerLido:
        dataHora = linha[len(linha)-1].split('_')
        data = dataHora[0].replace('/', '-')
        if (data == diaParametro):
            csv_arquivoSerEscrito.writerow(linha)

    arquivoSerLido.close()
    arquivoSerEscrito.close()

minhasDatas = ['2022-10-23', '2022-10-24', '2022-10-25', '2022-10-
26', '2022-10-27', '2022-10-28', '2022-10-29', '2022-10-30', '2022-
10-31', '2022-11-01', '2022-11-02', '2022-11-03', '2022-11-04',
'2022-11-05', '2022-11-06', '2022-11-07', '2022-11-08', '2022-11-
09', '2022-11-10', '2022-11-11']

arquivosTodos = listdir()
arquivosTodos.sort()
```

```

for i in range(1, len(minhasDatas)):
    dataAnterior = minhasDatas[i-1]
    dataAtual = minhasDatas[i]

    ultimoArquivo = ''
    arquivosSeraoAnalisados = []

    for arquivo in arquivosTodos:
        if (arquivo.find('leituraArduino') > -1):
            if (arquivo.find(dataAnterior) > -1):
                ultimoArquivo = arquivo
            elif (arquivo.find(dataAtual) > -1):
                arquivosSeraoAnalisados.append(arquivo)

    arquivoDataAtual = 'leituraDia_' + dataAtual
    with open(arquivoDataAtual, "w") as arquivo:
        cabecalhos=
["Umidade1_V", "Umidade2_V", "Umidade3_V", "Temperatura", "UmidadeAr", "I
ndiceUV_V", "IndiceUV_Ref", "outputVoltage", "indiceUV", "TempoMs", "Data
Hora"]

        csv_arquivoSerEscrito = csv.writer(arquivo)
        csv_arquivoSerEscrito.writerow(cabecalhos)

    if (ultimoArquivo != ''):
        salvaLinhasDiaArquivo(dataAtual, ultimoArquivo,
arquivoDataAtual)

    for arquivo in arquivosSeraoAnalisados:
        salvaLinhasDiaArquivo(dataAtual, arquivo, arquivoDataAtual)

```

Anexo 5 - Script Python para detectar erros nos arquivos

```
import csv

diaEscolhido = ""

parametrosAvaliados = {
    'Umidade1_V': {'min': 300, 'max': 1000, 'type': int},
    'Umidade2_V': {'min': 300, 'max': 1000, 'type': int},
    'Umidade3_V': {'min': 300, 'max': 1000, 'type': int},
    'Temperatura': {'min': 000, 'max': 50.00, 'type': float},
    'UmidadeAr': {'min': 0.00, 'max': 99.00, 'type': float},
    'indiceUV': {'min': -0.26, 'max': 15, 'type': float}
}

erros = 0
linhasTotais = 0
problemasEmSequencia = []

with open("leituraDia_" + diaEscolhido, "r") as arquivoAnalisado:
    csv_arquivoAnalisado = csv.DictReader(arquivoAnalisado)
    problema = []
    next(csv_arquivoAnalisado)
    for linha in csv_arquivoAnalisado:
        linhasTotais += 1
        for parametro in parametrosAvaliados:
            valorAvaliado = 0
            if parametrosAvaliados[parametro]['type'] == 'int':
                valorAvaliado = int(linha[parametro])
            else:
                valorAvaliado = float(linha[parametro])
```

```
                if (valorAvaliado < parametrosAvaliados[parametro]['min']
or
                    valorAvaliado >
parametrosAvaliados[parametro]['max']):
                    print(linha, end="")
                    print(" -> " + parametro + " = " +
str(valorAvaliado))
                    erros += 1
                    break

print("Foram encontradas {0} linhas".format(linhasTotais))
print("Foram encontrados {0} erros.".format(erros))
```

Anexo 6 - Script Python para remover atributos de controle

```

import csv

def eliminarInuteisDiaArquivo(diaParametro, arquivoSerLidoPath,
arquivoSerEscritoPath):

    arquivoSerLido = open(arquivoSerLidoPath, "r")
    arquivoSerEscrito = open(arquivoSerEscritoPath, "a")

    csv_arquivoSerLido = csv.DictReader(arquivoSerLido)
    csv_arquivoSerEscrito = csv.DictWriter(arquivoSerEscrito,
["Umidade1_V", "Umidade2_V", "Umidade3_V", "Temperatura", "UmidadeAr", "i
ndiceUV", "DataHora"])

    csv_arquivoSerEscrito.writeheader()

    for linha in csv_arquivoSerLido:
        outputVoltage = 3.3 / float(linha['IndiceUV_Ref']) *
float(linha['IndiceUV_V'])
        indiceUV = (outputVoltage - 0.99) * (15.0 - 0.0) / (2.8 -
0.99) + 0.0;
        del linha['IndiceUV_V']
        del linha['IndiceUV_Ref']
        del linha['outputVoltage']
        linha['indiceUV'] = indiceUV
        del linha['TempoMs']
        csv_arquivoSerEscrito.writerow(linha)
    arquivoSerLido.close()
    arquivoSerEscrito.close()

minhasDatas = ['2022-10-22', '2022-10-23', '2022-10-24', '2022-10-
25', '2022-10-26', '2022-10-27', '2022-10-28', '2022-10-29', '2022-
10-30', '2022-10-31', '2022-11-05', '2022-11-06', '2022-11-07',
'2022-11-09', '2022-11-10', '2022-11-11']

for diaEscolhido in minhasDatas:
    eliminarInuteisDiaArquivo(diaEscolhido, 'limpoLeituraDia_' +
        diaEscolhido, 'souteisLeituraDia_' + diaEscolhido)

```

Anexo 7 - Script Python para reduzir leituras de segundos para leituras de minuto

```

import csv

import statistics

def descobrirHoraMinuto(stringHoraMinutoSegundo):
    horaMinutoSegundo =
stringHoraMinutoSegundo.split("_")[1].split(":")
    return [int(horaMinutoSegundo[0]), int(horaMinutoSegundo[1])]

def descobrirDiaString(stringHoraMinutoSegundo):
    return stringHoraMinutoSegundo.split("_")[0]

def formatarDiaHoraMinuto(stringDia, vetorHoraMinutoSegundo):
    return stringDia + "_" + f"{vetorHoraMinutoSegundo[0]:02}" + ":" +
+ f"{vetorHoraMinutoSegundo[1]:02}"

def calcularMediasSegundo(vetorLinhasHoraMinutoSegundo):
    if (vetorLinhasHoraMinutoSegundo != [] and
len(vetorLinhasHoraMinutoSegundo) >= 45 and
len(vetorLinhasHoraMinutoSegundo) <= 75):
        keys = ["Umidade1_V", "Umidade2_V", "Umidade3_V",
"Temperatura", "UmidadeAr", "indiceUV"]
        dictListas = {"Umidade1_V": [], "Umidade2_V": [],
"Umidade3_V": [], "Temperatura": [], "UmidadeAr": [], "indiceUV":
[]}
        for linha in vetorLinhasHoraMinutoSegundo:

dictListas["Umidade1_V"].append(int(linha['Umidade1_V']))

dictListas["Umidade2_V"].append(int(linha['Umidade2_V']))

dictListas["Umidade3_V"].append(int(linha['Umidade3_V']))

dictListas["Temperatura"].append(float(linha['Temperatura']))

dictListas["UmidadeAr"].append(float(linha['UmidadeAr']))

```

```

        dictListas["indiceUV"].append(float(linha['indiceUV']))
    mediasDictLists = {"Umidade1_V": 0, "Umidade2_V": 0,
"Umidade3_V": 0, "Temperatura": 0.0, "UmidadeAr": 0.0, "indiceUV":
0.0}

    for key in keys:
        if type(mediasDictLists[key]) == int:
            media = statistics.mean(dictListas[key])
            mediasDictLists[key] = round(media)
        else:
            media = statistics.mean(dictListas[key])
            mediasDictLists[key] = round(media, 2)
    return mediasDictLists
return {}

def reduzParaHoraMinuto(arquivoSerLidoPath, arquivoSerEscritoPath):
    arquivoSerLido = open(arquivoSerLidoPath, "r")
    arquivoSerEscrito = open(arquivoSerEscritoPath, "a")

    csv_arquivoSerLido = csv.DictReader(arquivoSerLido)
    csv_arquivoSerEscrito = csv.DictWriter(arquivoSerEscrito,
["Umidade1_V", "Umidade2_V", "Umidade3_V", "Temperatura", "UmidadeAr", "i
ndiceUV", "DataHora"])

    csv_arquivoSerEscrito.writeheader()

    dadosDoDia = []
    horaMinutoAnterior = []
    for linha in csv_arquivoSerLido:
        horaMinutoAnterior = descobrirHoraMinuto(linha['DataHora'])
        break
    for linha in csv_arquivoSerLido:
        horaMinutoObtida = descobrirHoraMinuto(linha['DataHora'])
        if (horaMinutoAnterior == horaMinutoObtida):
            dadosDoDia.append(linha)

```

```

else:
    medias = calcularMediasSegundo(dadosDoDia)
    if (medias != {}):
        diaAtual = descobrirDiaString(linha['DataHora'])
        medias['DataHora'] = formatarDiaHoraMinuto(diaAtual,
horaMinutoAnterior)
        csv_arquivoSerEscrito.writerow(medias)
    dadosDoDia = []
    dadosDoDia.append(linha)
    horaMinutoAnterior = horaMinutoObtida
medias = calcularMediasSegundo(dadosDoDia)
if (medias != {}):
    diaAtual = descobrirDiaString(linha['DataHora'])
    medias['DataHora'] = formatarDiaHoraMinuto(diaAtual,
horaMinutoAnterior)
    csv_arquivoSerEscrito.writerow(medias)

arquivoSerLido.close()
arquivoSerEscrito.close()

minhasDatas = ['2022-10-22', '2022-10-23', '2022-10-24', '2022-10-
25', '2022-10-26', '2022-10-27', '2022-10-28', '2022-10-29', '2022-
10-30', '2022-10-31', '2022-11-05', '2022-11-06', '2022-11-07',
'2022-11-09', '2022-11-10', '2022-11-11']

for diaEscolhido in minhasDatas:
    reduzParaHoraMinuto('souteisLeituraDia_' + diaEscolhido,
'minutoDia_' + diaEscolhido)

```

Anexo 8 - Script Python para reduzir leituras de minutos para leituras de hora

```

import csv

import statistics

def descobrirHora(stringHoraMinuto):
    horaMinuto = stringHoraMinuto.split("_")[1].split(":")
    return int(horaMinuto[0])

def descobrirDiaString(stringHoraMinuto):
    return stringHoraMinuto.split("_")[0]

def calcularMediasMinuto(vetorLinhasHoraMinuto):
    if (vetorLinhasHoraMinuto != [] and
        len(vetorLinhasHoraMinuto) >= 45 and len(vetorLinhasHoraMinuto) <=
        75):
        keys = ["Umidade1_V", "Umidade2_V", "Umidade3_V",
                "Temperatura", "UmidadeAr", "indiceUV"]
        dictListas = {"Umidade1_V": [], "Umidade2_V": [],
                      "Umidade3_V": [], "Temperatura": [], "UmidadeAr": [], "indiceUV":
                      []}
        for linha in vetorLinhasHoraMinuto:
            dictListas["Umidade1_V"].append(int(linha['Umidade1_V']))
            dictListas["Umidade2_V"].append(int(linha['Umidade2_V']))
            dictListas["Umidade3_V"].append(int(linha['Umidade3_V']))
            dictListas["Temperatura"].append(float(linha['Temperatura']))
            dictListas["UmidadeAr"].append(float(linha['UmidadeAr']))
            dictListas["indiceUV"].append(float(linha['indiceUV']))
        mediasDictLists = {"Umidade1_V": 0, "Umidade2_V": 0,
                           "Umidade3_V": 0, "Temperatura": 0.0, "UmidadeAr": 0.0, "indiceUV":
                           0.0}
        for key in keys:
            if type(mediasDictLists[key]) == int:

```

```

        media = statistics.mean(dictListas[key])
        mediasDictLists[key] = round(media)
    else:
        media = statistics.mean(dictListas[key])
        mediasDictLists[key] = round(media, 2)
    return mediasDictLists
return {}

def reduzParaHora(arquivoSerLidoPath, arquivoSerEscritoPath):
    arquivoSerLido = open(arquivoSerLidoPath, "r")
    arquivoSerEscrito = open(arquivoSerEscritoPath, "a")

    csv_arquivoSerLido = csv.DictReader(arquivoSerLido)
    csv_arquivoSerEscrito = csv.DictWriter(arquivoSerEscrito,
["Umidade1_V", "Umidade2_V", "Umidade3_V", "Temperatura", "UmidadeAr", "i
ndiceUV", "Data", "Hora"])

    csv_arquivoSerEscrito.writeheader()

    dadosDoDia = []
    horaAnterior = []
    for linha in csv_arquivoSerLido:
        horaAnterior = descobrirHora(linha['DataHora'])
        break
    for linha in csv_arquivoSerLido:
        horaObtida = descobrirHora(linha['DataHora'])
        if (horaAnterior == horaObtida):
            dadosDoDia.append(linha)
        else:
            medias = calcularMediasMinuto(dadosDoDia)
            if (medias != {}):
                diaAtual = descobrirDiaString(linha['DataHora'])
                medias['Data'] = diaAtual

```

```
        medias['Hora'] = horaAnterior
        csv_arquivoSerEscrito.writerow(medias)
    dadosDoDia = []
    dadosDoDia.append(linha)
    horaAnterior = horaObtida
medias = calcularMediasMinuto(dadosDoDia)
if (medias != {}):
    diaAtual = descobrirDiaString(linha['DataHora'])
    medias['Data'] = diaAtual
    medias['Hora'] = horaAnterior
    csv_arquivoSerEscrito.writerow(medias)

arquivoSerLido.close()
arquivoSerEscrito.close()

minhasDatas = ['2022-10-22', '2022-10-23', '2022-10-24', '2022-10-
25', '2022-10-26', '2022-10-27', '2022-10-28', '2022-10-29', '2022-
10-30', '2022-10-31', '2022-11-05', '2022-11-06', '2022-11-07',
'2022-11-09', '2022-11-10', '2022-11-11']

for diaEscolhido in minhasDatas:
    reduzParaHora('minutoDia_' + diaEscolhido, 'horaDia_' +
diaEscolhido)
```

Anexo 9 - Script Python para unificar as instâncias em um único *dataset*

```
from os import listdir
from os.path import isfile, join
import csv

arquivosTodos = listdir()
arquivosTodos.sort()

arquivosSeraoAnalisados = []

for arquivo in arquivosTodos:
    ultimoArquivo = ''
    if (arquivo.find('horaDia_') > -1):
        arquivosSeraoAnalisados.append(arquivo)

arquivoSerEscrito = open('datasetUnificado', "a")
cabecalhos =
["Umidade1_V", "Umidade2_V", "Umidade3_V", "Temperatura", "UmidadeAr", "i
ndiceUV", "Data", "Hora"]
csv_arquivoSerEscrito = csv.DictWriter(arquivoSerEscrito,
cabecalhos)
csv_arquivoSerEscrito.writeheader()

for arquivo in arquivosSeraoAnalisados:
    arquivoSerLido = open(arquivo, "r")

    csv_arquivoSerLido = csv.DictReader(arquivoSerLido)

    for linha in csv_arquivoSerLido:
        csv_arquivoSerEscrito.writerow(linha)

    arquivoSerLido.close()
arquivoSerEscrito.close()
```

Anexo 10 - Script Python para adicionar médias e desvios-padrão das 12 horas anteriores

```

from datetime import date, timedelta

import csv

import statistics

def isMomentoAnterior(linhaAnterior, linhaAtual,
quantidadeAnterior):

    horaAnterior = int(linhaAnterior["Hora"])

    horaAtual = int(linhaAtual["Hora"])

    dataAnterior =
date.fromisoformat(linhaAnterior["Data"].replace('/', '-'))
    dataAtual = date.fromisoformat(linhaAtual["Data"].replace('/',
'-'))

    if ((horaAtual == (horaAnterior + quantidadeAnterior)) and
dataAnterior == dataAtual):

        return True

    elif (((horaAnterior + quantidadeAnterior) % 24) == horaAtual)
and (dataAtual == (dataAnterior + timedelta(1))):

        return True

    return False

def ha12HorasAnteriores(linhasAnteriores, linhaAtual):

    for i in range(1, 12):

        if (not isMomentoAnterior(linhasAnteriores[i-1], linhaAtual,
i)):

            return False

    return True

refTipos = {"Umidade1_V": int, "Umidade2_V": int, "Umidade3_V": int,
"Temperatura": float, "UmidadeAr": float, "indiceUV": float}

def calcularMedia(linhas, chave):

```

```

valoresConvertidos = []
if (refTipos[chave] == int):
    for linha in linhas:
        valoresConvertidos.append(int(linha[chave]))
else:
    for linha in linhas:
        valoresConvertidos.append(float(linha[chave]))
if refTipos[chave] == int:
    media = statistics.mean(valoresConvertidos)
    media = round(media)
else:
    media = statistics.mean(valoresConvertidos)
    media = round(media, 2)
return media

def calcularDesvioPadrao(linhas, chave):
    valoresConvertidos = []
    for linha in linhas:
        valoresConvertidos.append(float(linha[chave]))
    desvioPadrao = statistics.stdev(valoresConvertidos)
    desvioPadrao = round(desvioPadrao, 2)
    return desvioPadrao

arquivoSerLido = open('datasetUnificado', "r")
csv_arquivoSerLido = csv.DictReader(arquivoSerLido)
cabecalhos =
["Umidade1_V", "Umidade2_V", "Umidade3_V", "Temperatura", "UmidadeAr", "i
ndiceUV",
"Umidade1_V_MEDIA12H", "Umidade2_V_MEDIA12H", "Umidade3_V_MEDIA12H", "T
emperatura_MEDIA12H", "UmidadeAr_MEDIA12H", "indiceUV_MEDIA12H", "Umida
de1_V_DP12H", "Umidade2_V_DP12H", "Umidade3_V_DP12H", "Temperatura_DP12
H", "UmidadeAr_DP12H", "indiceUV_DP12H", "Data", "Hora"]

arquivoSerEscrito = open('datasetCom12H', "a")

```

```
csv_arquivoSerEscrito = csv.DictWriter(arquivoSerEscrito,
cabecalhos)

csv_arquivoSerEscrito.writeheader()

linhasAnteriores = []
linhasCertas = 0

for linha in csv_arquivoSerLido:
    if (len(linhasAnteriores) > 12):
        print(linha)
        print("ALGO DEU ERRADO!")
        exit()

    if ((len(linhasAnteriores) == 12) and
hal2HorasAnteriores(linhasAnteriores, linha)):
        linhasCertas += 1

        linhaSerEscrita = {}
        linhaSerEscrita["Umidade1_V"] = linha["Umidade1_V"]
        linhaSerEscrita["Umidade2_V"] = linha["Umidade2_V"]
        linhaSerEscrita["Umidade3_V"] = linha["Umidade3_V"]
        linhaSerEscrita["Temperatura"] = linha["Temperatura"]
        linhaSerEscrita["UmidadeAr"] = linha["UmidadeAr"]
        linhaSerEscrita["indiceUV"] = linha["indiceUV"]

        linhaSerEscrita["Umidade1_V_MEDIA12H"] =
calcularMedia(linhasAnteriores, 'Umidade1_V')

        linhaSerEscrita["Umidade2_V_MEDIA12H"] =
calcularMedia(linhasAnteriores, 'Umidade2_V')

        linhaSerEscrita["Umidade3_V_MEDIA12H"] =
calcularMedia(linhasAnteriores, 'Umidade3_V')

        linhaSerEscrita["Temperatura_MEDIA12H"] =
calcularMedia(linhasAnteriores, 'Temperatura')

        linhaSerEscrita["UmidadeAr_MEDIA12H"] =
calcularMedia(linhasAnteriores, 'UmidadeAr')
```

```
        linhaSerEscrita["indiceUV_MEDIA12H"] =
calcularMedia(linhasAnteriores, 'indiceUV')

        linhaSerEscrita["Umidade1_V_DP12H"] =
calcularDesvioPadrao(linhasAnteriores, 'Umidade1_V')

        linhaSerEscrita["Umidade2_V_DP12H"] =
calcularDesvioPadrao(linhasAnteriores, 'Umidade2_V')

        linhaSerEscrita["Umidade3_V_DP12H"] =
calcularDesvioPadrao(linhasAnteriores, 'Umidade3_V')

        linhaSerEscrita["Temperatura_DP12H"] =
calcularDesvioPadrao(linhasAnteriores, 'Temperatura')

        linhaSerEscrita["UmidadeAr_DP12H"] =
calcularDesvioPadrao(linhasAnteriores, 'UmidadeAr')

        linhaSerEscrita["indiceUV_DP12H"] =
calcularDesvioPadrao(linhasAnteriores, 'indiceUV')

        linhaSerEscrita["Data"] = linha["Data"]
        linhaSerEscrita["Hora"] = linha["Hora"]

        csv_arquivoSerEscrito.writerow(linhaSerEscrita)

        linhasAnteriores.pop()
    elif (len(linhasAnteriores) == 12):
        linhasAnteriores = []
        linhasAnteriores.insert(0, linha)

print(linhasCertas)
arquivoSerLido.close()
arquivoSerEscrito.close()
```

Anexo 11 - Script para adicionar a classe

```

from datetime import date, timedelta
import csv

def isMomentoPosterior12Horas(linhaAtual, linhaPosterior):
    horaAtual = int(linhaAtual["Hora"])
    horaPosterior = int(linhaPosterior["Hora"])

    dataAtual = date.fromisoformat(linhaAtual["Data"].replace('/',
    '-'))
    dataPosterior =
date.fromisoformat(linhaPosterior["Data"].replace('/', '-'))

    if ((horaAtual == (horaPosterior - 12)) and dataAtual ==
dataPosterior):
        return True

    elif (((horaAtual + 12) % 24) == horaPosterior) and
(dataPosterior == (dataAtual + timedelta(1))):
        return True

    return False

arquivoSerLido = open('datasetCom12H', "r")
csv_arquivoSerLido = csv.DictReader(arquivoSerLido)
atributoPrevisto = "Umidade1_V"
classe = "Umidade1_V_CLASSE"

cabecalhos =
["Umidade1_V", "Umidade2_V", "Umidade3_V", "Temperatura", "UmidadeAr", "i
ndiceUV",
"Umidade1_V_MEDIA12H", "Umidade2_V_MEDIA12H", "Umidade3_V_MEDIA12H", "T
emperatura_MEDIA12H", "UmidadeAr_MEDIA12H", "indiceUV_MEDIA12H", "Umida
de1_V_DP12H", "Umidade2_V_DP12H", "Umidade3_V_DP12H", "Temperatura_DP12
H", "UmidadeAr_DP12H", "indiceUV_DP12H", "Data", "Hora", classe]

linhasAnteriores = []

```

```

for linha in csv_arquivoSerLido:
    if (len(linhasAnteriores) < 12):
        linhasAnteriores.append(linha)
    else:
        break

linhasSemPosterior = 0

linhaSerEscrita = {}

arquivoSerEscrito = open('datasetComClasse', "a")
csv_arquivoSerEscrito = csv.DictWriter(arquivoSerEscrito,
cabecalhos)
csv_arquivoSerEscrito.writeheader()

for linhaPosterior in csv_arquivoSerLido:
    haAnterior = False
    for linha in linhasAnteriores:
        if (isMomentoPosterior12Horas(linha, linhaPosterior)):
            linhaSerEscrita = linha
            haAnterior = True
            break
    if (haAnterior):
        linhaSerEscrita[classe] = linhaPosterior[atributoPrevisto]
        csv_arquivoSerEscrito.writerow(linhaSerEscrita)
    else:
        linhasSemPosterior += 1
    linhasAnteriores.append(linhaPosterior)
    linhasAnteriores.pop(0)
    if (len(linhasAnteriores) > 12):
        print("ALGO DEU ERRADO!")
        print(linhaPosterior)
        exit()

```

```
print("Linhas sem posterior de 12 horas na frente: ",  
linhasSemPosterior)  
  
arquivoSerLido.close()  
  
arquivoSerEscrito.close()
```

Anexo 12 - Amostra do *dataset* de treino da planta 1

Umidade1_V, Umidade2_V, Umidade3_V, Temperatura, UmidadeAr, indiceUV, Umidade1_V_MEDIA12H, Umidade2_V_MEDIA12H, Umidade3_V_MEDIA12H, Temperatura_MEDIA12H, UmidadeAr_MEDIA12H, indiceUV_MEDIA12H, Umidade1_V_DP12H, Umidade2_V_DP12H, Umidade3_V_DP12H, Temperatura_DP12H, UmidadeAr_DP12H, indiceUV_DP12H, Data, Hora, Umidade1_V_CLASSE

593, 491, 526, 18.61, 75.86, 0.04, 590, 495, 545, 20.2, 72.92, 0.05, 1.82, 5.47, 2.3, 1.85, 4.87, 0.02, 2022/10/24, 2, 608

594, 483, 524, 18.54, 77.44, 0.04, 590, 495, 543, 19.72, 74.22, 0.05, 2.04, 5.61, 5.51, 1.37, 2.81, 0.01, 2022/10/24, 3, 610

594, 484, 542, 18.21, 79.06, 0.04, 590, 493, 541, 19.35, 75.11, 0.04, 2.35, 6.18, 7.45, 0.93, 1.76, 0.01, 2022/10/24, 4, 611

594, 484, 542, 17.83, 78.67, 0.05, 591, 492, 540, 19.08, 75.73, 0.04, 2.53, 6.07, 7.28, 0.72, 1.73, 0.01, 2022/10/24, 5, 610

595, 482, 542, 18.57, 80.54, 0.07, 591, 491, 540, 18.87, 76.0, 0.04, 2.39, 6.01, 7.21, 0.66, 1.92, 0.0, 2022/10/24, 6, 610

596, 479, 542, 19.29, 80.07, 0.07, 592, 489, 540, 18.77, 76.25, 0.04, 2.17, 5.68, 7.14, 0.61, 2.29, 0.01, 2022/10/24, 7, 610

597, 478, 543, 19.74, 78.83, 0.08, 593, 488, 540, 18.71, 76.68, 0.05, 2.06, 6.35, 7.06, 0.5, 2.49, 0.01, 2022/10/24, 8, 610

601, 479, 544, 22.6, 67.81, 0.11, 594, 486, 540, 18.85, 77.27, 0.05, 2.35, 7.2, 6.97, 0.79, 2.12, 0.02, 2022/10/24, 10, 611

608, 479, 550, 26.31, 54.08, 0.16, 595, 485, 540, 19.16, 76.59, 0.06, 2.79, 7.45, 6.97, 1.34, 3.46, 0.02, 2022/10/24, 11, 612

606, 481, 548, 24.22, 61.62, 0.12, 596, 484, 540, 19.8, 74.92, 0.07, 4.47, 6.77, 7.5, 2.44, 7.38, 0.04, 2022/10/24, 12, 612

607, 476, 547, 23.71, 64.06, 0.08, 597, 483, 541, 20.29, 73.71, 0.08, 5.05, 6.27, 7.76, 2.71, 8.29, 0.04, 2022/10/24, 13, 613

608, 479, 547, 23.0, 68.93, 0.08, 599, 481, 541, 20.71, 72.75, 0.08, 5.53, 4.32, 7.98, 2.82, 8.71, 0.04, 2022/10/24, 14, 614

610, 477, 547, 22.06, 71.14, 0.08, 600, 480, 543, 21.07, 72.17, 0.08, 5.84, 2.94, 6.55, 2.81, 8.72, 0.03, 2022/10/24, 15, 614

611, 471, 547, 21.27, 75.79, 0.07, 601, 479, 545, 21.36, 71.65, 0.09, 6.21, 2.87, 2.86, 2.7, 8.56, 0.03, 2022/10/24, 16, 614

610, 516, 545, 20.4, 77.97, 0.06, 603, 478, 545, 21.62, 71.38, 0.09, 6.36, 3.39, 2.79, 2.51, 8.35, 0.03, 2022/10/24, 17, 614

610, 521, 545, 20.05, 77.04, 0.04, 604, 481, 545, 21.83, 71.32, 0.09, 6.07, 11.39, 2.61, 2.26, 8.3, 0.03, 2022/10/24, 18, 615

610, 520, 544, 19.39, 78.91, 0.04, 605, 484, 546, 21.96, 71.03, 0.09, 5.59, 16.24, 2.39, 2.1, 8.0, 0.03, 2022/10/24, 19, 616

Anexo 13 - Amostra do *dataset* de testes da planta 1

Umidade1_V, Umidade2_V, Umidade3_V, Temperatura, UmidadeAr, indiceUV, Umidade1_V_MEDIA12H, Umidade2_V_MEDIA12H, Umidade3_V_MEDIA12H, Temperatura_MEDIA12H, UmidadeAr_MEDIA12H, indiceUV_MEDIA12H, Umidade1_V_DP12H, Umidade2_V_DP12H, Umidade3_V_DP12H, Temperatura_DP12H, UmidadeAr_DP12H, indiceUV_DP12H, Data, Hora, Umidade1_V_CLASSE

616, 477, 445, 19.76, 78.05, 0.03, 613, 482, 450, 20.61, 78.47, 0.04, 1.23, 9.63, 4.42, 1.27, 3.47, 0.02, 2022/11/13, 3, 626

617, 477, 446, 19.66, 82.71, 0.03, 614, 480, 448, 20.31, 79.09, 0.03, 1.44, 7.23, 3.45, 0.94, 2.44, 0.01, 2022/11/13, 4, 626

616, 475, 444, 19.43, 81.77, 0.04, 614, 478, 448, 20.07, 79.81, 0.03, 1.73, 3.75, 2.2, 0.63, 2.07, 0.01, 2022/11/13, 5, 626

616, 473, 443, 19.9, 77.01, 0.06, 614, 477, 447, 19.88, 80.33, 0.03, 1.8, 2.35, 1.6, 0.4, 1.65, 0.01, 2022/11/13, 6, 625

618, 476, 443, 21.5, 68.36, 0.07, 614, 477, 446, 19.8, 80.21, 0.03, 1.73, 2.35, 1.59, 0.26, 1.84, 0.01, 2022/11/13, 7, 625

632, 516, 457, 27.34, 51.21, 0.08, 654, 565, 473, 23.21, 79.77, 0.07, 12.81, 23.74, 7.62, 2.59, 13.66, 0.02, 2022/11/14, 15, 627

630, 512, 455, 27.09, 50.56, 0.07, 650, 561, 471, 23.8, 76.39, 0.08, 12.2, 27.69, 6.99, 2.68, 15.33, 0.02, 2022/11/14, 16, 628

630, 513, 455, 26.39, 52.53, 0.07, 646, 559, 468, 24.34, 73.05, 0.08, 10.72, 30.72, 7.48, 2.63, 16.28, 0.02, 2022/11/14, 17, 628

630, 510, 454, 25.37, 56.84, 0.05, 644, 555, 466, 24.79, 69.93, 0.08, 10.5, 33.45, 7.74, 2.45, 16.33, 0.02, 2022/11/14, 18, 628

628, 501, 453, 24.52, 62.48, 0.04, 642, 549, 465, 25.13, 67.25, 0.08, 10.31, 34.87, 7.89, 2.2, 15.53, 0.02, 2022/11/14, 19, 629

628, 500, 455, 24.03, 65.51, 0.04, 640, 543, 463, 25.32, 65.34, 0.08, 10.53, 35.89, 7.96, 2.02, 14.48, 0.02, 2022/11/14, 20, 630

628, 500, 455, 23.67, 67.33, 0.04, 638, 535, 462, 25.37, 64.41, 0.07, 10.98, 33.71, 8.09, 1.97, 14.03, 0.02, 2022/11/14, 21, 631

629, 502, 453, 23.27, 67.8, 0.04, 637, 529, 460, 25.48, 62.82, 0.07, 11.37, 33.43, 7.75, 1.81, 12.28, 0.03, 2022/11/14, 22, 632

629, 502, 453, 22.95, 70.07, 0.03, 635, 521, 459, 25.57, 61.12, 0.07, 9.53, 26.16, 7.26, 1.66, 9.58, 0.03, 2022/11/14, 23, 635

629, 502, 453, 22.98, 70.91, 0.04, 633, 515, 457, 25.5, 60.16, 0.06, 6.29, 19.5, 5.93, 1.75, 7.72, 0.03, 2022/11/15, 0, 634

628, 498, 449, 22.93, 70.8, 0.04, 631, 509, 456, 25.22, 60.39, 0.06, 4.4, 10.77, 3.82, 1.87, 8.02, 0.02, 2022/11/15, 1, 634

628, 497, 448, 22.48, 72.27, 0.04, 630, 506, 454, 24.86, 61.31, 0.05, 2.81, 7.25, 2.64, 1.86, 8.56, 0.02, 2022/11/15, 2, 634

Anexo 14 - Amostra do *dataset* de treino da planta 2

Umidade1_V,Umidade2_V,Umidade3_V,Temperatura,UmidadeAr,indiceUV,Umidade1_V_MEDIA12H,Umidade2_V_MEDIA12H,Umidade3_V_MEDIA12H,Temperatura_MEDIA12H,UmidadeAr_MEDIA12H,indiceUV_MEDIA12H,Umidade1_V_DP12H,Umidade2_V_DP12H,Umidade3_V_DP12H,Temperatura_DP12H,UmidadeAr_DP12H,indiceUV_DP12H,Data,Hora,Umidade2_V_CLASSE

593,491,526,18.61,75.86,0.04,590,495,545,20.2,72.92,0.05,1.82,5.47,2.3,1.85,4.87,0.02,2022/10/24,2,479

594,483,524,18.54,77.44,0.04,590,495,543,19.72,74.22,0.05,2.04,5.61,5.51,1.37,2.81,0.01,2022/10/24,3,477

594,484,542,18.21,79.06,0.04,590,493,541,19.35,75.11,0.04,2.35,6.18,7.45,0.93,1.76,0.01,2022/10/24,4,471

594,484,542,17.83,78.67,0.05,591,492,540,19.08,75.73,0.04,2.53,6.07,7.28,0.72,1.73,0.01,2022/10/24,5,516

595,482,542,18.57,80.54,0.07,591,491,540,18.87,76.0,0.04,2.39,6.01,7.21,0.66,1.92,0.0,2022/10/24,6,521

596,479,542,19.29,80.07,0.07,592,489,540,18.77,76.25,0.04,2.17,5.68,7.14,0.61,2.29,0.01,2022/10/24,7,520

597,478,543,19.74,78.83,0.08,593,488,540,18.71,76.68,0.05,2.06,6.35,7.06,0.5,2.49,0.01,2022/10/24,8,517

601,479,544,22.6,67.81,0.11,594,486,540,18.85,77.27,0.05,2.35,7.2,6.97,0.79,2.12,0.02,2022/10/24,10,517

608,479,550,26.31,54.08,0.16,595,485,540,19.16,76.59,0.06,2.79,7.45,6.97,1.34,3.46,0.02,2022/10/24,11,517

606,481,548,24.22,61.62,0.12,596,484,540,19.8,74.92,0.07,4.47,6.77,7.5,2.44,7.38,0.04,2022/10/24,12,516

607,476,547,23.71,64.06,0.08,597,483,541,20.29,73.71,0.08,5.05,6.27,7.76,2.71,8.29,0.04,2022/10/24,13,517

608,479,547,23.0,68.93,0.08,599,481,541,20.71,72.75,0.08,5.53,4.32,7.98,2.82,8.71,0.04,2022/10/24,14,517

610,477,547,22.06,71.14,0.08,600,480,543,21.07,72.17,0.08,5.84,2.94,6.55,2.81,8.72,0.03,2022/10/24,15,517

611,471,547,21.27,75.79,0.07,601,479,545,21.36,71.65,0.09,6.21,2.87,2.86,2.7,8.56,0.03,2022/10/24,16,515

610,516,545,20.4,77.97,0.06,603,478,545,21.62,71.38,0.09,6.36,3.39,2.79,2.51,8.35,0.03,2022/10/24,17,514

610,521,545,20.05,77.04,0.04,604,481,545,21.83,71.32,0.09,6.07,11.39,2.61,2.26,8.3,0.03,2022/10/24,18,515

610,520,544,19.39,78.91,0.04,605,484,546,21.96,71.03,0.09,5.59,16.24,2.39,2.1,8.0,0.03,2022/10/24,19,514

Anexo 15 - Amostra do *dataset* de testes da planta 2

Umidade1_V,Umidade2_V,Umidade3_V,Temperatura,UmidadeAr,indiceUV,Umidade1_V_MEDIA12H,Umidade2_V_MEDIA12H,Umidade3_V_MEDIA12H,Temperatura_MEDIA12H,UmidadeAr_MEDIA12H,indiceUV_MEDIA12H,Umidade1_V_DP12H,Umidade2_V_DP12H,Umidade3_V_DP12H,Temperatura_DP12H,UmidadeAr_DP12H,indiceUV_DP12H,Data,Hora,Umidade2_V_CLASSE

616,477,445,19.76,78.05,0.03,613,482,450,20.61,78.47,0.04,1.23,9.63,4.42,1.27,3.47,0.02,2022/11/13,3,484

617,477,446,19.66,82.71,0.03,614,480,448,20.31,79.09,0.03,1.44,7.23,3.45,0.94,2.44,0.01,2022/11/13,4,483

616,475,444,19.43,81.77,0.04,614,478,448,20.07,79.81,0.03,1.73,3.75,2.2,0.63,2.07,0.01,2022/11/13,5,482

616,473,443,19.9,77.01,0.06,614,477,447,19.88,80.33,0.03,1.8,2.35,1.6,0.4,1.65,0.01,2022/11/13,6,482

618,476,443,21.5,68.36,0.07,614,477,446,19.8,80.21,0.03,1.73,2.35,1.59,0.26,1.84,0.01,2022/11/13,7,482

632,516,457,27.34,51.21,0.08,654,565,473,23.21,79.77,0.07,12.81,23.74,7.62,2.59,13.66,0.02,2022/11/14,15,495

630,512,455,27.09,50.56,0.07,650,561,471,23.8,76.39,0.08,12.2,27.69,6.99,2.68,15.33,0.02,2022/11/14,16,494

630,513,455,26.39,52.53,0.07,646,559,468,24.34,73.05,0.08,10.72,30.72,7.48,2.63,16.28,0.02,2022/11/14,17,494

630,510,454,25.37,56.84,0.05,644,555,466,24.79,69.93,0.08,10.5,33.45,7.74,2.45,16.33,0.02,2022/11/14,18,494

628,501,453,24.52,62.48,0.04,642,549,465,25.13,67.25,0.08,10.31,34.87,7.89,2.2,15.53,0.02,2022/11/14,19,493

628,500,455,24.03,65.51,0.04,640,543,463,25.32,65.34,0.08,10.53,35.89,7.96,2.02,14.48,0.02,2022/11/14,20,494

628,500,455,23.67,67.33,0.04,638,535,462,25.37,64.41,0.07,10.98,33.71,8.09,1.97,14.03,0.02,2022/11/14,21,491

629,502,453,23.27,67.8,0.04,637,529,460,25.48,62.82,0.07,11.37,33.43,7.75,1.81,12.28,0.03,2022/11/14,22,492

629,502,453,22.95,70.07,0.03,635,521,459,25.57,61.12,0.07,9.53,26.16,7.26,1.66,9.58,0.03,2022/11/14,23,495

629,502,453,22.98,70.91,0.04,633,515,457,25.5,60.16,0.06,6.29,19.5,5.93,1.75,7.72,0.03,2022/11/15,0,493

628,498,449,22.93,70.8,0.04,631,509,456,25.22,60.39,0.06,4.4,10.77,3.82,1.87,8.02,0.02,2022/11/15,1,499

Anexo 16 - Amostra do *dataset* de treino da planta 3

Umidade1_V, Umidade2_V, Umidade3_V, Temperatura, UmidadeAr, indiceUV, Umidade1_V_MEDIA12H, Umidade2_V_MEDIA12H, Umidade3_V_MEDIA12H, Temperatura_MEDIA12H, UmidadeAr_MEDIA12H, indiceUV_MEDIA12H, Umidade1_V_DP12H, Umidade2_V_DP12H, Umidade3_V_DP12H, Temperatura_DP12H, UmidadeAr_DP12H, indiceUV_DP12H, Data, Hora, Umidade3_V_CLASSE

593, 491, 526, 18.61, 75.86, 0.04, 590, 495, 545, 20.2, 72.92, 0.05, 1.82, 5.47, 2.3, 1.85, 4.87, 0.02, 2022/10/24, 2, 547

594, 483, 524, 18.54, 77.44, 0.04, 590, 495, 543, 19.72, 74.22, 0.05, 2.04, 5.61, 5.51, 1.37, 2.81, 0.01, 2022/10/24, 3, 547

594, 484, 542, 18.21, 79.06, 0.04, 590, 493, 541, 19.35, 75.11, 0.04, 2.35, 6.18, 7.45, 0.93, 1.76, 0.01, 2022/10/24, 4, 547

594, 484, 542, 17.83, 78.67, 0.05, 591, 492, 540, 19.08, 75.73, 0.04, 2.53, 6.07, 7.28, 0.72, 1.73, 0.01, 2022/10/24, 5, 545

595, 482, 542, 18.57, 80.54, 0.07, 591, 491, 540, 18.87, 76.0, 0.04, 2.39, 6.01, 7.21, 0.66, 1.92, 0.0, 2022/10/24, 6, 545

596, 479, 542, 19.29, 80.07, 0.07, 592, 489, 540, 18.77, 76.25, 0.04, 2.17, 5.68, 7.14, 0.61, 2.29, 0.01, 2022/10/24, 7, 544

597, 478, 543, 19.74, 78.83, 0.08, 593, 488, 540, 18.71, 76.68, 0.05, 2.06, 6.35, 7.06, 0.5, 2.49, 0.01, 2022/10/24, 8, 543

601, 479, 544, 22.6, 67.81, 0.11, 594, 486, 540, 18.85, 77.27, 0.05, 2.35, 7.2, 6.97, 0.79, 2.12, 0.02, 2022/10/24, 10, 545

608, 479, 550, 26.31, 54.08, 0.16, 595, 485, 540, 19.16, 76.59, 0.06, 2.79, 7.45, 6.97, 1.34, 3.46, 0.02, 2022/10/24, 11, 544

606, 481, 548, 24.22, 61.62, 0.12, 596, 484, 540, 19.8, 74.92, 0.07, 4.47, 6.77, 7.5, 2.44, 7.38, 0.04, 2022/10/24, 12, 544

607, 476, 547, 23.71, 64.06, 0.08, 597, 483, 541, 20.29, 73.71, 0.08, 5.05, 6.27, 7.76, 2.71, 8.29, 0.04, 2022/10/24, 13, 545

608, 479, 547, 23.0, 68.93, 0.08, 599, 481, 541, 20.71, 72.75, 0.08, 5.53, 4.32, 7.98, 2.82, 8.71, 0.04, 2022/10/24, 14, 545

610, 477, 547, 22.06, 71.14, 0.08, 600, 480, 543, 21.07, 72.17, 0.08, 5.84, 2.94, 6.55, 2.81, 8.72, 0.03, 2022/10/24, 15, 546

611, 471, 547, 21.27, 75.79, 0.07, 601, 479, 545, 21.36, 71.65, 0.09, 6.21, 2.87, 2.86, 2.7, 8.56, 0.03, 2022/10/24, 16, 544

610, 516, 545, 20.4, 77.97, 0.06, 603, 478, 545, 21.62, 71.38, 0.09, 6.36, 3.39, 2.79, 2.51, 8.35, 0.03, 2022/10/24, 17, 544

Anexo 17 - Amostra do *dataset* de teste da planta 3

Umidade1_V, Umidade2_V, Umidade3_V, Temperatura, UmidadeAr, indiceUV, Umidade1_V_MEDIA12H, Umidade2_V_MEDIA12H, Umidade3_V_MEDIA12H, Temperatura_MEDIA12H, UmidadeAr_MEDIA12H, indiceUV_MEDIA12H, Umidade1_V_DP12H, Umidade2_V_DP12H, Umidade3_V_DP12H, Temperatura_DP12H, UmidadeAr_DP12H, indiceUV_DP12H, Data, Hora, Umidade3_V_CLASSE

616, 477, 445, 19.76, 78.05, 0.03, 613, 482, 450, 20.61, 78.47, 0.04, 1.23, 9.63, 4.42, 1.27, 3.47, 0.02, 2022/11/13, 3, 448

617, 477, 446, 19.66, 82.71, 0.03, 614, 480, 448, 20.31, 79.09, 0.03, 1.44, 7.23, 3.45, 0.94, 2.44, 0.01, 2022/11/13, 4, 448

616, 475, 444, 19.43, 81.77, 0.04, 614, 478, 448, 20.07, 79.81, 0.03, 1.73, 3.75, 2.2, 0.63, 2.07, 0.01, 2022/11/13, 5, 447

616, 473, 443, 19.9, 77.01, 0.06, 614, 477, 447, 19.88, 80.33, 0.03, 1.8, 2.35, 1.6, 0.4, 1.65, 0.01, 2022/11/13, 6, 446

618, 476, 443, 21.5, 68.36, 0.07, 614, 477, 446, 19.8, 80.21, 0.03, 1.73, 2.35, 1.59, 0.26, 1.84, 0.01, 2022/11/13, 7, 446

632, 516, 457, 27.34, 51.21, 0.08, 654, 565, 473, 23.21, 79.77, 0.07, 12.81, 23.74, 7.62, 2.59, 13.66, 0.02, 2022/11/14, 15, 447

630, 512, 455, 27.09, 50.56, 0.07, 650, 561, 471, 23.8, 76.39, 0.08, 12.2, 27.69, 6.99, 2.68, 15.33, 0.02, 2022/11/14, 16, 447

630, 513, 455, 26.39, 52.53, 0.07, 646, 559, 468, 24.34, 73.05, 0.08, 10.72, 30.72, 7.48, 2.63, 16.28, 0.02, 2022/11/14, 17, 447

630, 510, 454, 25.37, 56.84, 0.05, 644, 555, 466, 24.79, 69.93, 0.08, 10.5, 33.45, 7.74, 2.45, 16.33, 0.02, 2022/11/14, 18, 446

628, 501, 453, 24.52, 62.48, 0.04, 642, 549, 465, 25.13, 67.25, 0.08, 10.31, 34.87, 7.89, 2.2, 15.53, 0.02, 2022/11/14, 19, 446

628, 500, 455, 24.03, 65.51, 0.04, 640, 543, 463, 25.32, 65.34, 0.08, 10.53, 35.89, 7.96, 2.02, 14.48, 0.02, 2022/11/14, 20, 447

628, 500, 455, 23.67, 67.33, 0.04, 638, 535, 462, 25.37, 64.41, 0.07, 10.98, 33.71, 8.09, 1.97, 14.03, 0.02, 2022/11/14, 21, 446

629, 502, 453, 23.27, 67.8, 0.04, 637, 529, 460, 25.48, 62.82, 0.07, 11.37, 33.43, 7.75, 1.81, 12.28, 0.03, 2022/11/14, 22, 447

629, 502, 453, 22.95, 70.07, 0.03, 635, 521, 459, 25.57, 61.12, 0.07, 9.53, 26.16, 7.26, 1.66, 9.58, 0.03, 2022/11/14, 23, 450

629, 502, 453, 22.98, 70.91, 0.04, 633, 515, 457, 25.5, 60.16, 0.06, 6.29, 19.5, 5.93, 1.75, 7.72, 0.03, 2022/11/15, 0, 448

628, 498, 449, 22.93, 70.8, 0.04, 631, 509, 456, 25.22, 60.39, 0.06, 4.4, 10.77, 3.82, 1.87, 8.02, 0.02, 2022/11/15, 1, 449

628, 497, 448, 22.48, 72.27, 0.04, 630, 506, 454, 24.86, 61.31, 0.05, 2.81, 7.25, 2.64, 1.86, 8.56, 0.02, 2022/11/15, 2, 450