

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Raphael Azambuja Silva Macedo

**Protótipo para Interface Web de  
Gerenciamento de Dados Meteorológicos**

**Uberlândia, Brasil**

**2025**

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Raphael Azambuja Silva Macedo

**Protótipo para Interface Web de Gerenciamento de  
Dados Meteorológicos**

Trabalho de conclusão de curso apresentado  
à Faculdade de Computação da Universidade  
Federal de Uberlândia, como parte dos requi-  
sitos exigidos para a obtenção título de Ba-  
charel em Ciência da Computação.

Orientador: Mauricio Cunha Escarpinati

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2025

Raphael Azambuja Silva Macedo

## **Protótipo para Interface Web de Gerenciamento de Dados Meteorológicos**

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Uberlândia, Brasil, 09 de maio de 2025:

---

**Mauricio Cunha Escarpinati**  
Orientador

---

**Professor**

---

**Professor**

Uberlândia, Brasil  
2025

*“My compass is curiosity.”*

- Mili

# Resumo

Este trabalho aborda o desafio no armazenamento de dados meteorológicos enfrentados pelo grupo de meteorologia da UFU, devido à limitação do atual sistema de banco de dados, que envolve inserção manual de dados em planilhas do Excel. Diante da necessidade de eficiência e segurança, propõe-se migrar para um banco de dados relacional SQL com interface web, já que isso facilitaria análises, comparações e iria oferecer uma solução mais adaptável às necessidades em evolução do grupo.

**Palavras-chave:** SQL, Interface Web, Banco de Dados, Meteorologia, CLIMA/UFU.

# Lista de ilustrações

Figura 1 – Sensores meteorológicos do CLIMA/UFU. Fonte:< <a href="https://www.iciag.ufu.br/unidades/laboratorio/clima-laboratorio-de-climatologia-e-meteorologia-ambiental">https://www.iciag.ufu.br/unidades/laboratorio/clima-laboratorio-de-climatologia-e-meteorologia-ambiental</a>	
Figura 2 – CSV recebido transformado em uma tabela para melhor entendimento. Fonte: autor (2025).	21
Figura 3 – Tabelas que representam o modelo do banco de dados. Fonte: autor (2025).	23
Figura 4 – Referente ao Teste 1: Numero de leituras, assim como o tempo gasto, os erros encontrados são os dados iguais tentando serem inseridos novamente. Fonte: autor (2025).	27
Figura 5 – Referente ao Teste 2: Numero de leituras, assim como o tempo gasto, os erros encontrados são os dados iguais tentando serem inseridos novamente. Fonte: autor (2025).	27
Figura 6 – Referente ao Teste 3: Numero de leituras, assim como o tempo gasto, os erros encontrados são os dados iguais tentando serem inseridos novamente. Fonte: autor (2025).	28

# Lista de tabelas

Tabela 1 – Configuração do Ambiente de Teste . . . . .	26
Tabela 2 – Comparação de desempenho entre diferentes configurações . . . . .	28

# Lista de abreviaturas e siglas

UFU	Universidade Federal de Uberlândia
SQL	Structured query language
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
CSV	Comma-separated values
CLIMA	Laboratório de Climatologia e Meteorologia Ambiental



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
<b>1.1</b>	<b>Laboratório de Climatologia e Meteorologia Ambiental</b>	<b>11</b>
1.1.1	Infraestrutura e Equipamentos	11
1.1.2	Atividades Educacionais e de Extensão	11
1.1.3	Colaboração com o CLIMA	12
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>14</b>
<b>2.1</b>	<b>Banco de dados Relacional</b>	<b>14</b>
<b>2.2</b>	<b>Interface Web</b>	<b>15</b>
<b>2.3</b>	<b>Desenvolvimento Web</b>	<b>15</b>
2.3.1	Arquitetura e Tecnologias Fundamentais	15
2.3.2	Práticas e Ferramentas Essenciais	16
2.3.3	Relevância para o Projeto	16
<b>2.4</b>	<b>Html, CSS, Javascript e Node</b>	<b>16</b>
2.4.1	HTML	16
2.4.2	CSS	17
2.4.3	JavaScript	17
2.4.4	Node	17
<b>2.5</b>	<b>SQL</b>	<b>17</b>
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>19</b>
<b>4</b>	<b>MÉTODOLOGIA</b>	<b>20</b>
<b>4.1</b>	<b>Visão Geral</b>	<b>20</b>
<b>4.2</b>	<b>Formato dos dados recebidos</b>	<b>20</b>
<b>4.3</b>	<b>Modelo do Banco de Dados</b>	<b>22</b>
<b>4.4</b>	<b>Captura dos dados e tratamento de colisões</b>	<b>23</b>
<b>4.5</b>	<b>Armazenamento</b>	<b>24</b>
<b>5</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>26</b>
<b>5.1</b>	<b>Análise Experimental do Tempo de Processamento de Dados em Planilha</b>	<b>28</b>
5.1.1	Perfil dos Voluntários	28
5.1.2	Resultados Experimentais	29
5.1.3	Considerações sobre a Escalabilidade da Tarefa	29
<b>6</b>	<b>CONCLUSÃO</b>	<b>30</b>

6.1      Trabalhos futuros . . . . . 30

REFERÊNCIAS . . . . . 31

# 1 Introdução

A gestão de dados meteorológicos no grupo de meteorologia da UFU enfrenta desafios e problemas devido à simplicidade do Excel, além do gasto de tempo excessivo em suas inserções. Este trabalho surge da necessidade de uma solução mais eficiente e precisa para gerenciar os dados meteorológicos do grupo, atualmente inseridos manualmente em planilhas do Excel. Este processo, além de ineficiente, está sujeito a erros humanos.

A justificativa para esta pesquisa é clara: o método atual de armazenamento e gerenciamento de dados meteorológicos é insuficiente para permitir um acompanhamento histórico adequado, além de inseguro. A falta de um sistema estruturado dificulta análises e comparações, logo, a proposta é migrar para um banco de dados SQL com uma interface web, tornando assim, o acesso a esses dados mais global, ou seja, o pesquisador poderá acessar a interface web de qualquer lugar, além disso, não precisará ter uma cópia da planilha Excel em cada computador dos pesquisadores.

A capacidade de acompanhar o histórico de dados facilitará não apenas a pesquisa atual, mas também futuras análises comparativas, além de ser bem mais seguro quando a possibilidade da perda de um arquivo Excel, em comparação a um banco de dados inteiro, que estará localizado em um servidor, ou seja, mais difícil de apagar ou alterar os dados sem que a intenção fosse essa.

Outros trabalhos na área discutiram a criação de bancos de dados e aplicativos, tanto web quanto móveis. ([SILVA, 2021](#)) propôs tanto um aplicativo quanto uma interface web, enquanto ([PEREIRA, 2022b](#)) recomenda a criação de uma interface web, destacando a preferência por essa abordagem em comparação com aplicativos locais ou móveis.

A proposta principal deste trabalho é desenvolver uma interface web automatizada capaz de receber dados no formato .txt (formato texto), inserindo-os automaticamente no banco de dados. O desafio está na escolha de um banco de dados que possa acomodar as diversas variáveis meteorológicas, já que o grupo meteorológico pode conseguir equipamentos novos, e com medidas diferentes no futuro, além de novos sensores, ou seja, o banco deve acomodar a inserção de novas variáveis.

O projeto será implementado como um aplicativo web, utilizando SQL como linguagem de banco de dados. O software Workbench será empregado para a gestão do banco de dados, enquanto CSS, HTML e Javascript são utilizados para fazer o front-end da interface web, e Node no backend.

## 1.1 Laboratório de Climatologia e Meteorologia Ambiental

O Laboratório de Climatologia e Meteorologia Ambiental é uma unidade vinculada ao Instituto de Ciências Agrárias da Universidade Federal de Uberlândia, localizada no Campus Glória. Sua missão é promover atividades de ensino, pesquisa e extensão nas áreas de climatologia e meteorologia ambiental, com ênfase na coleta, análise e interpretação de dados meteorológicos para aplicações acadêmicas e práticas, é para este laboratório que a interface web será desenvolvida.

### 1.1.1 Infraestrutura e Equipamentos

O CLIMA conta com três estações meteorológicas equipadas com diversos sensores que permitem o monitoramento contínuo de variáveis atmosféricas, alguns deles são:

- **Termômetro:** mede a temperatura do ar.
- **Barômetro:** registra a pressão atmosférica.
- **Higrômetro:** avalia a umidade relativa do ar.
- **Anemômetro:** determina a velocidade do vento.
- **Biruta:** indica a direção do vento.
- **Pluviômetro:** quantifica a precipitação pluviométrica.
- **Piranômetro:** mede a radiação solar incidente.
- **Heliógrafo:** registra a duração da insolação diária.

Esses equipamentos possibilitam a obtenção de dados precisos, porém não em tempo real. É necessário que os pesquisadores vão até as estações retirar os arquivos de dado, que estão em formato CSV, e então, colocar em planilhas excel.

### 1.1.2 Atividades Educacionais e de Extensão

O Laboratório de Climatologia e Meteorologia Ambiental da Universidade Federal de Uberlândia é muito importante na promoção da educação ambiental e na disseminação do conhecimento climatológico. Por meio de visitas monitoradas, o laboratório proporciona experiências práticas enriquecedoras para estudantes de diversos níveis de ensino.

Durante essas visitas, os alunos têm a oportunidade de interagir diretamente com os instrumentos meteorológicos utilizados na estação do CLIMA. Eles aprendem sobre o funcionamento de equipamentos como termômetros, barômetros, higrômetros, anemômetros, birutas, pluviômetros, piranômetros e heliógrafos.

Além disso, as visitas são acompanhadas por membros da equipe do laboratório, que conduzem discussões sobre temas relacionados ao clima, sua medição e influência no cotidiano das pessoas.

O CLIMA também atua como um centro de referência para a comunidade acadêmica e externa, disponibilizando os dados coletados para pesquisas e projetos que visam compreender e mitigar os impactos das mudanças climáticas.

### 1.1.3 Colaboração com o CLIMA

Este trabalho foi desenvolvido com o objetivo de apoiar o CLIMA. Atualmente, o CLIMA realiza a coleta e análise de dados meteorológicos por meio de estações instaladas nas Fazendas Experimentais da UFU, como Água Limpa, Capim Branco e Glória.

No entanto, a gestão desses dados enfrenta desafios significativos. A utilização de planilhas de excel para o armazenamento das informações pode levar a inconsistências, perdas de dados e dificuldades na recuperação de históricos. Esses problemas são comuns em sistemas que dependem de registros manuais ou semiautomáticos, especialmente quando não há uma padronização nos processos de entrada e manutenção dos dados.

Reconhecendo a importância do CLIMA para a comunidade acadêmica e para a sociedade em geral, este projeto propõe a implementação de uma solução tecnológica que automatize a coleta, o armazenamento e a visualização dos dados meteorológicos. Através de uma interface web intuitiva e segura.

Além disso, a sistematização dos dados permitirá a geração de relatórios e gráficos que auxiliem na compreensão dos padrões climáticos da região. A colaboração entre este projeto e o CLIMA visa, portanto, fortalecer a infraestrutura de dados meteorológicos da UFU, promovendo a excelência na pesquisa e no ensino.



Figura 1 – Sensores meteorológicos do CLIMA/UFU. Fonte: <<https://www.iciag.ufu.br/unidades/laboratorio/clima-laboratorio-de-climatologia-e-meteorologia-ambiental>>

## 2 Fundamentação Teórica

### 2.1 Banco de dados Relacional

Um banco de dados relacional organiza dados em tabelas, cada uma composta por linhas e colunas. Essas tabelas são interligadas por meio de chaves primárias e secundárias, que servem para fazer conexões entre os elementos de tais tabelas.

O conceito de banco de dados relacional foi pioneiramente desenvolvido por ([Codd, 1990](#)) na década de 1970, e revolucionou a forma como os dados eram manipulados. A modelagem relacional proporcionou uma abordagem mais organizada e eficiente para armazenar informações, ou seja, era mais fácil mostrar para o cliente como o banco funcionaria na prática, além de ser bem mais eficiente e rápido na busca e inserção de informações.

Ao longo dos anos, a evolução tecnológica trouxe uma variedade de sistemas de bancos de dados, cada um adaptado para atender a diferentes necessidades. Esses sistemas não apenas expandiram a escalabilidade, mas também aprimoraram a usabilidade dessas plataformas. A introdução de práticas de DevOps e a integração de tecnologias de nuvem têm desempenhado um papel significativo na otimização da gestão e acessibilidade aos dados, agora sendo possível ter um banco de dados fora da sua máquina local.

Projetos contemporâneos, como o mencionado trabalho de ([Pereira, 2022b](#)), destacam a aplicação prática de bancos de dados relacionais. O uso específico do PostgreSQL nesse projeto evidencia a escolha como ele é uma tecnologia forte e confiável.

## 2.2 Interface Web

A interface web é o que vai gerir a interação do usuário com um sistema, ela é acessível por meio de um navegador da web, como o Firefox, ou Chrome, isso torna seu acesso extremamente difundido, ao invés de ter um computador potente para gerir todas as informações e operações de inserções e busca, isso é feito no servidor, ou seja, qualquer computador com acesso a internet e um navegador consegue utilizar de uma interface web.

O surgimento da interface web está ligado ao desenvolvimento da internet, ela foi criada pelo advento de tecnologias como HTML, CSS e JavaScript. Essas tecnologias democratizaram a informação, já que antigamente a internet funcionava apenas pelo envio de textos por um simples protocolo TCP, o HTML surgiu primeiro, em 1991 criado por Tim Berners-Lee ([Giovani, Marllon Bianchini, 2012](#)).

Nos dias atuais, as interfaces web evoluíram significativamente, graças as novas tecnologias feitas para o frontend. Ferramentas como React e Angular são amplamente utilizadas, elas conseguem facilitar bastante o processo de criação de sites. Projetos recentes, como o apresentado por ([PAVANELI, 2023](#)), demonstram o uso de frameworks, como TypeScript, para desenvolver interfaces web. Da mesma forma, o projeto de ([OLIVEIRA, 2022](#)) utiliza o React, outro framework bem comum.

## 2.3 Desenvolvimento Web

O desenvolvimento web é uma área da engenharia de software dedicada à criação de aplicações acessíveis por meio de navegadores, abrangendo desde simples páginas estáticas até sistemas complexos e dinâmicos. Essa disciplina envolve a integração de diversas tecnologias e práticas que permitem a construção de soluções escaláveis, seguras e eficientes.

### 2.3.1 Arquitetura e Tecnologias Fundamentais

O desenvolvimento web é tradicionalmente dividido em duas grandes áreas:

- **Front-end:** Refere-se à interface com a qual o usuário interage diretamente. Utiliza tecnologias como HTML para estruturação do conteúdo, CSS para estilização e JavaScript para interatividade. Frameworks modernos como React, Angular e Vue.js facilitam a criação de interfaces dinâmicas e responsivas ([CARVALHO, 2023](#)).
- **Back-end:** Responsável pelo processamento de dados, lógica de negócio e comunicação com bancos de dados. Linguagens como Node.js, Python, PHP e Java são comumente utilizadas ([PEREIRA, 2022a](#)).



### 2.3.2 Práticas e Ferramentas Essenciais

Além das tecnologias básicas, o desenvolvimento web moderno incorpora diversas práticas e ferramentas que aprimoram o processo de desenvolvimento:

- **Controle de Versão:** Ferramentas como Git permitem o gerenciamento eficiente de alterações no código, além de um gerenciamento de versionamento. No caso deste manuscrito, utilizou-se o GitHub para esse controle, cujo repositório pode ser acessado pelo link: <https://github.com/mekanss/TCC/tree/master>.
- **Segurança:** A implementação de práticas de segurança, como autenticação, autorização e proteção contra vulnerabilidades comuns (por exemplo, XSS, CSRF, injeção de SQL), é necessária para manter a interface web segura. Embora este trabalho seja um protótipo e não esteja disponível publicamente, é fundamental considerar essas práticas em ambientes de produção (GUIMARÃES, 2023; BARBOSA; FERREIRA; NEVES, 2023).

### 2.3.3 Relevância para o Projeto

No contexto deste trabalho, o desenvolvimento web é central para a criação de uma interface que permita aos usuários interagir com o sistema de forma rápida, eficiente e sem dificuldades. A escolha por tecnologia web proporciona:

- **Acessibilidade:** Usuários podem acessar o sistema de qualquer dispositivo com navegador e conexão à internet, sem a necessidade de instalar software adicional.
- **Escalabilidade:** A arquitetura web facilita a expansão do sistema para atender a um número crescente de usuários e funcionalidades.
- **Manutenibilidade:** A separação entre front-end e back-end permite atualizações e manutenções mais organizadas e menos propensas a erros.

## 2.4 Html, CSS, Javascript e Node

### 2.4.1 HTML

O HTML é a linguagem de marcação padrão para a criação de páginas web. Ele define a estrutura e o conteúdo de um documento, utilizando elementos como títulos, parágrafos, links e imagens. Desde sua primeira versão, o HTML evoluiu significativamente, culminando no HTML5, que consegue pegar tags, e simplificar mais ainda o desenvolvimento web, com ele também é possível trabalhar com áudio e vídeo (Giovani, Marllon Bianchini, 2012).

### 2.4.2 CSS

O CSS é a linguagem responsável pela apresentação visual das páginas web. Ele permite a estilização de elementos HTML, definindo cores, fontes, layouts e outros aspectos estéticos, no caso, ele consegue utilizar as tags e classes criadas no HTML para deixar a página com um aspecto visual melhor. Porém ele não é necessário para o funcionamento em si da página.

Existe ainda o CSS3, que traz muitas novas funcionalidades como animações pré montadas e várias funções prontas que facilitam bastante chegar no design previsto pelo artista ou designer, porém como o programa será utilizado por fins mais próximos a pesquisa, decidi não utilizar o CSS3, e sim o simples CSS ([FRANÇA, 2015](#)).

### 2.4.3 JavaScript

JavaScript é uma linguagem de programação interpretada que permite a criação de páginas web dinâmicas e interativas, graças a ela é possível adicionar e remover elementos da interface, além de enviar formulários, realizar requisições assíncronas e diversas outras funcionalidades ([LOVATTO, 2022](#)).

Originalmente concebida para ser executada no lado do cliente (navegador), sua funcionalidade foi expandida para o lado do servidor com o surgimento do Node.js, que também é utilizado nesse manuscrito.

### 2.4.4 Node

Node.js é um ambiente de execução JavaScript que permite a utilização dessa linguagem no lado do servidor. Com ele, é possível desenvolver servidores e serviços web que, tradicionalmente, eram implementados em outras linguagens como Java, PHP ou Python. Node.js opera de maneira assíncrona e orientada a eventos, o que o torna ideal para uma escalabilidade, caso necessário ([SOUZA, 2016](#)).

O ambiente foi desenvolvido sobre o motor V8 do Google Chrome, o que permite a execução de código JavaScript fora do navegador. A escolha dessa linguagem, ao invés de PHP e outras linguagens focada no backend, foi pela familiaridade do autor com ela, além disso ao unificar tanto o backend com o frontend com Javascript é possível tornar a leitura do código mais leve.

## 2.5 SQL

SQL é uma linguagem de consulta estruturada amplamente utilizada para gerenciar e manipular bancos de dados relacionais. Ela permite a definição, consulta, atualiza-

ção e controle de acesso aos dados de forma padronizada e com uma grande facilidade de modelagem. Com comandos como SELECT, INSERT, UPDATE e DELETE.

Os usuários e desenvolvedores podem inserir, atualizar, pesquisar e até deletar dados, sobre a proteção dos constraints — restrições que evitam a inserção de dados inconsistentes. Essas restrições, como primary keys, foreign keys, check constraints, e unique constraints, diminuem bastante erros de usuários, já que se torna quase impossível apagar dados importantes, ou inserir dados em colunas erradas.

Isso é uma outra grande vantagem em comparação aos bancos de dados simples, como Excel, que não possuem mecanismos robustos para assegurar a precisão e a consistência dos dados em larga escala.

### 3 Trabalhos Relacionados

Um artigo científico que tem como objetivo algo similar a esta monografia é o de ([Cassim, H. T. Sanches, N, 2022](#)), nele, um grupo de pesquisadores fazem tanto um banco de dados meteorológico quanto um sensor utilizando Arduino, que é uma placa barata e altamente customizável, porém é notável o quão simples o banco de dados que eles vão utilizar é, ele contém apenas uma tabela com variáveis já definidas, em contrapartida, no caso desse manuscrito, um dos maiores desafios é tentar adaptar o banco de dados a possibilidades de novas variáveis e sensores.

Já o de ([PEREIRA, 2022b](#)) tem metodologias e práticas mais similares a dessa monografia, por exemplo, o autor utiliza da linguagem Java, além do SQL, que é uma das tecnologias que também será utilizadas nesse projeto. Porém, o autor fez um aplicativo local, ao invés de uma aplicação web. Também existe o fato de que ele utilizou o Swing, que é um pacote já inerente da linguagem Java, feito para exibir interfaces todavia, nesse manuscrito será utilizado apenas o CSS, não tendo necessidade de algo tão complexo para necessitar um framework.

No caso de ([SILVA, 2021](#)), o autor decidiu utilizar dois sistemas de bancos de dados, tanto o PostgreSQL e o MongoDB, ele não deu motivo para tal, porém no caso dessa monografia creio que apenas um sistema deverá servir, já que segundo ([FONTAINE, 2019](#)) poucos problemas, necessitam mais que um framework de banco de dados, ele elogia especificamente o PostgreSQL que é bem parecido com o SQL, ferramenta que será usada nesse manuscrito. O autor do TCC citado anteriormente também decidiu fazer a parte da interface web com o React, porém, ele também achou necessário criar um aplicativo móvel para mais comodidade, no caso desse projeto, não vi necessidade para tal, já que inicialmente será um pequeno grupo que utilizará do programa.

## 4 Metodologia

### 4.1 Visão Geral

Esta monografia visa aprimorar os métodos que os pesquisadores de meteorologia da UFU utilizam para armazenar seus dados, diante da necessidade de adequar novos equipamentos à pesquisa e do uso de anos passados como referência para artigos e dissertações. Aqui, é proposta uma abordagem com o modelo de banco de dados relacional, incluindo a criação de tabelas e uma interface web.

Primeiramente, será necessário tratar os diversos dados ou padronizá-los, para que exista um diagrama de banco de dados relacional que consiga representá-los. Afinal, se os métodos padrões de criação de banco de dados forem seguidos passo a passo, é possível chegar em tabelas que satisfaçam todas as necessidades do problema, conforme mencionado em (ELMASRI; NAVATHE, 1994).

Após a definição do modelo relacional, serão criadas tabelas tendo em vista tal modelo. Como se trata de um banco de dados relacional, serão utilizadas chaves primárias e estrangeiras. Essa será a etapa em que se deve buscar a perfeição, visto que um erro aqui acarretará graves desdobramentos quando o banco de dados estiver sendo programado.

Em seguida, uma interface web será projetada para facilitar a interação dos pesquisadores com o banco de dados. Essa interface proporcionará uma maneira fácil de inserir dados meteorológicos por meio da importação de arquivos txt.

Por fim, serão realizados testes para verificar a integridade dos dados armazenados, a facilidade com que os pesquisadores acessam esses dados e se este aplicativo web é mais conveniente para os pesquisadores do que o método que utilizam atualmente.

### 4.2 Formato dos dados recebidos

Os dados registrados pelas estações meteorológicas são exportados no formato CSV. Esse formato geralmente segue uma estrutura tabular, onde cada linha do arquivo representa uma linha da tabela, com as vírgulas delimitando as colunas. A primeira linha do arquivo CSV contém informações como o modelo da estação, sua localização e outras especificações relevantes.

O elemento mais importante, no caso apresentado na Figura 1 (parte do CSV convertido em tabela), é o primeiro dado, identificado como "TOA5". Esse valor identifica a estação meteorológica e serve como o ID da estação na tabela SQL. É importante

ressaltar que, antes de inserir o arquivo CSV na interface web, o usuário precisa definir previamente algumas informações, incluindo as estações meteorológicas já existentes. Caso o primeiro dado, "TOA5", não esteja cadastrado no banco de dados como uma estação, os dados do arquivo não serão inseridos.

A segunda linha do CSV contém as variáveis coletadas pela estação, representando os diferentes sensores. Cada uma dessas variáveis, com exceção do "TIMESTAMP"(que indica a data e hora da coleta), representa um sensor específico. Assim como o ID da estação, esses sensores devem ser cadastrados como "tipos de dados"no sistema. Uma das variáveis utilizadas nas tabelas, por exemplo, é a "Temp-Avg"ou "Temp-Max", que indicam valores médios ou máximos de temperatura, respectivamente.

As terceira e quarta linhas do CSV trazem informações adicionais sobre a unidade de medida de cada variável. Por exemplo, "Deg C"na coluna "Temp-Avg"indica que os dados desta variável estão em graus Celsius. No entanto, essas linhas não são utilizadas na estrutura do banco de dados.

A partir da quinta linha, encontram-se os dados de leitura propriamente ditos, com cada linha representando um horário específico. Na estrutura do modelo de dados, esses valores são inseridos na tabela "faz-leitura"do modelo do banco de dados.

	A	B	C	D	E	F	G	H	I	J
1	TOA5	faz_gloria	CR20x	No_SN	CR200X.Std.01	estacao_cr200x_gloria_v6.CR2	3714	Diario		
2	TIMESTAMP	RECORD	Temp_Avg	Temp_Max	Temp_TmX	Temp_Min	Temp_TMn	UR_Avg	UR_Max	UR_TMx
3	TS	RN	Deg C	Deg C	Deg C	Deg C	Deg C			
4			Avg	Max	TMx	Min	TMn	Avg	Max	TMx
5	2015-10-24 00:00:00	0	NAN	32.07	2015-10-23 16:28:18	21.244	2015-10-23 23:51:48	73.60117	88.019	2015-10-23 23:53:18
6	2015-10-25 00:00:00	1	25.81514	34.115	2015-10-24 13:50:00	19.618	2015-10-24 05:47:36	65.10754	92.195	2015-10-24 01:57:18
7	2015-10-26 00:00:00	2	25.74313	32.667	2015-10-25 14:54:36	20.401	2015-10-25 23:59:36	63.66667	90.21	2015-10-25 23:58:24
8	2015-10-27 00:00:00	3	22.95668	28.939	2015-10-26 16:30:18	19.036	2015-10-26 06:03:00	80.5262	95.798	2015-10-26 06:16:18
9	2015-10-28 00:00:00	4	23.57354	32.058	2015-10-27 14:03:51	19.989	2015-10-27 05:47:33	79.81361	95.277	2015-10-27 22:38:42
10	2015-10-29 00:00:00	5	22.14987	29.49	2015-10-28 12:38:33	19.578	2015-10-28 04:45:57	87.59243	98.539	2015-10-28 06:12:42
11	2015-10-30 00:00:00	6	23.16046	31.938	2015-10-29 14:35:30	19.237	2015-10-29 05:34:21	83.27364	99.474	2015-10-29 06:19:48
12	2015-10-31 00:00:00	7	24.33376	30.558	2015-10-30 16:07:30	19.649	2015-10-30 00:48:24	76.51057	95.009	2015-10-30 03:42:45
13	2015-11-01 00:00:00	8	22.68131	28.339	2015-10-31 11:39:24	18.915	2015-10-31 22:10:51	81.01621	98.533	2015-10-31 23:58:15
14	2015-11-02 00:00:00	9	21.93992	29.839	2015-11-01 16:20:45	18.514	2015-11-01 23:10:51	87.2933	98.732	2015-11-01 00:41:57
15	2015-11-03 00:00:00	10	22.44232	28.259	2015-11-02 14:17:42	18.634	2015-11-02 00:31:12	85.84477	99.328	2015-11-02 05:30:18
16	2015-11-04 00:00:00	11	22.4575	29.398	2015-11-03 13:31:36	19.879	2015-11-03 23:23:54	88.42428	98.72	2015-11-03 04:43:54
17	2015-11-05 00:00:00	12	24.86477	32.677	2015-11-04 14:00:15	18.976	2015-11-04 03:53:12	73.24378	97.814	2015-11-04 03:00:15
18	2015-11-06 00:00:00	13	24.82921	32.658	2015-11-05 14:16:24	19.558	2015-11-05 04:30:57	74.87202	96.789	2015-11-05 03:06:48
19	2015-11-07 00:00:00	14	25.325	33.267	2015-11-06 15:54:03	19.237	2015-11-06 05:14:54	71.8549	94.107	2015-11-06 06:09:36
20	2015-11-08 00:00:00	15	23.81507	32.287	2015-11-07 13:17:03	19.809	2015-11-07 22:48:39	78.47782	96.455	2015-11-07 05:58:00
21	2015-11-09 00:00:00	16	25.18188	32.408	2015-11-08 15:28:18	18.212	2015-11-08 05:31:18	67.33856	90.941	2015-11-08 05:40:39
22	2015-11-10 00:00:00	17	25.88479	32.646	2015-11-09 13:39:21	20.371	2015-11-09 05:59:24	69.1179	88.82	2015-11-09 05:59:21
23	2015-11-11 00:00:00	18	26.7907	33.317	2015-11-10 14:54:48	20.653	2015-11-10 23:50:33	57.7596	84.552	2015-11-11 00:00:00
24	2015-11-12 00:00:00	19	26.66794	34.534	2015-11-11 16:23:12	20.08	2015-11-11 00:06:48	59.13585	91.941	2015-11-11 23:23:00
25	2015-11-13 00:00:00	20	23.99821	33.724	2015-11-12 13:57:36	19.558	2015-11-12 04:21:33	74.83202	95.384	2015-11-12 21:44:06
26	2015-11-14 00:00:00	21	25.79994	33.645	2015-11-13 13:53:42	19.317	2015-11-13 06:11:27	71.54264	96.07	2015-11-13 03:12:18
27	2015-11-15 00:00:00	22	24.39091	31.948	2015-11-14 15:50:00	18.453	2015-11-14 05:52:00	77.94453	98.011	2015-11-14 03:14:51

Figura 2 – CSV recebido transformado em uma tabela para melhor entendimento. Fonte: autor (2025).

### 4.3 Modelo do Banco de Dados

As tabelas do modelo do banco de dados se relacionam por meio de chaves primárias (cada linha da tabela possui uma chave única) e chaves estrangeiras (chaves que fazem referência a outra tabela). Como ilustrado na Figura 2, essa relação torna-se mais clara.

A tabela tipo de dado é criada pelo usuário; na interface web, o usuário deve inserir tanto a forma como o dado aparece no CSV (por exemplo, "Temp-Avg") quanto um nome de exibição. Esse nome será utilizado nas tabelas futuras para exibição, enquanto o ID é transformado em um número inteiro por meio de uma função hash.

A tabela leitura relaciona uma variável (Dado) a um valor específico (Valor). O ID da tabela é incremental, ou seja, começa em 0 e aumenta em 1 a cada nova entrada de leitura. Assim, supondo que existam cinco variáveis ou "tipos de dado" e dez registros de tempo (ou seja, dez linhas com diferentes timestamps), teremos cinquenta leituras no total.

A tabela sensor identifica qual leitura pertence a qual sensor. Informações adicionais, como localização e marca, não são essenciais para o funcionamento básico, mas podem ser úteis para o grupo de meteorologia comparar o desempenho e as leituras de sensores de diferentes fabricantes. O nome é similar ao nome do "tipo de dado" e serve apenas para exibição, enquanto o ID corresponde ao primeiro valor do CSV. Como mencionado anteriormente, o usuário precisa cadastrar os dados dos "sensores" antes de inserir os dados do CSV.

A tabela defeito não possui dados que existem no CSV, mas foi solicitada pelo grupo de meteorologia. Atualmente, a interface web permite apenas o cadastramento de defeitos. Ao inserir um "Faz-Leitura", o valor padrão para o ID de defeito é 0, indicando que não há defeito. A ideia futura é que seja possível marcar determinadas leituras como defeituosas, criando alertas para facilitar o monitoramento.

Por fim, a tabela faz leitura consolida todas as informações, unindo o sensor específico, a leitura, e o respectivo horário e data, o que gera um registro completo de leitura. Esse modelo elimina redundâncias e vai facilitar muito a exibição de dados em futuros projetos que aproveitem o banco de dados criado nessa monografia.

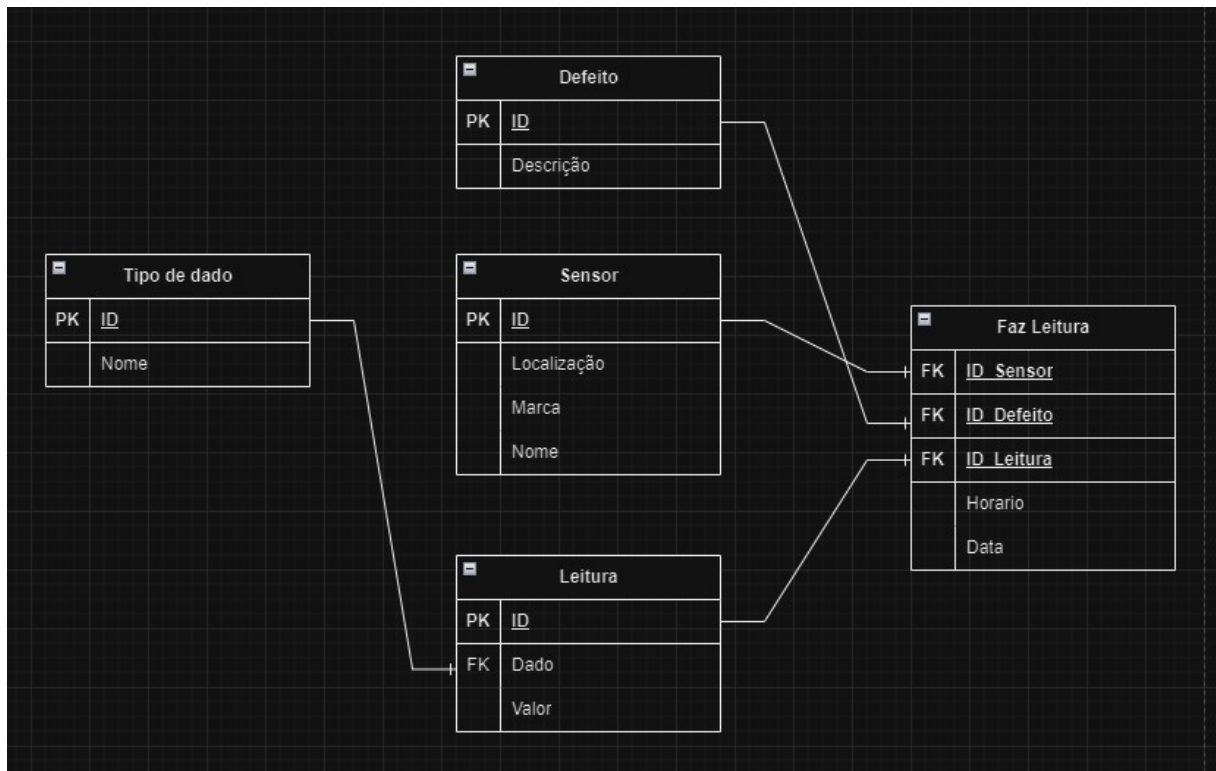


Figura 3 – Tabelas que representam o modelo do banco de dados. Fonte: autor (2025).

## 4.4 Captura dos dados e tratamento de colisões

O programa primeiramente organiza os timestamps em um vetor, em que `timestamp[0]` corresponde ao timestamp de todas as entradas faz leitura da primeira linha. Em seguida, ele obtém o primeiro dado do CSV, que representa o ID do sensor. Quando a requisição para enviar os dados ao backend é feita, o sistema verifica no banco de dados se já existe algum sensor com o mesmo ID encontrado no CSV. Caso exista, a inserção prossegue normalmente; caso contrário, o processo é interrompido.

As tabelas leitura e faz leitura são inseridas em uma única chamada. Como o ID de leitura é incremental, é necessário obter o último ID criado para relacioná-lo ao faz leitura. Antes disso, no entanto, é essencial tratar colisões, isto é, situações em que dados idênticos sejam enviados novamente.

Inicialmente, o algoritmo verifica a existência de registros com o mesmo identificador de estação informado. Essa estratégia é adotada em função do fato de que o número de estações é significativamente menor em comparação à quantidade de valores únicos encontrados em timestamps ou nos nomes das variáveis.

Em seguida, o programa analisa o tipo de dado, definido pelo nome da variável (por exemplo, unidade-do-ar). A partir dos dados contidos no CSV fornecido pela equipe de meteorologia, nota-se que essa dimensão apresenta o segundo menor número de unidades



distintas, o que leva a uma verificação mais rápida, caso uma leitura com um novo tipo de dado surja.

Posteriormente, o sistema procura registros com valores idênticos que correspondam à mesma estação e tipo de dado. Caso não haja registros com tais características, os novos dados são inseridos normalmente na base. Se existirem registros com valores iguais, o algoritmo realiza uma verificação adicional comparando os timestamps. Se estes também forem iguais, os dados são identificados como duplicados e não são inseridos na tabela.

Segue um pseudocódigo que resume o processo.

---

**Algorithm 1** Pseudocódigo para verificação dos dados

---

```

if dado.sensor in TabelaSQL then
  if dado.tipodedado == TabelaSQL[n].tipodedado then
    if dado.valor == TabelaSQL[n].valor then
      if dado.datahora == TabelaSQL[n].datahora then
        | dadosiguais();
      end
      else
        | dadosdiferente();
      end
    end
    else
      | dadosdiferente();
    end
  end
else
  | dadosdiferente();
end

```

---

## 4.5 Armazenamento

No banco de dados SQL utilizado para esta aplicação, o armazenamento dos dados é feito no disco local, preferencialmente em um servidor dedicado. De forma que a interface web esteja sempre acessível a todos os membros do time de meteorologia (ELMASRI; NAVATHE, 2015).

Fisicamente, os dados são organizados em blocos dentro de arquivos específicos, chamados arquivos de dados. Esses arquivos, por sua vez, são divididos em unidades menores chamadas páginas e extents. As páginas representam a menor unidade de leitura

e gravação para o sistema de banco de dados SQL, enquanto os extents são conjuntos de páginas que facilitam a alocação eficiente de espaço de armazenamento (DATE, 2003).

Quando um dado é inserido ou atualizado no sistema, essa informação é registrada em uma página específica, alocada no disco rígido ou SSD do servidor. Esse processo é eficiente em termos de desempenho, especialmente em sistemas com alta frequência de consultas e atualizações de dados o que não é o foco deste trabalho (ELMASRI; NAVATHE, 2015).

A segurança dos dados no SQL é garantida por meio de uma combinação de criptografia e controle de acesso. A criptografia dos dados pode ocorrer em nível de armazenamento, para proteger as informações contidas nos discos locais, e em nível de transmissão (SILBERSCHATZ; KORTH; SUDARSHAN, 2010).

Além disso, apenas usuários autorizados, que possuem a senha do banco de dados, têm permissão para realizar alterações que não estão previstas pelo sistema, ou seja, que não estão disponíveis na interface frontend, que possui as validações e restrições necessárias para proteger os dados.

Vale lembrar que o uso de servidores dedicados ou de soluções de armazenamento em nuvem oferece vantagens de segurança e flexibilidade. Com o armazenamento em nuvem, por exemplo, é possível configurar backups automáticos e redundância geográfica, protegendo os dados contra falhas físicas ou desastres naturais (ARMBRUST et al., 2010). Essa prática é cada vez mais comum, pois reduz a dependência de uma única localização física e facilita o escalonamento do armazenamento à medida que o volume de dados aumenta.

Para garantir a integridade dos dados, também são realizados backups periódicos, que podem ser configurados para ocorrer em intervalos específicos, como diários ou semanais. Os backups completos, diferenciais e incrementais são estratégias comumente adotadas para garantir a rápida recuperação dos dados em caso de falhas ou corrupções inesperadas (DATE, 2003).

## 5 Resultados e Discussões

Nesta seção, são apresentados os resultados obtidos a partir da implementação do sistema proposto e uma análise dos mesmos. Foram realizados testes para avaliar o desempenho do banco de dados relacional e da interface web, considerando aspectos como:

- Tempo de inserção dos dados;
- Precisão no tratamento de colisões;
- Facilidade de uso e acessibilidade da interface web.

Primeiro, é importante deixar claro as configurações da máquina onde esses testes foram realizados:

Tabela 1 – Configuração do Ambiente de Teste

<b>Componente</b>	<b>Especificação</b>
Processador	AMD Ryzen 7 5700G
Memória RAM	16 GB
Armazenamento	SSD de 256 GB
Sistema Operacional	Windows 11 Pro
Versão do SQL	MySQL 8.0

A seguir serão apresentadas figuras onde pode ser visualizada uma comparação entre as inserções com ordens diferentes. Uma ilustração que representa o teste 1 pode ser observado na figura 4, com a ordem de busca sendo Tipo de Dado → Valor da Leitura → Sensor → Data → Horário. Posteriormente a Tabela 2 resume os principais indicadores medidos durante os testes.

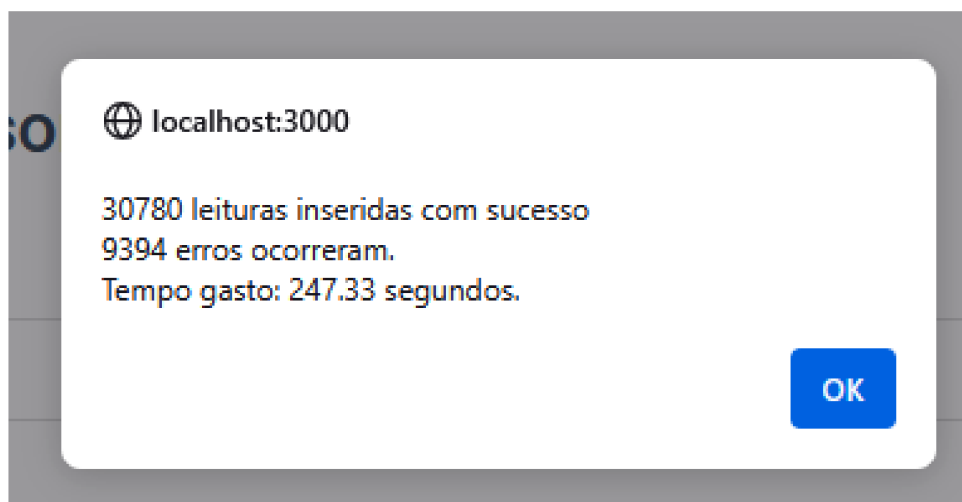


Figura 4 – Referente ao Teste 1: Numero de leituras, assim como o tempo gasto, os erros encontrados são os dados iguais tentando serem inseridos novamente. Fonte: autor (2025).

O teste 2 é ilustrado na Figura 5, é possível observar a ordem de busca: Sensor → Tipo de Dado → Data → Horário → Valor.

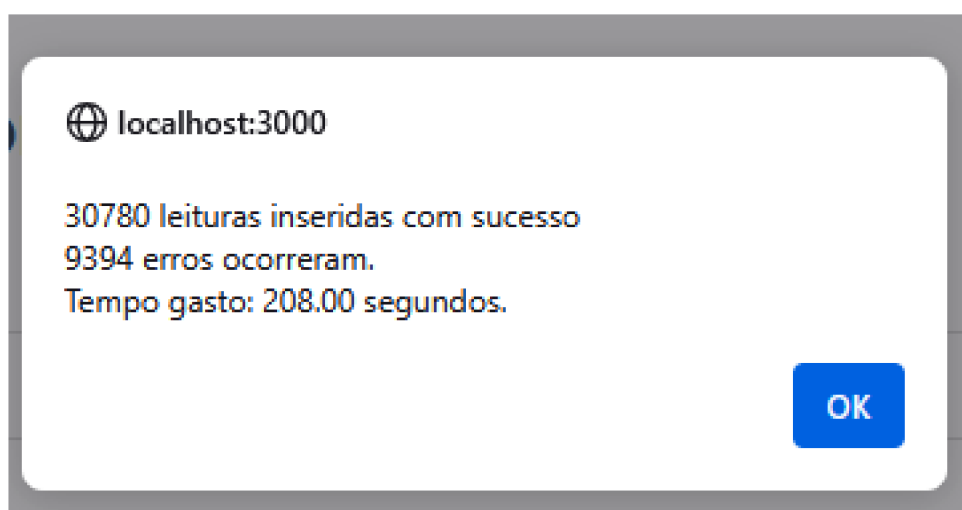


Figura 5 – Referente ao Teste 2: Numero de leituras, assim como o tempo gasto, os erros encontrados são os dados iguais tentando serem inseridos novamente. Fonte: autor (2025).

Por fim na Figura 6 é apresentado o teste 3, com a ordem de busca sendo Sensor → Tipo de Dado → Valor → Data → Horário.

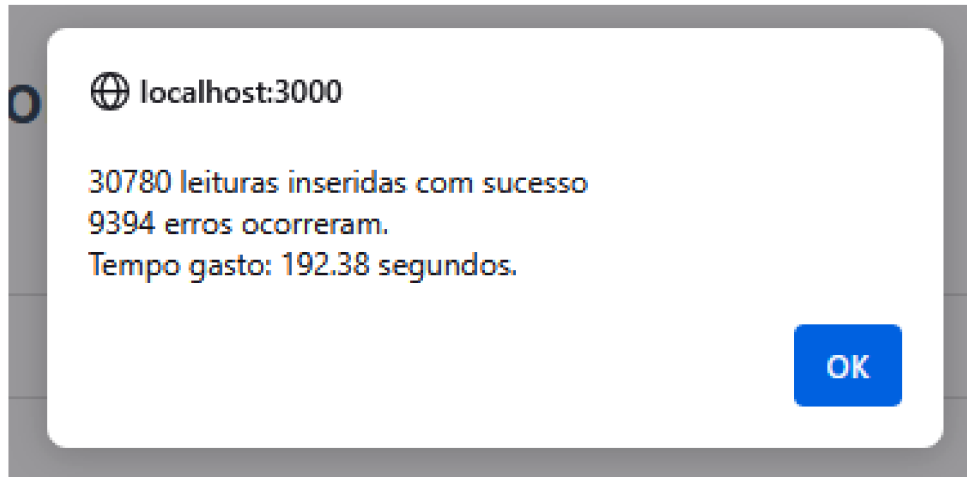


Figura 6 – Referente ao Teste 3: Numero de leituras, assim como o tempo gasto, os erros encontrados são os dados iguais tentando serem inseridos novamente. Fonte: autor (2025).

Tabela 2 – Comparação de desempenho entre diferentes configurações

Indicador	Teste 1	Teste 2	Teste 3
Tempo de inserção (s)	247,33	208,00	192,38
Taxa de acerto (%)	100	100	100
Dados inseridos	30 780	30 780	30 780

## 5.1 Análise Experimental do Tempo de Processamento de Dados em Planilha

Foi realizado um experimento envolvendo 10 voluntários, no qual inicialmente foram apresentadas as 10 primeiras linhas de um arquivo CSV. Cada participante foi instruído a organizar os dados em uma planilha do Excel, registrando o tempo necessário para a execução da tarefa. Adicionalmente, os voluntários foram indagados sobre o tempo estimado para organizar 3306 linhas – correspondentes a 30.780 leituras – e sobre a viabilidade de conclusão da tarefa.

### 5.1.1 Perfil dos Voluntários

Do grupo experimental, 4 voluntários afirmaram utilizar o Excel diariamente em suas atividades profissionais, 3 consideraram-se usuários de nível intermediário (não utilizando o programa diariamente, mas possuindo conhecimento de atalhos e comandos específicos), e os 3 restantes se definiram como usuários de nível básico.

### 5.1.2 Resultados Experimentais

- **Usuários Diários:** Dos participantes que utilizam o Excel cotidianamente, 3 completaram a tarefa em aproximadamente 1 minuto. Este desempenho deve-se ao conhecimento do comando específico que converte automaticamente um arquivo CSV em uma nova planilha. Ressalta-se, contudo, que tal comando possibilita apenas a criação de uma nova planilha, não sendo aplicável para a inserção de dados em planilhas já existentes, o que exigiria o emprego de outros métodos.
- **Usuários Intermediários e Básicos:** Para os demais voluntários, o tempo médio registrado foi de cerca de 7,5 minutos, devido ao fato de estes realizarem a inserção das linhas de forma manual.

Observa-se que, ao se considerar um volume de dados tão reduzido, a interface web proposta exibe um tempo de processamento da ordem de aproximadamente 1,05 segundos, o que mostra que até para os mais experientes, a interface web seria uma grande facilitadora.

### 5.1.3 Considerações sobre a Escalabilidade da Tarefa

A diferença de desempenho é particularmente notável entre os usuários classificados como intermediários e básicos. Quando indagados sobre a forma de proceder para a inserção das 3306 linhas, a maioria dos participantes estimou que a tarefa demandaria um período que poderia variar de várias horas a dias, ou que seria necessário recorrer a métodos alternativos para a inserção dos dados. Por outro lado, a conversão de um arquivo CSV para a criação de uma nova planilha Excel apresentava um tempo médio de execução em torno de 1 minuto, tempo esse gasto pelos usuários avançados.

Em comparação aos 190 segundos utilizados pela interface para o processamento das 3306 linhas, parece que o sistema é mais lento e ineficiente. Todavia, é necessário destacar que os dados processados pela interface dispõem de um nível de segurança significativamente superior ao oferecido por uma planilha Excel, ele dá acessibilidade universal a todos os usuários autorizados.

Ademais, a aplicação de consultas (queries) confere à interface uma versatilidade ampliada, permitindo a geração automatizada de gráficos e a realização de comparações detalhadas, procedimentos que seriam consideravelmente mais complexos em um ambiente de planilhas convencionais.

## 6 Conclusão

Este trabalho apresentou uma abordagem para o gerenciamento de dados meteorológicos, migrando de um método manual baseado em planilhas Excel para uma solução automatizada utilizando um banco de dados relacional e uma interface web.

Entre as principais contribuições, destacam-se:

- A automatização do processo de inserção de dados, que é bem mais veloz;
- A implementação de um sistema de tratamento de colisões, dando certeza que não existam dados duplicados;
- A flexibilidade do modelo proposto, que permite a inclusão de novos sensores e variáveis, no pior dos casos apenas mudando algumas linhas de código.

Com base nos testes realizados, notou-se que a solução apresenta ganhos importantes em desempenho e confiabilidade, além de oferecer uma plataforma mais consistente para a análise e comparação dos dados meteorológicos. Ainda assim, é necessário um futuro trabalho para que a interface web esteja pronta, esse que deverá mostrar os dados inseridos e realizar as consultas.

### 6.1 Trabalhos futuros

Este trabalho teve como proposta principal um sistema que agilize a captura dos dados capturados nas estações meteorológicas da UFU, a partir desse trabalho uma série de outras possibilidades de trabalhos se fazem presente, destacando-se:

- A integração em tempo real com os sensores;
- Criação de um sistema de segurança para que apenas aqueles com autorização possam enviar os dados;
- A realização de testes com cenários de alta demanda para validar a escalabilidade da solução.

Em suma, o trabalho apresentado facilitará o projeto do time de meteorologia da UFU; no entanto, ainda será preciso completá-lo em outro TCC futuro, ao adicionar as funções de exibição dos dados e comparações.

# Referências

ARMBRUST, M.; FOX, A.; GRIFFITH, R.; JOSEPH, A. D.; KATZ, R. H.; KONWINSKI, A.; LEE, G.; PATTERSON, D. A.; RABKIN, A.; STOICA, I.; ZAHARIA, M. A view of cloud computing. **Communications of the ACM**, Association for Computing Machinery, New York, NY, USA, v. 53, n. 4, p. 50–58, 2010. Citado na página 25.

BARBOSA, P.; FERREIRA, M.; NEVES, J. E. D. **Abordagem de segurança no desenvolvimento de aplicações web**. 2023. Artigo apresentado no Congresso de Segurança da Informação, Fatec Ourinhos. Disponível em: <<https://www.fatecourinhos.edu.br/fatecseg/index.php/fatecseg/article/view/107>>. Citado na página 16.

CARVALHO, T. R. d. S. **Explorando o desenvolvimento front-end: um estudo de caso em catálogo de filmes**. 2023. Trabalho de Conclusão de Curso, Pontifícia Universidade Católica de Goiás. Disponível em: <<https://repositorio.pucgoias.edu.br/jspui/handle/123456789/6997>>. Citado na página 15.

Cassim, H. T. Sanches, N. **Sistema de banco de dados e página HTML para estação meteorológica utilizando a plataforma Arduino**. 2022. <<https://ifpr.edu.br/goioere/wp-content/uploads/sites/13/2022/12/Artigo-Cientifico-Estacao-Meteorologica-Versao-Final.pdf>>. Citado na página 19.

CODD, E. F. **The relational model for database management: version 2**. 75 Arlington Street, Suite 300 Boston, MA United States: Addison-Wesley Longman Publishing Co., Inc., 1990. Citado na página 14.

DATE, C. J. **An Introduction to Database Systems**. 8th. ed. Boston, MA, USA: Addison-Wesley, 2003. Citado na página 25.

ELMASRI, R.; NAVATHE, S. B. **Fundamentals of Database Systems**. Reading, MA: Addison-Wesley, 1994. Citado na página 20.

\_\_\_\_\_. **Fundamentals of Database Systems**. 7th. ed. Boston, MA, USA: Pearson, 2015. Citado 2 vezes nas páginas 24 e 25.

FONTAINE, D. **The Art of PostgreSQL**. Paris: Apress, 2019. Citado na página 19.

FRANÇA, S. d. S. Web design responsivo: Caminhos para um site adaptável. **Interfaces Científicas - Exatas e Tecnológicas**, v. 1, n. 2, 2015. Disponível em: <<https://doi.org/10.17564/2359-4942.2015v1n2p75-84>>. Citado na página 17.

Giovani, Marllon Bianchini. **Análise sobre a Tecnologia HTML5**. 2012. <<https://pt.slideshare.net/slideshow/html-5-tcc-marllon-bianchini/23038222>>. Citado 2 vezes nas páginas 15 e 16.

GUIMARÃES, L. **Segurança no desenvolvimento web: análise de casos e um guia para mitigar exposições de dados sensíveis**. 2023. Trabalho de Conclusão de Curso, Universidade de Brasília. Disponível em: <<https://www.escavador.com/sobre/7777293/elaine-venson>>. Citado na página 16.



LOVATTO, P. L. **Estudo sobre o desenvolvimento de aplicações web com a linguagem JavaScript e a utilização do framework React**. 2022. Trabalho de Conclusão de Curso, Universidade Federal de Santa Maria. Disponível em: <https://www.ufsm.br/app/uploads/sites/584/2022/07/TCC-PedroLovatto.pdf>. Citado na página 17.

OLIVEIRA, D. P. **Proposta de interface baseada no Material Design para um aplicativo Android no domínio de barbearias**. 2022. Citado na página 15.

PAVANELI, T. H. **Desenvolvimento de uma API e uma Aplicação Web para auxiliar na análise de dados providos por Aplicação Móvel**. 2023. Citado na página 15.

PEREIRA, P. d. S. **Desenvolvimento de um front-end web para uma ferramenta de experimentação em cenários de mobile cloud computing**. 2022. Trabalho de Conclusão de Curso, Universidade Federal do Ceará. Disponível em: <https://repositorio.ufc.br/handle/riufc/68201>. Citado na página 15.

PEREIRA, W. R. **Desenvolvimento de software para gerenciamento de informações da secretaria de saúde de patos de minas e proposta de uma ferramenta para análise de dados**. 2022. Citado 3 vezes nas páginas 10, 14 e 19.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Database System Concepts**. 6th. ed. New York, NY, USA: McGraw-Hill, 2010. Citado na página 25.

SILVA, G. d. S. **Desenvolvimento de sistema para gerenciamento e histórico de fichas de pacientes em leitos de UTI**. 2021. Citado 2 vezes nas páginas 10 e 19.

SOUZA, T. Oliveira de. **COMUNICAÇÃO REAL-TIME COM NODE.JS E WEBSOCKETS**. 2016. Trabalho de Conclusão de Curso, Universidade Federal Fluminense. Disponível em: [https://app.uff.br/riuff/bitstream/handle/1/5267/TCC\\_THIAGO\\_OLIVEIRA\\_DE\\_SOUZA.pdf](https://app.uff.br/riuff/bitstream/handle/1/5267/TCC_THIAGO_OLIVEIRA_DE_SOUZA.pdf). Citado na página 17.