

**ESTÁCIO DE SÁ  
CAMPUS NOVA AMÉRICA**

**PROPOSTA DE PROJETO DE IOT COM TELA LCD E SENSOR DE TEMPERATURA PARA O  
AÇOUGUE E HORTIFRUTI DO CRIS**

**Nome do(s) discente(s) integrantes do grupo:**

**Antonio Marcos Silva de Oliveira - 202402574302**

**Pedro Henrique Cesario Justo da Silva - 202303351585**

**Cristhian Silveira Paulino Dos Santos - 202303300654**

**Elias Daniel Viana Alves Alvarenga 202303532458**

**Prof. Lucas Antunes Floriano**

**Link do GitHub**

[https://github.com/CristhianSilveira/sensor\\_temperatura\\_arduino.git](https://github.com/CristhianSilveira/sensor_temperatura_arduino.git)

**2025**

**Rio de Janeiro/RJ**

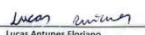
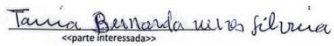
## Sumário

1. DIAGNÓSTICO E TEORIZAÇÃO.....	4
1.1. Identificação das partes interessadas e parceiros.....	4
1.2. Problemática e/ou problemas identificados .....	5
1.3. Justificativa .....	5
1.4. Objetivos/resultados/efeitos a serem alcançados (em relação ao problema identificado e sob a perspectiva dos públicos envolvidos) .....	6
1.5. Referencial teórico (subsídio teórico para propositura de ações da extensão) .....	6
2. PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO .....	7
2.1. Plano de trabalho (usando ferramenta acordada com o docente).....	7
2.2. Descrição da forma de envolvimento do público participante na formulação do projeto, seu desenvolvimento e avaliação, bem como as estratégias pelo grupo para mobilizá-los. ....	7
2.3. Grupo de trabalho (descrição da responsabilidade de cada membro) .....	8
2.4. Metas, critérios ou indicadores de avaliação do projeto .....	8
2.5. Recursos previstos .....	9
2.6. Detalhamento técnico do projeto .....	10
3. ENCERRAMENTO DO PROJETO .....	12
3.1. Relato Coletivo:.....	13
3.1.1. Avaliação de reação da parte interessada.....	13
3.2. Relato de Experiência Individual.....	15
3.2.1. CONTEXTUALIZAÇÃO .....	15
3.2.2. METODOLOGIA .....	16
3.2.3. RESULTADOS E DISCUSSÃO: .....	16
3.2.4. REFLEXÃO APROFUNDADA .....	17
3.2.5. CONSIDERAÇÕES FINAIS .....	17

# 1. DIAGNÓSTICO E TEORIZAÇÃO

## 1.1. Identificação das partes interessadas e parceiros

O projeto "Termômetro Inteligente" é desenvolvido em colaboração com o Açougue e Hortifruti do Cris, um estabelecimento comercial de pequeno porte que atua como mercearia e hortifruti, situado no bairro Anchieta, Rio de Janeiro, RJ. O projeto é voltado para o Açougue e Hortifruti do Cris um comercio local de pequeno porte que atua como mercearia e hortifruti, com interesse em modernizar sua loja. Os participantes do projeto têm escolaridade variada e pertencem a faixas etárias entre 18 e 60 anos. O projeto visa diretamente beneficiar o público-alvo do estabelecimento, que inclui moradores locais de diversas faixas etárias e perfis socioeconômicos, melhorando sua experiência de compra através de informações claras e dinâmicas.

CARTA DE APRESENTAÇÃO						
<p>Vimos por esta apresentar o grupo de acadêmicos da Universidade Estácio de Sá – UNESA listado na tabela a final deste documento, a fim de convidá-lo a participar de uma atividade extensionista associada à disciplina APLICAÇÃO DE CLOUD, IOT E INDÚSTRIA 4.0 EM PYTHON sob responsabilidade do Prof. Lucas Antunes Floriano.</p> <p>Em consonância ao Plano Nacional de Educação e demais normativas educacionais vigentes, a Universidade Estácio de Sá – UNESA desenvolve atividade extensionista que, norteados pela metodologia de aprendizagem baseada em projetos, tem por princípios fundantes o diagnóstico dos problemas/demandas/necessidades, a participação ativa dos interessados/públicos participantes, a construção dialógica, coletiva e experiencial de conhecimentos, o planejamento de ações, o desenvolvimento e avaliação das ações, a sistematização dos conhecimentos, a avaliação das ações desenvolvidas.</p> <p>Nesse contexto, a disciplina acima mencionada tem como principal escopo os temas relacionados à aplicação de cloud, aplicação de conceitos de IOT, projetos de automação com ferramentas de IOT.</p> <p>Sendo assim, pedimos o apoio dessa organização/entidade/coletivo/associação/outra, que aqui chamaremos de parte interessada, para a realização das seguintes atividades: diagnósticos, análises, entrevistas, levantamentos, projetos ou qualquer outra metodologia de estudo de caso que auxilie no desenvolvimento das competências de nossos acadêmicos e ao mesmo tempo possa contribuir para a comunidade em que estamos inseridos.</p> <p>Como se trata de atividade de ensino/aprendizagem de caráter extensionista, prevista no Projeto Pedagógico do Curso, salientamos que:</p> <ul style="list-style-type: none"><li>• não há cobrança de remuneração de qualquer natureza por parte da Universidade Estácio de Sá, seus alunos ou o docente da disciplina, à parte interessada;</li><li>• as atividades desenvolvidas no âmbito do projeto extensionista não configuram relação de trabalho entre os alunos e o docente da Universidade Estácio de Sá – UNESA disciplina APLICAÇÃO DE CLOUD, IOT E INDÚSTRIA 4.0 EM PYTHON, e a parte interessada;</li><li>• os resultados do projeto só poderão ser implantados para uso efetivo mediante Anotação de Responsabilidade Técnica de um profissional habilitado;</li><li>• os resultados do projeto podem ser implantados pela parte interessada para fins lucrativos, sem a necessidade de pagamento de quaisquer benefícios aos alunos, ao docente da disciplina e à Universidade Estácio de Sá – UNESA;</li></ul>	<ul style="list-style-type: none"><li>• quaisquer custos relativos à implantação e operação contínua do projeto fora do escopo das atividades do presente projeto serão arcados pela parte interessada.</li></ul> <p>Aproveitamos a oportunidade e solicitamos que, em caso de aceite, seja formalizado, mediante assinatura da Carta de Autorização, as atividades e informações que o(s) aluno(s) poderá(ão) ter acesso.</p> <p>Desde já nos colocamos à sua disposição para quaisquer esclarecimentos. Professor Lucas Antunes Floriano – 98820-4968 e/ou <a href="mailto:lucas.floriano@estacio.br">lucas.floriano@estacio.br</a> e -----</p> <table border="1"><thead><tr><th>Grupo de Alunos</th></tr></thead><tbody><tr><td>&lt;Elias Daniel Viana Alves Alvarenga&gt; Matricula: &lt;202303532458&gt;</td></tr><tr><td>&lt;Antonio Marcos Silva De Oliveira&gt; Matricula: &lt;202402574302&gt;</td></tr><tr><td>&lt;Cristhian Silveira Paulino Dos Santos &gt; Matricula: &lt;202303300654&gt;</td></tr><tr><td>&lt;Pedro Henrique Cesário&gt; Matricula: &lt;202303351585&gt;</td></tr></tbody></table> <p>Atenciosamente,  Lucas Antunes Floriano Docente da disciplina: APLICAÇÃO DE CLOUD, IOT E INDÚSTRIA 4.0 EM PYTHON Semestre: 2025.1 Matricula: 10643417</p> <p> &lt;&lt;parte interessada&gt;&gt;</p> <p>Rio de Janeiro, 15 de maio de 2025.</p>	Grupo de Alunos	<Elias Daniel Viana Alves Alvarenga> Matricula: <202303532458>	<Antonio Marcos Silva De Oliveira> Matricula: <202402574302>	<Cristhian Silveira Paulino Dos Santos > Matricula: <202303300654>	<Pedro Henrique Cesário> Matricula: <202303351585>
Grupo de Alunos						
<Elias Daniel Viana Alves Alvarenga> Matricula: <202303532458>						
<Antonio Marcos Silva De Oliveira> Matricula: <202402574302>						
<Cristhian Silveira Paulino Dos Santos > Matricula: <202303300654>						
<Pedro Henrique Cesário> Matricula: <202303351585>						



### 1.2. Problemática e/ou problemas identificados

Durante encontros e conversas preliminares com o proprietário e colaboradores do Açougue e Hortifruti do Cris, foi identificada uma lacuna na gestão e comunicação de informações ambientais cruciais para a conservação de produtos e para o conforto dos clientes. Especificamente, a problemática priorizada para este projeto reside na dificuldade de monitorar e comunicar de forma eficiente a temperatura ambiente em áreas de exposição de produtos sensíveis ao calor como hortifruti e no ambiente geral do açougue.

Esta deficiência impacta indiretamente a percepção de qualidade dos produtos, a eficiência energética e o conforto do cliente e dos colaboradores. A ausência de uma solução tecnológica acessível para este fim demanda a elaboração de um sistema de monitoramento ambiental dinâmico e de baixo custo.

### 1.3. Justificativa

A elaboração deste projeto de extensão é academicamente, pois se alinha diretamente aos objetivos de formação do curso de Aplic. de Cloud, IoT e Indústria 4.0 em Python, especialmente na aplicação de conhecimentos em eletrônica, programação e sistemas embarcados para a resolução de demandas reais.

A motivação do grupo de trabalho reside na oportunidade de integrar competências técnicas de programação em Arduino e Python com uma clara aplicação social e comercial.

Ao desenvolver um sistema de monitoramento de temperatura acessível, o projeto demonstra como inovações simples podem otimizar operações em comércios locais, contribuindo para a sustentabilidade e modernização de pequenos negócios.

#### 1.4. Objetivos/resultados/efeitos a serem alcançados (em relação ao problema identificado e sob a perspectiva dos públicos envolvidos)

- Implementar um Sistema de Monitoramento de Temperatura de Baixo Custo:
  - Montar um protótipo funcional composto por um microcontrolador Arduino, um sensor de temperatura LM35 . O sistema deverá ser capaz de realizar a leitura da temperatura ambiente e exibi-la de forma contínua na interface do programa.
- Desenvolver uma Interface de Monitoramento em Python:
  - Criar uma aplicação desktop em Python que se comunique via serial com o Arduino, capaz de receber e exibir a temperatura ambiente em tempo real. informações da temperatura ambiente atual.
- Demonstrar a Aplicabilidade de Tecnologias de Código Aberto em Comércio Locais:
  - Promover o entendimento sobre como soluções baseadas em Arduino e Python podem ser economicamente viáveis e eficazes para modernizar pequenos estabelecimentos, otimizando o monitoramento de condições ambientais e potencialmente a comunicação interna.

#### 1.5. Referencial teórico (subsídio teórico para propositura de ações da extensão)

A relevância do controle e monitoramento de temperatura é amplamente abordada na literatura de automação e eletrônica. **Gustavo Fernandes de Lima [1]** destaca a importância do "controle de temperatura em sistemas de baixo custo utilizando a placa Arduino" como um tema pertinente em iniciação científica, ressaltando a acessibilidade e versatilidade da plataforma. Essa perspectiva justifica a escolha do Arduino como base para o desenvolvimento do nosso protótipo, dadas as limitações orçamentárias de um pequeno comércio e a necessidade de uma solução didática e facilmente replicável.

Complementarmente, **João Carlos Segatto Leite, Crisleine Draszewski e Ederson Abaide [2]**, ao descreverem o "controle de temperatura em tanque através de Arduino baseado no controle On/Off", fornecem insights sobre a lógica de controle. Embora aplicada a um contexto diferente, essa lógica ilustra os princípios de sensoriamento (captação de dados de temperatura) e atuação que podem ser adaptados ao nosso objetivo de monitoramento e eventual acionamento de alertas. A simplicidade do modelo On/Off serve como base para compreender como o Arduino interpreta dados de sensores e toma decisões, mesmo que, em nosso caso, a "ação" inicial seja a exibição em uma interface de software.

Por fim, a abordagem de **Guilherme Nunes de Moraes Bueno [3]**, que explora o "controle de iluminação e temperatura pela plataforma Arduino via celular", expande a compreensão sobre a comunicação remota e a integração de sistemas. Embora nosso projeto inicial se concentre em uma interface desktop Python, o conceito de coletar dados de sensores (temperatura) e transmiti-los para uma plataforma externa para monitoramento e, futuramente, controle, valida a escolha da comunicação serial entre Arduino e Python. Esse caminho permite a construção de soluções mais complexas e interativas, que transcendem a simples exibição local. Este referencial sustenta a decisão de não apenas exibir a temperatura no local, mas também de transmiti-la para uma interface mais robusta, permitindo análises futuras e a expansão para controle remoto.

[\[1\] de Lima, Gustavo Fernandes. "Controle de temperatura de um sistema de baixo custo utilizando a placa arduino." IX Congresso de Iniciação Científica do IFRN. 2013.](#) | [\[2\] Leite, João Carlos Segatto,](#)

[Crisleine Draszewski, and Ederson Abaide. "Controle de Temperatura em Tanque Através de Arduino Baseado no Controle On/Off." IV Simpósio Paranaense de Modelagem, Simulação e Controle de Processos 19 \(2019\): 136-141. | \[3\] Bueno, Guilherme Nunes de Moraes. Controle de iluminação e temperatura pela plataforma Arduino via celular. BS thesis. Universidade Tecnológica Federal do Paraná, 2015.](#)

## 2. PLANEJAMENTO E DESENVOLVIMENTO DO PROJETO

### 2.1. Plano de trabalho (usando ferramenta acordada com o docente)

- Semana 1: Diagnóstico e Pesquisa Inicial
  - - Levantamento das necessidades do parceiro (Açougue e Hortifruti do Cris)
  - - Pesquisa e seleção dos componentes (Arduino UNO, sensor LM35, protoboard, resistores, jumpers).
  - - Definição dos objetivos e problemática.
- Semana 2: Concepção e Montagem Básica
  - - Montagem do circuito físico básico (Arduino, LM35) na protoboard, garantindo todas as conexões via protoboard.
  - - Configuração inicial dos códigos do Arduino e do Python .
- Semana 3: Desenvolvimento do Firmware Arduino
  - - Programação do Arduino para leitura de temperatura do LM35.
  - - Implementação da exibição da temperatura na interface utilizando o Python.
  - - Configuração da comunicação serial para envio dos dados de temperatura para o PC.
- Semana 4: Desenvolvimento da Interface Python
  - - Criação da interface gráfica em Python para recebimento e exibição da temperatura em tempo real.
  - - Implementação da comunicação serial Python-Arduino (pyserial).
  - - Refinamento da interface para simplicidade e usabilidade.
- Semana 5: Testes, Ajustes e Documentação
  - - Testes integrados do sistema (Arduino + Termistor + Python) em ambiente de sala de aula e no comércio parceiro.
  - - Coleta de feedback qualitativo do parceiro sobre a clareza e utilidade das informações.
  - - Ajustes finos no código e circuito com base nos testes.
  - - Elaboração final da documentação e relatório do projeto.

### 2.2. Descrição da forma de envolvimento do público participante na formulação do projeto, seu desenvolvimento e avaliação, bem como as estratégias pelo grupo para mobilizá-los.

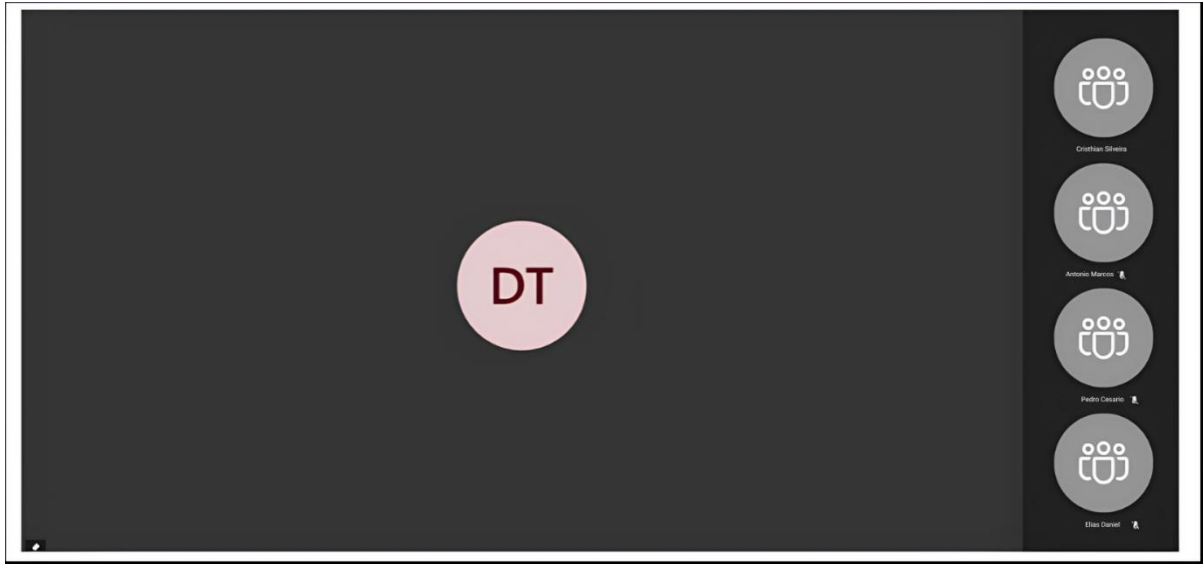
Foram realizados encontros e conversas informais com o comerciante. Através de uma escuta ativa, identificamos a necessidade de modernização e, especificamente, a carência de um sistema de



monitoramento ambiental de baixo custo para otimizar a conservação de produtos perecíveis e o conforto do cliente.

A ideia de um termômetro surgiu como uma demanda do próprio contexto do negócio, valorizando a construção conjunta da problemática.

O comerciante parceiro participou da definição das mensagens conforme as variações de temperatura. Também auxiliou nos testes e na instalação provisória do protótipo no local.



### 2.3. Grupo de trabalho (descrição da responsabilidade de cada membro)

- Elias:
  - Responsável pela integração física dos componentes, incluindo a montagem detalhada do circuito na protoboard e a verificação da conexão dos fios. Também contribuiu significativamente na montagem e depuração dos códigos em Python, garantindo a comunicação serial entre a interface e o Arduino.
- Antônio:
  - Foco na programação do sistema, sendo o principal responsável pela criação da estrutura do código Arduino, implementação da leitura do sensor e desenvolvimento da lógica para exibir a temperatura na interface Python. Contribuiu também com o apoio financeiro inicial para aquisição de alguns componentes de hardware.
- Cristhian:
  - Responsável pela refatoração e correção dos códigos em Python, garantindo a funcionalidade e estabilidade da interface gráfica. Liderou a documentação do projeto, incluindo a redação e formatação do relatório técnico e a organização das fotos e registros visuais.
- Pedro:
  - Auxiliou na aquisição de materiais e foi crucial na coleta de feedback junto ao parceiro comercial. Participou ativamente dos testes práticos do protótipo tanto em ambiente de sala de aula quanto no local de aplicação, trabalhando em conjunto com Elias na validação do funcionamento.

### 2.4. Metas, critérios ou indicadores de avaliação do projeto

- Meta 1: Implementar um Sistema de Monitoramento de Temperatura de Baixo Custo.



- Critério:
  - O protótipo deve ser capaz de realizar leituras de temperatura precisas e exibi-las na interface Python.
- Indicadores:
  - Funcionalidade da Leitura: O sistema apresenta leituras de temperatura de forma variada na interface no computador .
  - Conexão Serial: A interface Python estabelece e mantém comunicação serial estável com o Arduino.
  - Precisão das Leituras: A variação das leituras no Python não excede  $\pm 1^{\circ}\text{C}$  em comparação com um termômetro de referência.
  - Clareza no Software: A temperatura é exibida de forma legível no Computador.
- Meta 2: Demonstrar a Aplicabilidade da Tecnologia em Comércio Locais.
  - Critério:
    - A solução deve ser percebida como útil, relevante e de fácil adaptação pelo parceiro comercial.
  - Indicadores:
    - Aceitação do Parceiro: O proprietário e colaboradores consideram o sistema útil para o monitoramento do ambiente.
    - Facilidade de Uso: A interface e a operação do sistema são consideradas intuitivas pelos usuários finais.
    - Potencial de Aplicação: O parceiro identifica outras áreas onde tecnologias similares poderiam ser aplicadas em seu negócio.

## 2.5. Recursos previstos

- Materiais:
  - Kit Componentes Arduino UNO: R\$ 69,89
  - Arduino UNO R3 + Cabo USB: R\$ 33,98
  - Sensor de Temperatura Termistor (LM35): R\$ 15,98
  - Protoboard, Jumpers, Resistores (já inclusos no kit de componentes): R\$ 10,00
- Humanos:
  - Alunos do grupo de trabalho (Elias, Antônio, Cristhian, Pedro)
  - Professor Lucas, como orientador
  - Colaboradores do Açougue e Hortifruti do Cris (Tânia)
- Softwares:
  - Arduino IDE: Ambiente de desenvolvimento para programação do microcontrolador.
  - Python: Linguagem de programação para a interface gráfica e comunicação serial.
  - Git/GitHub: Para controle de versão do código e colaboração entre os membros da equipe.
  - Tinkercad: Ferramenta online para simulação do circuito eletrônico.
- Institucionais:
  - Apoio da instituição de ensino Estácio com infraestrutura (laboratórios, computadores, rede de internet) e orientação acadêmica através dos docentes responsáveis pela disciplina de extensão.

## 2.6. Detalhamento técnico do projeto

O projeto consiste no desenvolvimento de um sistema de monitoramento de temperatura ambiente utilizando o microcontrolador Arduino UNO. O circuito principal é composto por:

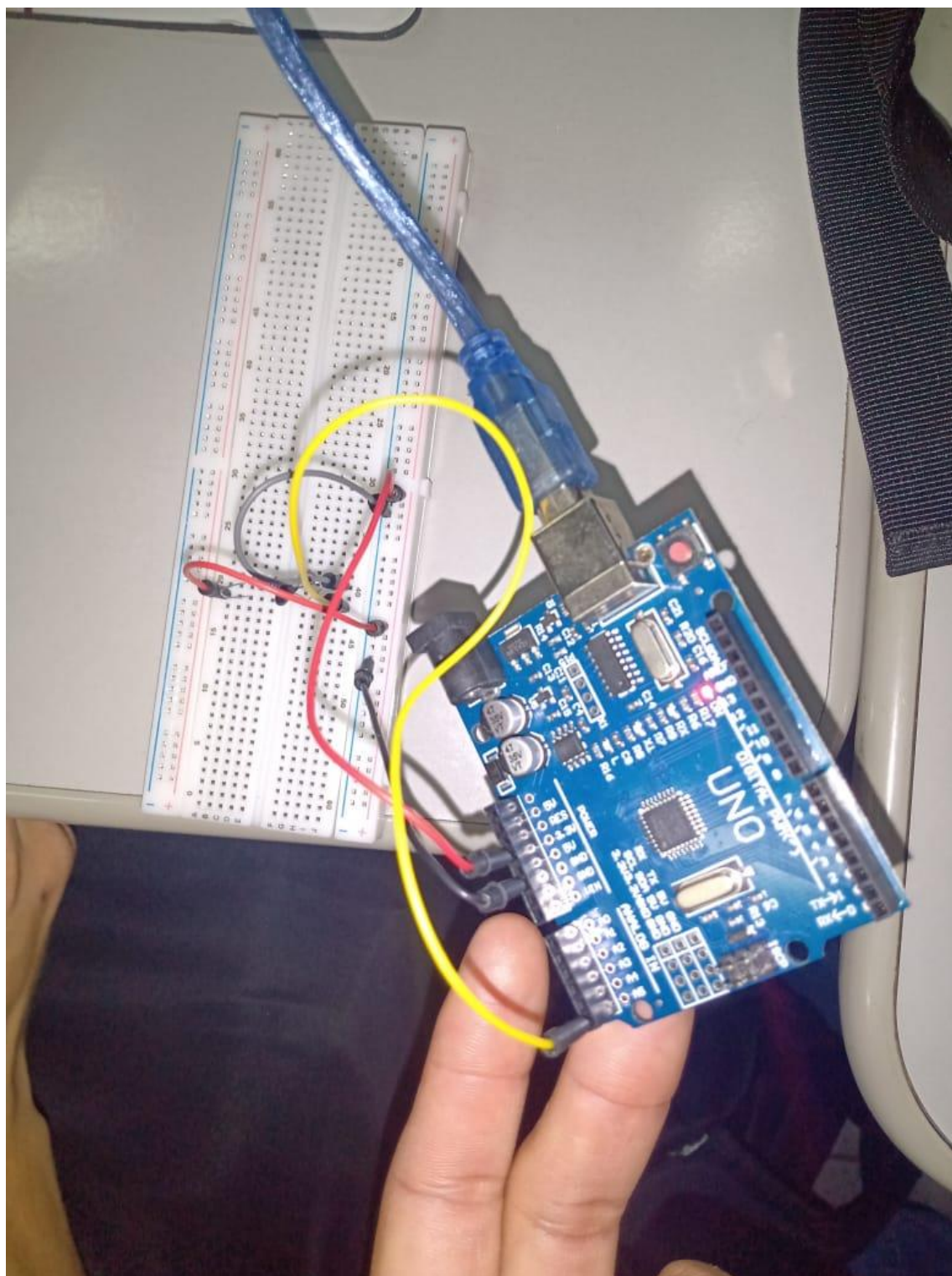
- Arduino UNO R3: Placa responsável pelo processamento dos dados do sensor .
- Sensor de Temperatura LM35: Um sensor analógico de precisão, que fornece uma tensão de saída proporcional à temperatura em graus Celsius.
- Protoboard: Utilizada para organizar todas as conexões entre o Arduino, o LM35 , garantindo que todos os fios passem por ela para maior clareza e facilidade de montagem e depuração.

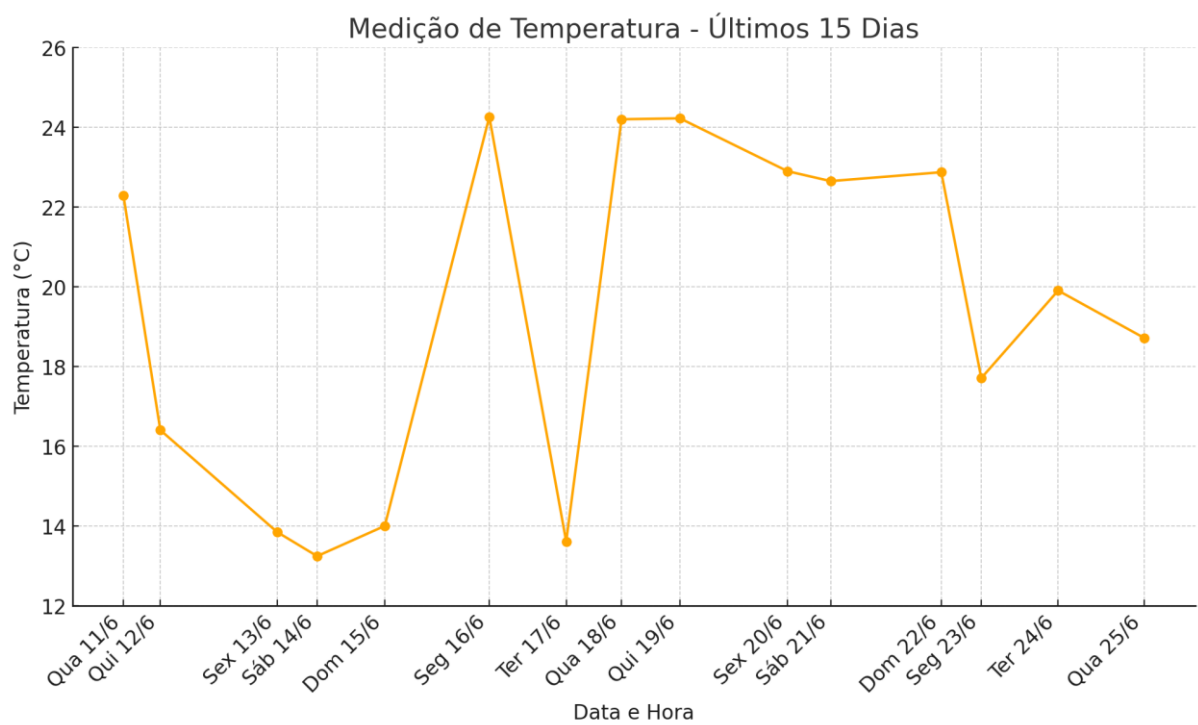
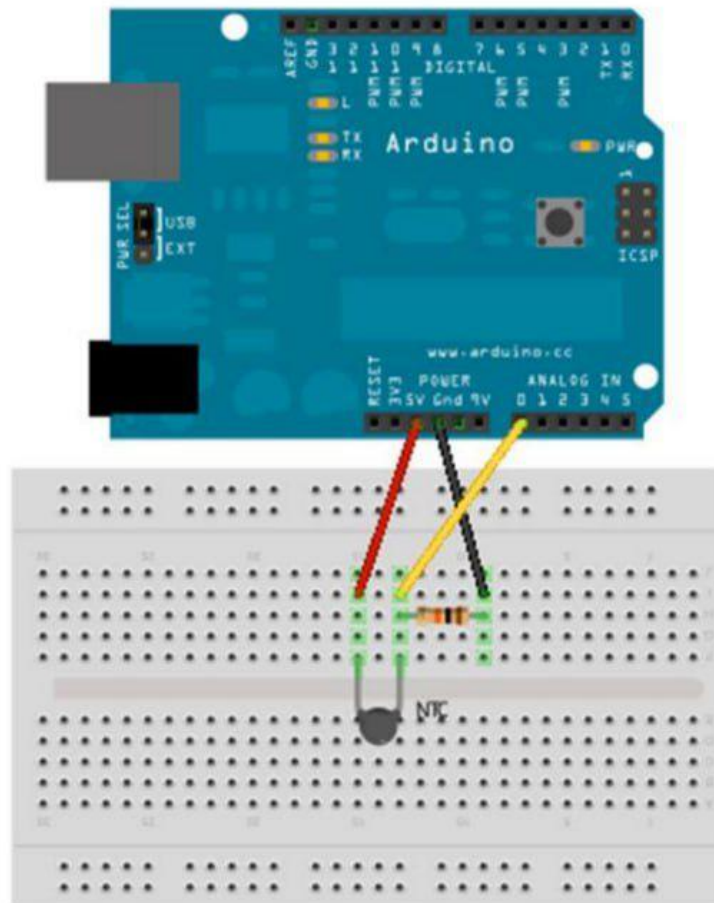
A lógica de funcionamento é implementada no Arduino utilizando a biblioteca padrão . O Arduino lê o valor analógico do LM35, converte-o para graus Celsius e exibe-o na interface do programa.

Paralelamente, foi desenvolvida uma interface gráfica em Python utilizando a biblioteca tkinter e pyserial. Esta interface se conecta à porta serial do Arduino, lê os dados enviados e exibe a temperatura em tempo real em uma janela do computador.

O protótipo foi testado com sucesso tanto em ambiente controlado de sala de aula quanto, de forma preliminar, no comércio parceiro, confirmando a precisão da leitura de temperatura e a estabilidade da comunicação entre o hardware e o software.

[https://github.com/CristhianSilveira/sensor\\_temperatura\\_arduino](https://github.com/CristhianSilveira/sensor_temperatura_arduino)





### 3. ENCERRAMENTO DO PROJETO

### 3.1. Relato Coletivo:

A execução do projeto atendeu satisfatoriamente aos objetivos propostos na fase de planejamento. A integração bem-sucedida entre hardware e software possibilitou a criação de uma solução prática e funcional para pequenos comércios, aliando tecnologia acessível à necessidade real de monitorar a temperatura ambiente, crucial para a conservação de produtos e o conforto no local. O grupo conseguiu exibir, em tempo real, a temperatura ambiente tanto em solo firme ou embarcado sendo exibido em uma interface gráfica intuitiva em Python.

Durante o desenvolvimento do projeto, foi notável o envolvimento engajado de todos os integrantes em cada etapa, desde a pesquisa e seleção dos componentes até a montagem do circuito, a programação e os testes em ambiente simulado e real. Essa experiência proporcionou um aprendizado técnico aprofundado em eletrônica, programação de microcontroladores e desenvolvimento de interfaces, além de reforçar habilidades interpessoais como o trabalho colaborativo, a comunicação eficaz e a resolução de problemas em equipe.

#### 3.1.1. Avaliação de reação da parte interessada

Segundo o relato, o projeto chamou a atenção de alguns clientes que passaram pela loja, o que evidenciou o potencial de engajamento visual que o sistema proporciona.

# Formulário de Avaliação do Projeto para AÇOUGUE E HORTIFRUTI DO CRIS

\* Indica uma pergunta obrigatória

Nome: \*

Tania Silveira

Função: \*

Gerente

Qual a sua satisfação com o projeto?

1



2



3



4



5



Acha que o projeto foi relevante pra sua empresa?

1



2



3



4



5



Você investiria em outro projeto com Arduino?

☒ Sim

☐ Não

Você faria alguma modificação neste projeto?

☐ Sim

☒ Não

Como foi a comunicação com a equipe envolvida no projeto?

1



2



3



4



5



Enviada: 14/06/2025, 21:32

## 3.2. Relato de Experiência Individual

### 3.2.1. CONTEXTUALIZAÇÃO

- Cristhian Silveira Paulino Dos Santos: Atuei na organização do conteúdo e documentação. Aprendi como descrever tecnicamente um projeto e a importância dos registros de cada etapa. Também participei dos testes e ajustes finais, refazendo o código para o funcionamento correto. A experiência foi enriquecedora do ponto de vista técnico e comunicacional.
- Antonio Marcos Silva de Oliveira: Fiquei responsável pela programação. A integração entre o sensor DHT11 exigiu atenção com bibliotecas específicas e o controle do fluxo de dados exibido. Senti realização ao ver o sistema funcionando corretamente e pude aprimorar muito minha lógica de programação com Arduino.
- Elias Daniel Viana Alves Alvarenga: Minha principal atividade foi a montagem do circuito. Aprendi a lidar com o Arduino e a fazer a ligação com o sensor de temperatura. Enfrentei dificuldades iniciais com o ajuste do contraste da tela, mas com ajuda da equipe e testes, conseguimos solucionar. A experiência foi valiosa, pois consegui aplicar conhecimento prático de eletrônica e fortalecer o trabalho em grupo.



- Pedro Henrique Cesario Justo da Silva: Fui responsável por aplicar o sistema no comércio e fazer os testes com o público. A experiência me mostrou como a tecnologia pode ter impacto direto na rotina das pessoas. Receber o feedback do comerciante foi uma parte gratificante do projeto e ajudou a validar a utilidade do sistema.

### 3.2.2. METODOLOGIA

A experiência do projeto foi vivenciada no Açougue e Hortifruti do Cris, um comércio local de pequeno porte que atuou como nosso parceiro para a aplicação prática. Os principais sujeitos envolvidos foram a equipe de desenvolvimento do projeto (Cristhian, Antonio, Elias e Pedro), Tania proprietária do Açougue e Hortifruti do Cris, e seus clientes, que indiretamente interagiram com o sistema.

- Cristhian Silveira Paulino Dos Santos (Organização, Documentação e Programação): Atuei na estruturação do conteúdo e registro técnico do projeto, garanti a documentação de cada fase do projeto e atuei na programação final do projeto.
- Antonio Marcos Silva de Oliveira (Programação do Sistema): Fui responsável pelo desenvolvimento do código para a integração do sensor DHT11 com o Arduino, focando no controle do fluxo de dados.
- Elias Daniel Viana Alves Alvarenga (Montagem do Circuito): Me dediquei à montagem física do circuito, conectando todos os componentes e o sensor de temperatura, e realizando ajustes de contraste.
- Pedro Henrique Cesario Justo da Silva (Aplicação e Testes em Campo): Realizei a instalação do sistema no comércio parceiro, executando testes com o público e coletando feedbacks do comerciante.

### 3.2.3. RESULTADOS E DISCUSSÃO:

- Cristhian Silveira Paulino Dos Santos: Estou impressionado com a qualidade do projeto. O esforço dos meus colegas superou as expectativas, houve uma boa comunicação que resultou em um trabalho de qualidade. O projeto resultou em um sistema capaz de cumprir o objetivo. Senti grande satisfação ao ver o projeto completo e bem apresentado. A maior aprendizagem foi a importância da documentação contínua e da comunicação para a eficácia de projetos em equipe.
- Antonio Marcos Silva de Oliveira: Fiquei satisfeito com o resultado final, houve boa comunicação entre os colegas de grupo e isso possibilitou um ótimo controle de qualidade dos componentes e no relatório durante o curso do projeto.
- Elias Daniel Viana Alves Alvarenga: Estou muito satisfeito com o trabalho dos meus colegas e meu próprio trabalho, e criamos algo extraordinário ao juntar as nossas partes.
- Pedro Henrique Cesario Justo da Silva: O resultado final é o que eu esperava, o projeto se saiu bem, a melhoria que poderia fazer seria coletar mais dados e dificuldades enfrentadas no comércio.

#### 3.2.4. REFLEXÃO APROFUNDADA

- Cristhian Silveira Paulino Dos Santos: Foi uma experiência de muito aprendizado. Aprendi coisas que jamais conseguiria sem a orientação do professor e a colaboração dos meus colegas.
- Antonio Marcos Silva de Oliveira: Me surpreendi com o que é possível fazer com o Arduino, aprendi bastante tanto nas aulas com o professor Lucas como em grupo com meus colegas.
- Elias Daniel Viana Alves Alvarenga: Foi uma ótima introdução ao mundo de IOT, aprendi muitos comandos na marra após ter problemas com o código do arduino e precisar sanitizá-la antes de torná-la útil para meus colegas.
- Pedro Henrique Cesario Justo da Silva: Eu concordo plenamente com o relato coletivo do grupo, acho que o trabalho foi muito produtivo, além de trazer um grande e significa para um primeiro projeto com Arduino.

#### 3.2.5. CONSIDERAÇÕES FINAIS

O projeto demonstrou que é possível implementar soluções tecnológicas simples, de baixo custo e alto impacto em ambientes que normalmente não contam com inovações. A interação entre os conhecimentos de hardware, software e necessidades comerciais resultou em um sistema funcional e com potencial de replicação.

Como desdobramento futuro, o sistema pode ser expandido para o controle de temperatura interno de loja, incluir sensores adicionais ou até conexão via Bluetooth/Wi-Fi para controle remoto. O projeto abre caminhos para mais ações de extensão envolvendo automação e inovação em contextos populares, fortalecendo a interdisciplinaridade entre tecnologia e sociedade.

cod\_arduino

```
1  #include <SPI.h>
2  #include <Adafruit_GFX.h>
3  #include <Adafruit_ST7789.h>
4  #include <math.h>
5
6  // Definições dos pinos do Arduino conectados ao display
7  #define TFT_CS    10  // Chip Select (CS)
8  #define TFT_DC    9   // Data/Command (DC)
9  #define TFT_RST   8   // Reset (RST)
10
11  Adafruit_ST7789 tft = Adafruit_ST7789(TFT_CS, TFT_DC, TFT_RST);
12
13  // Pino analógico onde o termistor está conectado
14  const int pinoNTC = A5;
15
16  const float R_SERIE = 10000.0; // Resistência do resistor em SÉRIE com 10kΩ.
17  const float BETA = 4000.0;
18  const float T0 = 298.15;
19  const float R0 = 10000.0;
20
21  void setup() {
22    Serial.begin(9600);
23
24    tft.init(240, 320);          // Display com sua resolução
25    tft.setRotation(1);
26    tft.fillScreen(ST77XX_BLACK); // Toda a tela com a cor preta
27    tft.setTextColor(ST77XX_GREEN); // Cor do texto para verde
28    tft.setTextSize(2);
29
30    tft.setCursor(50, 50);      // Define a posição do cursor (coluna, linha)
31    tft.print("Iniciando...");
32    delay(1000);
33  }
34
35
```



```

38         line = line_raw.decode('utf-8', errors='ignore').strip()
39         if line.startswith("TEMP:"):
40             print("Sincronização bem-sucedida!")
41             break
42         time.sleep(0.01)
43     except Exception as e:
44         print(f"Erro durante a sincronização serial: {e}")
45
46     while leitura_ativa:
47         if arduino_serial and arduino_serial.is_open:
48             try:
49                 linha = arduino_serial.readline().decode('utf-8', errors='ignore').strip()
50                 if linha.startswith("TEMP:"):
51                     try:
52                         temp_str = linha.split("TEMP:")[1].strip()
53                         temperatura = float(temp_str)
54                         root.after(10, lambda: atualizar_temperatura_na_interface(temperatura))
55                     except ValueError:
56                         print(f"Erro: Não foi possível converter temperatura de '{linha}'")
57
58             except serial.SerialException as e:
59                 if leitura_ativa:
60                     print(f"Erro na porta serial durante a leitura: {e}. Desconectando.")
61                     root.after(0, messagebox.showerror, "Erro de Leitura", f"Problema na comunicação serial: {e}")
62                     root.after(0, desconectar_arduino)
63                 break
64             except UnicodeDecodeError as e:
65                 print(f"Erro de decodificação: {e}. Dados brutos: '{linha.encode('utf-8')}'")
66             except Exception as e:
67                 print(f"Erro inesperado durante a leitura serial: {e}")
68         time.sleep(0.1)
69
70     def conectar_arduino():
71         global arduino_serial, leitura_ativa
72
73         if arduino_serial is None or not arduino_serial.is_open:

```

```

74         print("\nVerificando portas seriais disponíveis no sistema...")
75         ports = serial.tools.list_ports.comports()
76
77         if not ports:
78             status_label.config(text="Erro: Nenhuma porta!", foreground="red")
79             messagebox.showerror("Erro de Conexão", "Nenhuma porta serial encontrada. Conecte o Arduino e tente novamente.")
80             return
81
82         print("Portas seriais detectadas:")
83         for p in ports:
84             print(f" - {p.device} ({p.description})")
85
86         porta_para_conectar = SERIAL_PORT_PADRAO
87         if not any(p.device == porta_para_conectar for p in ports):
88             if ports:
89                 messagebox.showwarning("Aviso de Porta",
90                                         f"A porta '{SERIAL_PORT_PADRAO}' não foi encontrada.\n"
91                                         f"Tentando conectar à primeira porta disponível: '{ports[0].device}'")
92                 porta_para_conectar = ports[0].device
93             else:
94                 status_label.config(text="Nenhuma porta para conectar!", foreground="red")
95                 return
96
97         try:
98             print(f"\nTentando abrir conexão com a porta: '{porta_para_conectar}'...")
99             arduino_serial = serial.Serial(porta_para_conectar, BAUD_RATE, timeout=1)
100             time.sleep(2)
101
102             leitura_ativa = True
103
104             status_label.config(text=f"Conectado a {porta_para_conectar}", foreground="green")
105             btn_conectar.config(state=tk.DISABLED)
106             btn_desconectar.config(state=tk.NORMAL)
107             btn_enviar_email.config(state=tk.NORMAL)
108             btn_salvar_txt.config(state=tk.NORMAL)
109

```



```

110         if not any(t.name == 'serial_reader_thread' for t in threading.enumerate()):
111             threading.Thread(target=ler_dados_arduino, daemon=True, name='serial_reader_thread').start()
112             print("Conexão estabelecida com Arduino.")
113
114         except serial.SerialException as e:
115             status_label.config(text=f"Erro ao conectar: {e}", foreground="red")
116             messagebox.showerror("Erro de Conexão", f"Não foi possível conectar à porta '{porta_para_conectar}'".)
117             print(f"ERRO: Não foi possível conectar à porta '{porta_para_conectar}': {e}")
118             arduino_serial = None
119
120         except Exception as e:
121             status_label.config(text=f"Erro Inesperado: {e}", foreground="red")
122             messagebox.showerror("Erro", f"Ocorreu um erro inesperado ao tentar conectar.\nDetalhes: {e}")
123             print(f"ERRO INESPERADO: {e}")
124
125     def desconectar_arduino():
126         global arduino_serial, leitura_ativa
127         leitura_ativa = False
128
129         if arduino_serial and arduino_serial.is_open:
130             try:
131                 arduino_serial.close()
132                 print("Comunicação serial encerrada.")
133             except serial.SerialException as e:
134                 print(f"Erro ao fechar a porta serial: {e}")
135             except Exception as e:
136                 print(f"Erro inesperado ao desconectar: {e}")
137
138             status_label.config(text="Arduino Não Conectado", foreground="red")
139             btn_conectar.config(state=tk.NORMAL)
140             btn_desconectar.config(state=tk.DISABLED)
141             btn_enviar_email.config(state=tk.DISABLED)
142             btn_salvar_txt.config(state=tk.DISABLED)
143             lbl_temperatura.config(text="Temperatura: -- °C")
144             print("Desconectado com sucesso.")
145

```

```

146     def enviar_email():
147         email_destino = entry_email_destino.get()
148
149         if not email_destino or "@" not in email_destino:
150             messagebox.showwarning("Erro", "Por favor, insira um endereço de e-mail de destino válido.")
151             return
152         if ultima_temperatura_lida is None:
153             messagebox.showwarning("Aviso", "Nenhuma temperatura lida para enviar por e-mail.")
154             return
155
156         msg = MIMEText(corpo_email, 'plain')
157         msg['From'] = EMAIL
158         msg['To'] = email_destino
159         msg['Subject'] = f"Relatório de Temperatura - {time.strftime('%d/%m/%Y %H:%M:%S')}"
160
161         corpo_email = f"""
162         Prezado Cliente,
163
164         Este é um relatório automático da temperatura ambiente registrada pelo seu sensor.
165
166         Temperatura Atual: {ultima_temperatura_lida:.2f} °C
167         Data e Hora do Registro: {time.strftime('%d/%m/%Y %H:%M:%S')}
168         (Fuso Horário: {time.tzname[0]})
169
170         Atenciosamente,
171         Cristhian
172         """
173         msg.attach(MIMEText(corpo_email, 'plain'))
174
175         try:
176             server = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
177             server.starttls()
178             server.login(EMAIL, SENHA_APP)
179             text = msg.as_string()
180             server.sendmail(EMAIL, email_destino, text)
181             server.quit()

```

```

182     messagebox.showinfo("Sucesso", f"E-mail com a temperatura enviado para {email_destino}!")
183     print(f"E-mail enviado para {email_destino} com a temperatura {ultima_temperatura_lida:.2f}°C")
184 except smtplib.SMTPAuthenticationError:
185     messagebox.showerror("Erro de Autenticação", "Falha de autenticação. Verifique seu e-mail e SENHA DE APP.")
186     print("Erro de autenticação. Verifique as credenciais.")
187 except smtplib.SMTPException as e:
188     messagebox.showerror("Erro de Envio", f"Erro ao enviar e-mail: {e}.\nVerifique as configurações SMTP, sua conexão à internet ou as
189     permissões de envio do seu e-mail.")
190     print(f"Erro SMTP ao enviar e-mail: {e}")
191 except Exception as e:
192     messagebox.showerror("Erro Geral", f"Ocorreu um erro inesperado ao enviar o e-mail: {e}")
193     print(f"Erro inesperado no envio de email: {e}")
194
195 def salvar_registro_txt():
196     if ultima_temperatura_lida is None:
197         messagebox.showwarning("Aviso", "Nenhuma temperatura lida ainda para salvar.")
198         return
199
200     file_path = filedialog.asksaveasfilename(
201         defaultextension=".txt",
202         filetypes=[("Arquivos de Texto", "*.txt"), ("Todos os Arquivos", "*.*")],
203         title="Salvar Registro de Temperatura"
204     )
205
206     if not file_path:
207         return
208
209     try:
210         timestamp = time.strftime('%d/%m/%Y %H:%M:%S')
211         registro = f"[{timestamp}] Temperatura: {ultima_temperatura_lida:.2f} °C\n"
212
213         with open(file_path, 'a') as f:
214             f.write(registro)
215
216     except Exception as e:
217         messagebox.showerror("Erro ao Salvar", f"Não foi possível salvar o registro: {e}")
218         print(f"Erro ao salvar registro TXT: {e}")

```

```

216     print(f"Registro salvo: {registro.strip()} em {file_path}")
217 except Exception as e:
218     messagebox.showerror("Erro ao Salvar", f"Não foi possível salvar o registro: {e}")
219     print(f"Erro ao salvar registro TXT: {e}")
220
221
222 root = tk.Tk()
223 root.title("Monitor e Registro de Temperatura")
224 root.geometry("450x450")
225 root.resizable(False, False)
226
227 style = ttk.Style()
228 style.theme_use('clam')
229 style.configure("TFrame", background="#f0f0f0")
230 style.configure("TButton", font=("Helvetica", 10), padding=5, background="#e0e0e0", foreground="black")
231 style.map("TButton", background=[('active', '#c0c0c0')])
232
233 style.configure("Connect.TButton", background="#4CAF50", foreground="white")
234 style.map("Connect.TButton", background=[('active', '#45a049')])
235
236 style.configure("Disconnect.TButton", background="#FF4C4C", foreground="white")
237 style.map("Disconnect.TButton", background=[('active', '#e04040')])
238
239 style.configure("Email.TButton", background="#0078FF", foreground="white")
240 style.map("Email.TButton", background=[('active', '#0056b3')])
241
242 style.configure("Save.TButton", background="#8B4513", foreground="white")
243 style.map("Save.TButton", background=[('active', '#69340f')])
244
245 style.configure("TLabel", font=("Helvetica", 12), background="#f0f0f0")
246 style.configure("Title.TLabel", font=("Helvetica", 16, "bold"), background="#f0f0f0")
247 style.configure("Temp.TLabel", font=("Helvetica", 24, "bold"), foreground="#0056b3", background="#f0f0f0")
248
249
250 main_frame = ttk.Frame(root, padding="20")

```



```

251 main_frame.pack(fill=tk.BOTH, expand=True)
252
253 lbl_titulo = ttk.Label(main_frame, text="Monitor e Registro de Temperatura", style="Title.TLabel")
254 lbl_titulo.pack(pady=10)
255
256 lbl_temperatura = ttk.Label(main_frame, text="Temperatura: -- °C", style="Temp.TLabel")
257 lbl_temperatura.pack(pady=10)
258
259 status_label = ttk.Label(main_frame, text="Arduino Não Conectado", font=("Helvetica", 10), foreground="red")
260 status_label.pack(pady=5)
261
262 button_frame = ttk.Frame(main_frame)
263 button_frame.pack(pady=10)
264
265 btn_conectar = ttk.Button(button_frame, text="Conectar", command=conectar_arduino, style="Connect.TButton")
266 btn_conectar.grid(row=0, column=0, padx=5)
267
268 btn_desconectar = ttk.Button(button_frame, text="Desconectar", command=desconectar_arduino, state=tk.DISABLED, style="Disconnect.TButton")
269 btn_desconectar.grid(row=0, column=1, padx=5)
270
271 controls_frame = ttk.LabelFrame(main_frame, text="Personalização", padding="10")
272 controls_frame.pack(pady=15, fill=tk.X, padx=10)
273
274 ttk.Label(controls_frame, text="E-mail de Destino:", background="#f0f0f0").pack(anchor='w')
275 entry_email_destino = ttk.Entry(controls_frame, width=40, font=("Helvetica", 10))
276 entry_email_destino.pack(pady=5, fill=tk.X)
277
278 btn_enviar_email = ttk.Button(controls_frame, text="Enviar E-mail com Temperatura", command=enviar_email, style="Email.TButton", state=tk.DISABLED)
279 btn_enviar_email.pack(pady=5)
280
281 btn_salvar_txt = ttk.Button(controls_frame, text="Salvar Registro (TXT)", command=salvar_registro_txt, style="Save.TButton", state=tk.DISABLED)
282 btn_salvar_txt.pack(pady=5)
283
284

```

```

285 def ao_fechar_janela():
286     if messagebox.askokcancel("Sair", "Deseja realmente sair do Monitor de Temperatura?"):
287         desconectar_arduino()
288         root.destroy()
289
290 root.protocol("WM_DELETE_WINDOW", ao_fechar_janela)
291
292 root.mainloop()
293
294 if arduino_serial and arduino_serial.is_open:
295     arduino_serial.close()
296     print("Porta serial fechada.")

```

## Monitor e Registro de Temperatura

# Temperatura: 34.60 °C

Conectado a COM11

Conectar

Desconectar

### Personalização

E-mail de Destino:

lucasfloriano@poli.ufrj.br

Enviar E-mail com Temperatura

Salvar Registro (TXT)