



**KeepGrowing  
Gerdau**



## Controle do IoTDoc - documentação geral do projeto

### Histórico de revisões

Data	Autor	Versão	Resumo da atividade
13/10/2022	Carolina Fricks Giovanna Rodrigues Rafael Katalan	1.1	Criação do documento IoTDoc Preenchimento das seções: 1.1, 1.2.1, 1.2.2, 1.3.3
19/10/2022	Carolina Fricks Giovanna Rodrigues João Marques Rafael Katalan	1.2	Preenchemos as seções: 1.1, 1.3.1, 1.3.2, 1.3.4, 1.3.5, 1.4.1, 1.4.2, 2.1
20/10/2022	Gábrio lina	1.3	Preenchimento da seção 1.3 Detalhamento da seção 1.3.4
23/10/2022	Giovanna Rodrigues	1.4	Preenchimento da seção 1.4.3 Revisão
01/11/2022	João Marques, Giovanna Rodrigues e Carolina Fricks	2.1	Preenchimento da seção 3.1 e 2.2
03/11/2022	Carolina Fricks e Giovanna Rodrigues	2.2	Preenchimento e revisão das seções 1.4.4, 2.2, 3.1
05/11/2022	Gábrio Lina	2.3	Criação e preenchimento das seções 1.4.4.1, 1.4.4.2, 1.4.4.3
06/11/2022	Carolina Fricks e Gábrio Lina	2.4	Revisão das seções 1.4.4.1, 1.4.4.2, 1.4.4.3
19/11/2022	Marcos Moura	3.1	Preenchimento da seção 3.2
21/11/2022	Marcos Moura	3.2	Atualização da Matriz de Risco
21/11/2022	Carolina Fricks, Gábrio Lina, Giovanna Rodrigues, Marcos Moura e Rafael Katalan	3.3	Preenchimento da seção 2.3 e subseções 2.3.1, 2.3.2, 2.3.3.  Complemento da seção 3.2 Criação e preenchimento da seção 4, 4.1, 4.2 e 4.3

			Atualização do Sumário do documento
21/11/2022	João Marques	3.4	
21/11/2022	Carolina Fricks, Gábrio Lina, Giovanna Rodrigues, João Marques e Marcos Moura	3.5	Preenchimento da seção 4.1, 4.2 e subseções  Preenchimento da seção 4.3 e subseções  Preenchimento da seção 4.4 e subseções
22/11/2022	Carolina Fricks e Gábrio Lina	4.1	Revisão e detalhamento da seção 4.4
05/12/2022	Carolina Fricks e Gábrio Lina	4.2	Criação e preenchimento da seção 2.4
15/12/2022	Marcos Moura Carolina Fricks	Versão final	Revisão de todo o documento

# Sumário

## 1. Definições Gerais

- 1.1. Parceiro de Negócios
- 1.2. Definição do Problema e Objetivos
  - 1.2.1. Problema
  - 1.2.2. Objetivos
- 1.3. Análise de Negócio
  - 1.3.1. Contexto da indústria
  - 1.3.2. Análise SWOT
  - 1.3.3. Planejamento Geral da Solução
  - 1.3.4. Value Proposition Canvas
  - 1.3.5. Matriz de Riscos
- 1.4. Análise de Experiência do Usuário
  - 1.4.1. Personas
  - 1.4.2. Jornadas do Usuário e/ou Storyboard
  - 1.4.3. User Stories
  - 1.4.4. Protótipo de interface com o usuário

## 2. Arquitetura da solução

- 2.1. Arquitetura versão
- 2.2. Arquitetura versão
- 2.3. Arquitetura versão

## 3. Situações de uso

- 3.1. Entradas e Saídas por Bloco
- 3.2. Interações

## 4. Front-end, API e Banco de Dados

- 4.1. Front-end
- 4.2. API
- 4.3. Banco de Dados

# 1. Definições Gerais

## 1.1. Parceiro de Negócios

Fundada em 1901, a Gerdau começou quando João Gerdau adquiriu a Cia. A Fábrica de Pregos Pontas de Paris em Porto Alegre (RS), hoje, possui 120 anos de história contribuindo para uma sociedade sempre em evolução e sustentabilidade. No ramo da siderurgia, se tornou a maior empresa brasileira produtora de aço e expandiu seu negócio para outros setores, fundando então a Gerdau Next, focando em desenvolvimento, participação ou controle de empresas no setor de construção, logística, infraestrutura e energia renovável, além de aceleração e fundo de investimento em startups.

Mundialmente, está presente em 10 países, sendo a maior multinacional brasileira no ramo do aço, tem suas ações listadas nas bolsas de valores de São Paulo (B3), Nova Iorque (NYSE) e Madri (Latibex), e a maior recicladora da América Latina, tendo 73% do seu aço produzido a partir da sucata.

Dentre seus outros setores, há a Gerdau Florestal, responsável pelo plantio de eucalipto para a produção de carvão vegetal, um biorredutor, sendo uma solução sustentável na produção siderúrgica. Conta com 250 mil hectares de base florestal, contendo o plantio de eucalipto, mas também, áreas de preservação.

## 1.2. Definição do Problema e Objetivos

### 1.2.1. Problema

A Gerdau possui hoje uma quantidade de 1,6 milhão de mudas de eucalipto por mês, isso demanda uma quantidade muito alta de funcionários para o monitoramento das condições (temperatura e umidade) dessas mudas de hora em hora, para que as condições estejam favoráveis para o seu crescimento.

Contudo, o intervalo de tempo entre monitoramentos combinado com condições desfavoráveis para a planta podem aumentar drasticamente o seu risco de mortalidade, uma vez que o responsável por fazer a manutenção do ambiente não terá o conhecimento de que é necessário algum ajuste até a próxima medição, podendo manter o plantio por muito tempo em condições impróprias para a sua sobrevivência e levando a sua morte, já que são sensíveis e muito suscetíveis ao seu ambiente.

Consequentemente, nota-se uma quantidade escassa de dados coletados, por serem medidos a cada hora e apenas em horário comercial, logo, não se tem registros do comportamento do ambiente durante o período noturno. Isso dificulta a ocorrência de uma análise mais profunda da rotina desse plantio e a tomada de decisões futuras buscando a otimização do processo de enraizamento dos eucaliptos.

Além disso, as medições e os dados registrados em um formulário digital são feitas manualmente, então, o processo acaba ocupando uma parcela de tempo desnecessária dos funcionários, já que poderia ser realizada automaticamente com o uso de tecnologia e os operadores teriam um melhor aproveitamento de sua carga de trabalho com outras atividades.

### 1.2.2. Objetivos

O objetivo da proposta de solução é automatizar a medição de temperatura e umidade relativa das estufas de eucalipto da Gerdau Florestal através da utilização de sensores e um microcontrolador conectados à internet que enviam esses dados para o banco de dados e notificarão os funcionários em caso de necessidade na manutenção das condições do ambiente, como a abertura das janelas laterais e zenitais.

Além de efetivar ao máximo esta etapa do processo de plantação de eucaliptos, a solução visa otimizar o H/H de todos os colaboradores da empresa, podendo ser reposicionados para tarefas que gerem maior valor.

## 1.3. Análise de Negócio

### 1.3.1. Contexto da indústria

#### Principais Players

O cenário atual do mercado siderúrgico é bem definido com várias empresas que dominam grande parte desse mercado. Destacam-se as empresas CSN e Usiminas.

#### CSN:

A Companhia Siderúrgica Nacional é a maior siderúrgica do Brasil e a sexta maior exportadora do minério de ferro. Seu principal foco é a produção de aço bruto e a extração do minério de ferro, estanho e carvão.

### **Usiminas:**

A Usiminas (Usinas Siderúrgicas de Minas Gerais S.A) faz parte do setor siderúrgico e é líder na produção e comercialização de aços planos. A empresa está entre os 20 maiores complexos siderúrgicos de aços planos e o maior na América Latina. A empresa está presente em cinco estados e 13 cidades brasileiras.

### **Rivalidade entre concorrentes:**

A rivalidade é considerada baixa ao levar em consideração a pouca diversidade entre concorrentes, já que há a existência de um mercado estruturado em apenas três usinas consolidadas, entre elas a Gerdau, e também, um grau de diferenciação entre produtos muito pequeno, mas uma demanda grande, logo, favorável para aqueles que já estão consolidados.

### **Relação com Clientes:**

A maior parte dos clientes da companhia se encontram nos setores automobilístico e civil. A necessidade de altas quantidades de metais como ferro, para construção de prédios e carros por exemplo, e a alta concentração de empresas nestes setores, providencia uma capacidade elevada de negociação, visto que há uma gama de empresas e organizações clientes que necessitam dos recursos mencionados, de forma constante. Entretanto, a presença de novos entrantes internacionais, e o fechamento de contratos exclusivos pode afetar a relação com clientes presentes.

### **Fornecedores:**

Devido a alta capacidade de produção, quantidade de ferramentas e da abrangência dos setores de atuação da Gerdau, a maior parte de seus insumos e matéria-prima são produzidos dentro da própria empresa e subsidiárias. Isso diminui drasticamente qualquer risco relacionado à dependência de fornecedores ou terceiros, não sendo uma ameaça alta, além de facilitar o rastreio de matérias-primas e acelerar a sua linha de produção. Entretanto, o fornecimento de recursos tecnológicos, por parte de empresas de TI, pode ser considerado um risco, visto que operações de logística, e outras áreas, dependem destes elementos para funcionamento. A eventual indisponibilidade de um destes fornecedores, ou falta de recursos na cadeia tecnológica, pode afetar o fluxo de informações dentro da organização tão como suas operações.

### **Novos Entrantes:**

No cenário brasileiro atual, percebe-se um domínio estático de algumas empresas do setor. Ao adentrar o mercado de metais, necessidades legais e de alto investimento, dificultam a implantação de novos atuantes neste ramo. Também, a alta capacidade de fornecimento e de

produção, das empresas dominantes, facilita mudanças de preço e valores, o que dificulta ainda mais a capacidade de participação das novas organizações. Com isso, novas empresas no ramo se colocam como uma ameaça de baixo nível.

### Tendências do Mercado :

Analisando as evoluções econômicas e participação de empresas do setor siderúrgico, nos últimos anos, percebe-se uma drástica queda de investimentos no período de 2020, e uma alta recuperação no ano de 2021. Há um aumento na participação de empresas internacionais no cenário nacional, o que pode se colocar como um concorrente presente. Entretanto, a expansão de áreas e setores de investimentos, pela empresa, assim como foco na inovação, provocam uma fortificação da organização frente à novas mudanças, reforçando sua presença nacional.

### 1.3.2. Análise SWOT

#### Gerdau S.A.

Ambiente Interno	FORÇA	FRAQUEZA
	<ul style="list-style-type: none"> <li>-A Gerdau é uma empresa consolidada no mercado (120 anos).</li> <li>-Grande relevância: É a maior empresa de aço nacional.</li> <li>-Grande escala de atuação (quase toda América). Maior multinacional brasileira no ramo.</li> <li>-Forte autossuficiência produtiva.</li> <li>- Já segue políticas ESG.</li> </ul>	<ul style="list-style-type: none"> <li>-Empresa extremamente grande, o que pode gerar um descontrole administrativo central.</li> <li>- Administração com caráter familiar.</li> </ul>
Ambiente Externo	OPORTUNIDADE	AMEAÇA
	<ul style="list-style-type: none"> <li>-Automatização de processos com o uso da tecnologia.</li> <li>-Aplicação de novas tecnologias no modelo produtivo.</li> <li>-Constante demanda por conta do setor automobilístico e construção civil.</li> </ul>	<ul style="list-style-type: none"> <li>-Desaceleração da economia global no próximo ano.</li> <li>-Queda do preço do aço vendido no Brasil.</li> </ul>



### 1.3.3. Planejamento Geral da Solução

A solução proposta busca captar os dados de temperatura e umidade do ar da casa de vegetação, através da utilização de sensores e um microcontrolador, e armazená-los em um banco de dados de maneira automatizada para a análise de produtividade do plantio e notificar os funcionários caso ocorra a necessidade de uma interferência no viveiro.

Em virtude disso, temos como consequência a diminuição do intervalo entre medições, passando de uma hora para um minuto, além de monitorar 24 horas, logo, aumentando significativamente a quantidade de dados coletados, possibilitando uma análise profunda do comportamento das plantas, também, tornando essa coleta automática, sem interferência humana, evitando a ocorrência de erros e disponibilizando tempo para a realização de outras atividades pelos funcionários.

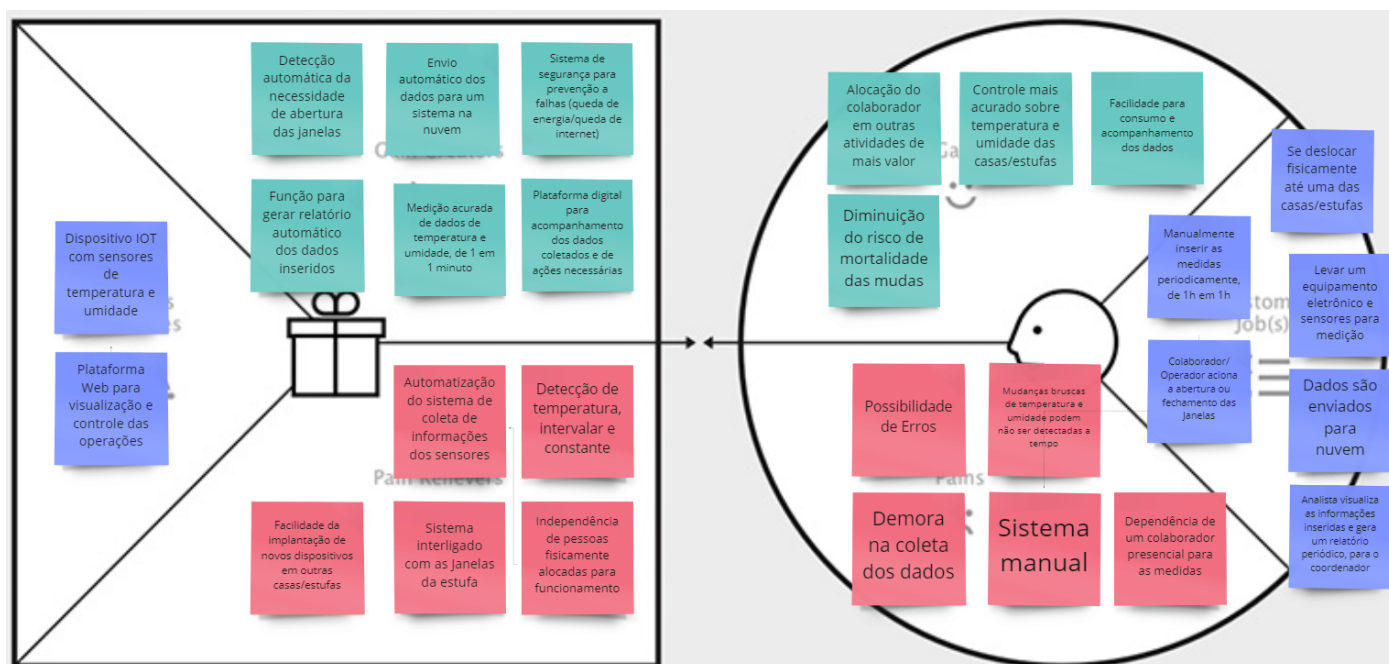
Através do TAP (documento de Termo de Abertura de Projeto), foram informados dados a respeito da maneira que a medição e registro dos dados são feitas atualmente, o funcionamento do viveiro, além de dados técnicos a respeito da infraestrutura desse, contudo, todos devem ser mantidos em sigilo.

Por meio da eventual utilização da solução, há como efeito a maximização de lucros da Gerdau, uma vez que a produtividade no plantio será aproveitada ao seu máximo, otimizando a produção de carvão vegetal que é utilizado como insumo na fabricação de aço.

A solução proposta pelo grupo traz dois principais benefícios para o parceiro: a redução na perda de mudas de eucalipto e a redução da necessidade de força humana no processo de tratamento das mudas de eucalipto. Nesse contexto, a partir da coleta dos dados de temperatura e umidade do ambiente, pode ser feita uma análise desses para que a produtividade de mudas aumente. Além disso, com a automatização do processo a partir da solução proposta, há a redução do envolvimento humano no processo, o que diminui erros comuns aos seres humanos e aumenta a velocidade de captação e verificação dos dados.

Para o projeto, os critérios de sucesso estabelecidos pelo grupo são os de disponibilidade, acurácia e produtividade. Nesse sentido, os critérios de disponibilidade e acurácia dizem respeito ao funcionamento e à qualidade do projeto, em que o primeiro expõe o período de tempo que o produto passa funcionando, assim, a medida utilizada é a de tempo. Já quanto à acurácia, tal medida será utilizada para comparar as temperaturas e umidades medidas pelo produto, e as reais do ambiente, sendo assim, medida uma porcentagem de acurácia do aparato. Além disso, para medir o impacto do produto na produtividade de mudas, será medido a quantidade de mudas com e sem o uso do aparato, o que iria expor a efetividade do produto no processo de cuidado das mudas.

### 1.3.4. Value Proposition Canvas



[Value Proposition Canvas Gerdau, Online Whiteboard for Visual Collaboration \(miro.com\)](#)

#### Customer Jobs:

- Se deslocar fisicamente até uma das casas/estufas
- Levar um equipamento eletrônico e sensores para medição
- Dados são enviados para nuvem
- Analista visualiza as informações inseridas e gera um relatório periódico, para o coordenador
- Colaborador/ Operador aciona a abertura ou fechamento das Janelas
- Manualmente inserir as medidas periodicamente, de 1h em 1h

#### Customer Pains:

- Possibilidade de Erros
- Mudanças bruscas de temperatura e umidade podem não ser detectadas a tempo
- Demora na coleta dos dados
- Sistema manual
- Dependência de um colaborador presencial para as medidas

### **Customer Pains Relievers:**

- Automatização do sistema de coleta de informações dos sensores
- Detecção de temperatura, intervalar e constante
- Independência de pessoas fisicamente alocadas para funcionamento
- Sistema interligado com as janelas da estufa
- Facilidade da implantação de novos dispositivos em outras casas/estufas

### **Products and Services:**

- Dispositivo IOT com sensores de temperatura e umidade
- Plataforma Web para visualização e controle das operações

### **Gain Creators:**

- Detecção automática da necessidade de abertura das janelas
- Envio automático dos dados para um sistema na nuvem
- Sistema de segurança para prevenção a falhas (queda de energia/queda de internet)
- Função para gerar relatório automático dos dados inseridos
- Medição acurada de dados de temperatura e umidade, de 1 em 1 minuto
- Plataforma digital para acompanhamento dos dados coletados e de ações necessárias

### **Gains:**

- Alocação do colaborador em outras atividades de mais valor
- Controle mais acurado sobre temperatura e umidade das casas/estufas
- Facilidade para consumo e acompanhamento dos dados
- Diminuição do risco de mortalidade das mudas

## 1.3.5. Matriz de Riscos

V.1:

Matriz de Risco											
Probabilidade		Riscos					Oportunidade				
Muito Alta	5						Aumentar a produtividade de outras áreas da empresa parceira				
Alta	4						Atender às necessidades do cliente.	Diminuir o risco de erro humano no processo de crescimento das mudas			
Médio	3				Poucos integrantes do grupo trabalharão (em todas as áreas do projeto)			Redução na perda de mudas	Aumentar a produção de mudas		
Baixa	2			Não entregar o projeto com todas as features definidas dentro do escopo	Aumento da exigência dos parceiros no que diz respeito aos entregáveis	Deixar passar algum problema que leve à perda de muitas mudas					
Muito Baixa	1					Não entregar o projeto no prazo definido					
		1	2	3	4	5	5	4	3	2	1
		Muito Baixo	Baixo	Médio	Alta	Muito Alta	Muito Alta	Alta	Médio	Baixo	Muito Baixo
Impacto											

V.2:

Matriz de Risco											
Probabilidade		Riscos					Oportunidade				
Muito Alta	5						Aumentar a produtividade de outras áreas da empresa parceira				
Alta	4						Atender às necessidades do cliente.	Diminuir o risco de erro humano no processo de crescimento das mudas			
Médio	3						Automatização de muitos processos da empresa parceira	Aumentar a produção de mudas			
Baixa	2			Não entregar todas as features definidas dentro do escopo	Aumento da exigência dos parceiros no que diz respeito aos entregáveis	Deixar passar algum problema que leve à perda de muitas mudas					
Muito Baixa	1			Poucos integrantes do grupo trabalharão	Não entregar o projeto no prazo definido						
		1	2	3	4	5	5	4	3	2	1
		Muito Baixo	Baixo	Médio	Alta	Muito Alta	Muito Alta	Alta	Médio	Baixo	Muito Baixo
Impacto											

[Matriz de riscos - Grupo 4 - Google Sheets](#)

## 1.4. Análise de Experiência do Usuário

### 1.4.1. Personas

#### Sabrina Cali



**Job Title**  
Operadora

**Age**  
25 to 34 years

**Highest Level of Education**  
Associate degree (e.g. AA, AS)

**Industry**  
Agriculture

**Organization Size**  
10,001+ employees

##### Tools They Need to Do Their Job

Sensores para medição da temperatura e umidade.

##### Job Responsibilities

Monitorar as condições das estufas para que sempre estejam em condições ideais para o crescimento dos eucaliptos.

##### Reports to

Supervisor Roger Chamas.

##### Goals or Objectives

Automatizar a medição de temperatura e umidade de minuto em minuto para que os eucaliptos se desenvolvam em melhores condições.

##### Biggest Challenges

Muito trabalho manual que poderia ser feito automaticamente.  
Trabalho árduo.

##### Hobbies

Assistir National Geographic;  
Yoga;  
Acampar e fazer trilha;  
Mãe de planta.

#### Roger Chamas



**Job Title**  
Supervisor

**Age**  
35 to 44 years

**Highest Level of Education**  
Bachelor's degree (e.g. BA, BS)

##### Social Networks



**Industry**  
Agriculture

**Organization Size**  
10,001+ employees

##### Job Responsibilities

Monitorar a operação do crescimento das mudas nas casas de vegetação.

##### Goals or Objectives

Aumentar a produtividade dos viveiros da Gerdau. Melhor distribuição dos seus operadores em outras tarefas

##### Biggest Challenges

- Os operadores poderiam gastar o tempo deles com atividades que não podem ser automatizadas.

##### Hobbies

Pesca  
Jardinagem  
Cozinhar

## Kassius



**Job Title**  
Analista

**Age**  
35 to 44 years

**Highest Level of Education**  
Master's degree (e.g. MA, MS,

**Social Networks**



**Industry**  
Agriculture

**Organization Size**  
10,001+ employees

### Tools They Need to Do Their Job

Software com os dados das casas de vegetações.

### Job Responsibilities

Faz relatórios a partir dos dados coletados pelo operador, esses relatórios são entregues ao coordenador que toma decisões.

### Goals or Objectives

Obter mais dados para uma análise mais profunda.

### Reports to

Supervisor Roger Chamas.

### Biggest Challenges

- Não tem dados o suficiente para fazer uma análise eficaz.

## 1.4.2. Jornadas do Usuário e/ou Storyboard








**Sabrina Cali, Operador**

**Cenário:** Sabrina quer ter mais praticidade e eficiência na estufa em que trabalha

### Expectativas

- Ter um sistema de análise do ambiente confiável (umidade e temperatura);
- Receber alertas na necessidade de abertura ou fechamento das janelas;
- Não precisar mais escrever os dados que serão enviados para os analistas;

Conferindo condições do ambiente	Adaptando janelas	Preparando os dados	Aplicando o hardware	Conferindo resultados
<p>1. Sabrina vai ao centro da estufa e leva os seus equipamentos tradicionais de medição de temperatura e umidade;</p> <p>2. Após conferir os equipamentos, se prepara para adaptar a estufa diante do que foi lido;</p> <p>'Um tanto que tedioso ter ir até a estufa apenas para conferir as condições do ambiente.'</p> 	<p>1. Conforme a leitura das condições do ambiente, Sabrina adapta as janelas da estufa;</p> <p>2. Após ajustar as janelas pela sua percepção humana, ele já se prepara para a próxima fase;</p> <p>'Ter que fechar as janelas por conta própria é trabalhoso demais, podia só ter um fácil botão ou ser automático mesmo. Além disso, não tenho certeza se a forma como eu adapto as janelas realmente é a melhor'</p> 	<p>1. Após todo esse processo, Sabrina registra por ele mesmo as condições ambientais e adaptações feitas na estufa naquele curto período de tempo ;</p> <p>2. Após concluir o registro, envia para o formulário que os analistas tem acesso;</p> <p>'Muito tedioso e maçante ter que anotar esses dados. Só piora o fato de ter que fazer a cada hora'</p> 	<p>1. Aplicando o novo hardware, a temperatura e umidade são conferidas automaticamente;</p> <p>2. Após a medição, uma ação sugerida para as janelas será informada no celular de Sabrina (informando também as condições climáticas), sendo necessário apenas apertar um botão no seu celular para as janelas mudarem de posição;</p> <p>3. Horários, ações e condições climáticas são enviados automaticamente para um banco de dados que os analistas tem acesso;</p> <p>'Que prático tudo ficou agora!'</p> 	<p>1. Depois de um período usando o hardware, Sabrina recebe uma devolutiva dos analistas dos resultados obtidos pelas mudas naquele espaço de tempo;</p> <p>2. Assim, Sabrina consegue ter um feedback do seu trabalho e comparar por ele mesmo os resultados de antes e depois do uso do hardware;</p> <p>'De fato, o uso do hardware facilita meu trabalho e ainda melhora os resultados. Fico muito satisfeito!'</p> 

### Oportunidades

- Prevenção em caso de queda de energia ou Internet.
- Criar uma interface fácil e amigável para o controle de Sabrina.

[Jornada de Usuário \(Mod.4\), Online Whiteboard for Visual Collaboration \(miro.com\)](#)








### Roges Chamas, Supervisor

**Cenário:** Roger quer aumentar a produtividade da casa de vegetação em que trabalha.

### Expectativas

- Automatizar ou deixar mais prático antigas atividades desenvolvidas na estufa;
- Conseguir melhores resultados em relação às mudas;
- Ter a possibilidade de distribuir seus operadores em outras tarefas;

Planejamento	Supervisão prática	Supervisão de resultados	Aplicando o hardware	Nova supervisão de resultados
<p>1. Roger se reúne com os operadores da estufa para fins organizativos do trabalho que será desempenhado;</p> <p>2. Após alinhar o trabalho com os operadores da estufa, Roger se prepara para supervisionar o que será desenvolvido ao longo do dia;</p> <p>'Muito bom poder falar diretamente com os operadores da estufa, um alinhamento é sempre necessário, é o coração do meu cargo '</p> 	<p>1.As atividades práticas começam, Roger supervisiona se cada procedimento está sendo feito com a qualidade necessária ;</p> <p>'Ao supervisionar, percebo como algumas atividades poderiam ser automatizadas, assim, o operador poderia desenvolver outras tarefas estritamente humanas'</p> 	<p>1.Após todo procedimento prático, Roger supervisiona como os dados estão sendo enviados para os analistas e a qualidade deles;</p> <p>2.Passando algum tempo, Roger e sua equipe recebem uma devolutiva dos analistas, tendo uma estatística do seu trabalho;</p> <p>'Vejo mais uma vez como parte do trabalho podia ser automatizado. Além disso, fica claro como o resultado final fica prejudicado por conta de falhas humanas minhas e dos operadores'</p> 	<p>1.Aplicando o novo hardware, procedimentos, antes manuais, são realizados sem assistência humana;</p> <p>2.Nessa aplicação, a coleta e envio de dados também se torna automática;</p> <p>'Finalmente várias funções estão sendo automatizadas com o uso desse hardware, agora os operadores conseguem se focar melhor em outras atividades'</p> 	<p>1. Aplicando o hardware, Roger recebe uma nova devolutiva dos analistas dos resultados obtidos;</p> <p>2.Assim, Roger consegue comparar os resultados de antes e depois do uso do hardware;</p> <p>'Fico muito contente! De fato, o uso do hardware aumenta a quantidade e qualidade do trabalho que fazemos aqui na casa de vegetação.'</p> 

### Oportunidades

- Dashboard para o supervisor poder acompanhar melhor a operação.

[https://miro.com/app/board/uXjVPL3SEz0=?share\\_link\\_id=719186113350](https://miro.com/app/board/uXjVPL3SEz0=?share_link_id=719186113350)









**Kassius Bugalho, Analista**

**Cenário:** Kassius quer produzir análises melhores da estufa que "observa".

### Expectativas

- Ter acesso as medições feitas de 1 em 1 minuto.

Aguardo do dados	Verificação dos dados	Análise dos dados	Aplicando o hardware
<p>1. Kassius naturalmente aguarda o recebimento dos dados, contudo, por ser coletado e enviado por pessoas, sempre existe uma variação de tempo nessas entregas;</p> <p>'Seria bom ter uma estrita constância no envio de dados, como analista prezo muito por essa estabilidade '</p> 	<p>1. Antes de realmente analisar o que os dados dizem, Kassius deve verificar se não houve algum erro humano na hora de coleta ou envio dos dados (colocar um zero a mais na temperatura térmica por exemplo) ;</p> <p>'Terrível! Acabo gastando muito tempo verificando os dados e ainda tenho o risco de ter meu trabalho prejudicado por conta de um dado errôneo'</p> 	<p>1. Após o aguardo e verificação dos dados, Kassius finalmente consegue fazer a análise, contudo, sua análises são baseadas numa quantidade pequena de dados;</p> <p>2. Traduzindo o que os dados querem dizer, Kassius formaliza isso em documento e envia para as outras equipes da estufa;</p> <p>'Após todos esses processos finalmente posso analisar os dados, contudo, ainda tenho receio da quantidade de dados afetar o resultado final'</p> 	<p>1. Aplicando o novo hardware o horário do envio de dados se torna muito mais constante.</p> <p>2. A precisão e qualidade dos dados também melhora de forma expressiva;</p> <p>3. A análise de dados se torna mais fácil e exata;</p> <p>'Agora sim! Posso me focar completamente na análise de dados! Acredito que a qualidade das análises vai aumentar bastante também! '</p> 

### Oportunidades

- Dashboard para o analista.

[https://miro.com/app/board/uXjVPL3SE\\_o=/?share\\_link\\_id=428864606093](https://miro.com/app/board/uXjVPL3SE_o=/?share_link_id=428864606093)

### 1.4.3. User Stories

Épico	User Story
Eu, como operadora, quero que a temperatura e umidade da plantação sejam medidas automaticamente para verificar se as condições estão favoráveis.	Eu, como operadora, quero receber notificações com sugestões das mudanças que devem ser feitas no viveiro para tornar o ambiente favorável para as plantas.
	Eu, como operadora, quero que as informações continuem sendo coletadas quando houver queda de energia para que as informações não sejam perdidas.
	Eu, como operador, quero receber um aviso sonoro de que algo deve ser modificado no viveiro para que eu ainda saiba quando interferir mesmo que não haja energia.
Eu, como supervisor, quero ter acesso as condições do viveiro em tempo real.	Eu, como supervisor, quero ter um dashboard atualizado em tempo real para ter informações mais próximas da realidade possível.
	Eu, como supervisor, quero receber notificações com sugestões das mudanças que devem ser feitas no viveiro para tornar o ambiente favorável para as plantas.
	Eu, como supervisor, quero que o aparelho tenha um indicativo de seu status para que eu saiba quando está funcionando.
Eu, como analista, quero ter acesso aos dados do viveiro para gerar relatórios para o meu coordenador.	Eu, como analista, quero ter um dashboard atualizado em tempo real para ter informações mais próximas da realidade possível.

### 1.4.4. Protótipo de interface com o usuário

Separamos nosso wireframe em três páginas: na primeira página a versão mobile, na segunda a versão web e na última o hardware.

[Projeto 4 – Figma](#)

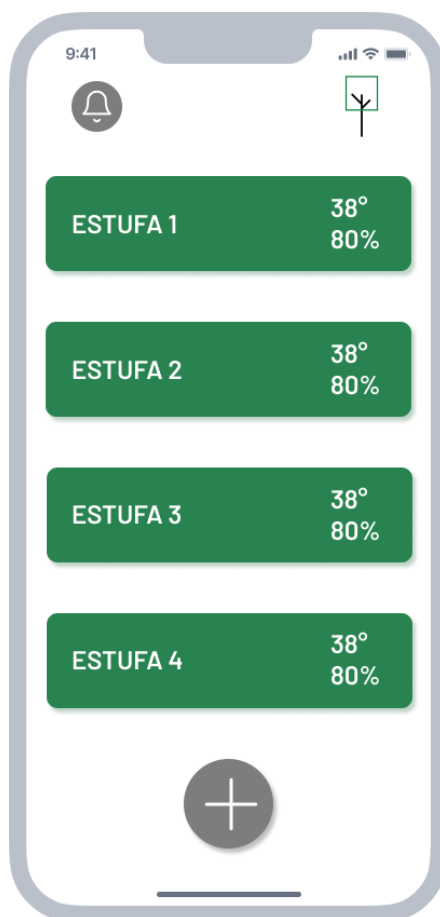
#### 1.4.4.1 Telas

Inicialmente, para a elaboração do protótipo inicial da aplicação, foram desenvolvidas as seguintes telas:

- Tela de seleção, listagem adição e resumo de temperatura e umidade de estufas presentes no sistema
- Tela de alertas/ações pendentes
- Tela intermediária para acompanhamento do processo de abertura/fechamento das janelas de uma estufa
- Tela de seleção de ferramentas, abrangendo desde mudanças em configurações até processos de extração de dados
- Tela de configurações de rede Wifi
- Tela de confirmação de configuração do WIFI
- Tela de edição de múltiplas configurações do dispositivo
- Tela de extração manual de medições de temperatura e umidade

#### 1.4.4.2 Descrição detalhada das telas Web/Mobile

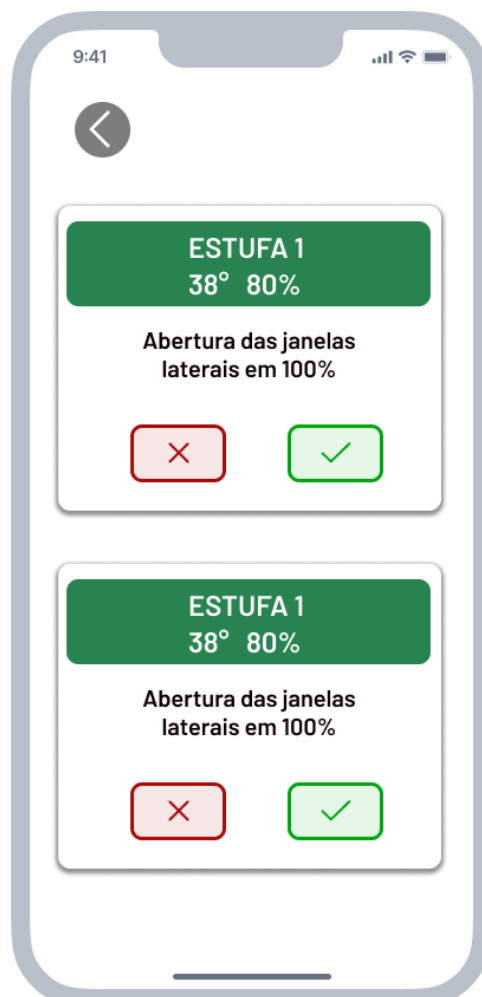
Tela de seleção, listagem adição e resumo de temperatura e umidade de estufas presentes no sistema



O usuário responsável pelo acompanhamento das estufas, tem, por meio dessa tela, a possibilidade de cadastro, edição e acompanhamento geral das informações de umidade e temperatura de cada estação de medida. Além disso, há, no formato de um sino, um indicador de alertas de urgência, para o entendimento de possíveis violações de regras de temperatura, umidade, e outros problemas.

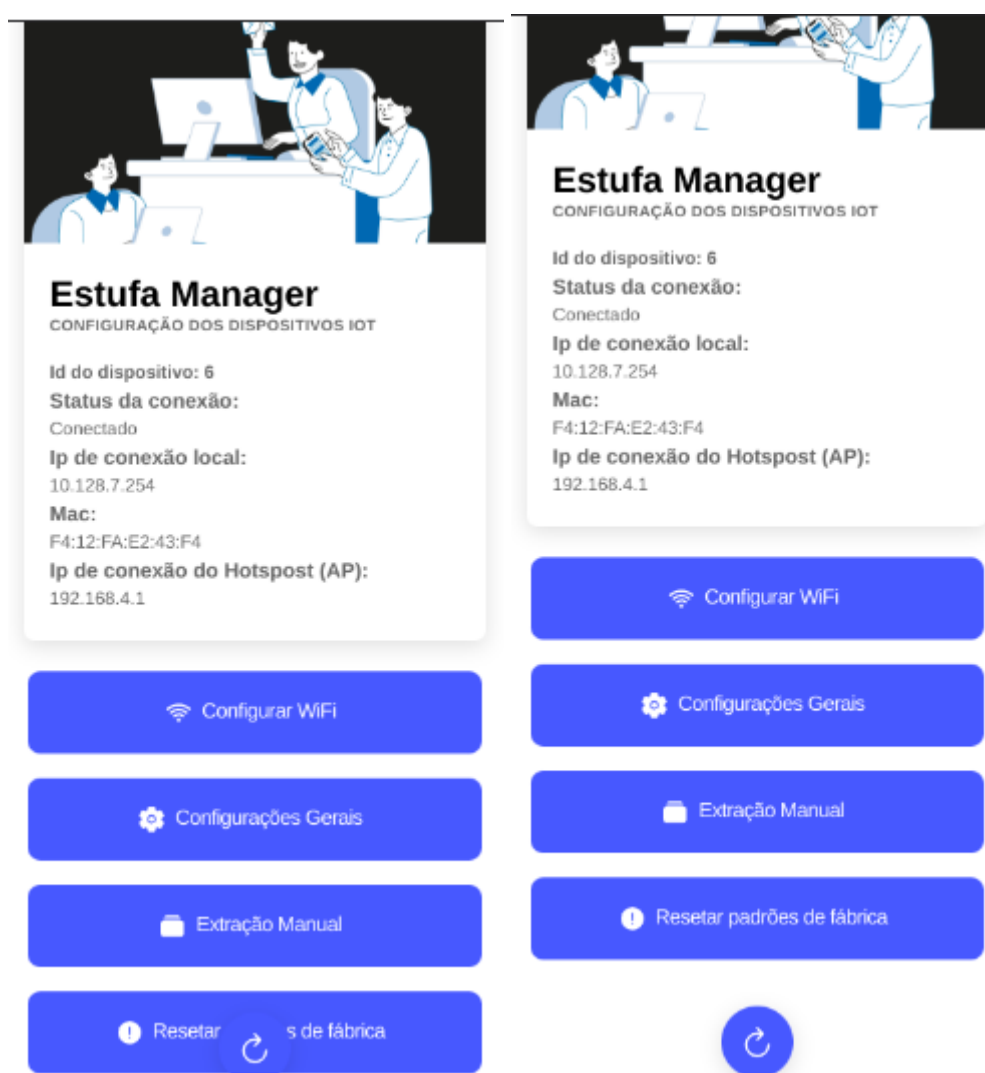
### Tela de alertas/ações pendentes

Durante as medições de temperatura e umidade, de uma ou mais estufas, caso haja a violação de limites pré-determinados nestes escopo, alertas serão gerados. Estes alertas, assim como ações de mitigação estarão disponíveis nesta tela, pendentes até a normalização das medidas ou execução de uma ação.



### 1.4.4.3 Descrição detalhada das telas do dispositivo ESP32

Tela de seleção de ferramentas, abrangendo desde mudanças em configurações até o processo de extração de dados



Levando em consideração o processo de instalação e uso do dispositivo desenvolvido, existe a necessidade da inserção de configurações iniciais, modificação destas configurações em um período futuro, testes, e também atualizações pontuais. Com isso, cria-se uma tela interna, capacitada de diversas opções como: configuração de wifi, configurações gerais, extração manual e restauração dos padrões de fábrica.

### Tela de configuração de rede Wifi

[< Configurações de WiFi](#)




[✓ Salvar Informações](#)

Um dos requisitos, para a transferência de dados entre o dispositivo e um servidor remoto, é a conectividade. Selecionando o botão de “Configurar WiFi”, o usuário deve inserir o nome e a senha da rede para que seja feita a conexão.

## Tela de edição de múltiplas configurações do dispositivo

< Configurações Gerais



Servidor remoto (API):  
https://keepgrowing-api.fly.dev

Servidor Ntp Principal:  
pool.ntp.org

Servidor Ntp Alternativo:  
time.nist.gov

Temperatura Mínima (°C):  
20


Temperatura Máxima (°C):  
30

Temperatura Alta (°C):  
31

Temperatura Extremamente Alta (°C):  
32

Umidade Baixa (%):  
66.5

< Configurações Gerais



Servidor remoto (API):  
https://keepgrowing-api.fly.dev

Servidor Ntp Principal:  
pool.ntp.org

Servidor Ntp Alternativo:  
time.nist.gov

Temperatura Mínima (°C):  
20

Temperatura Máxima (°C):  
30

Temperatura Alta (°C):  
31

Temperatura Extremamente Alta (°C):  
32

Umidade Baixa (%):  
66.5

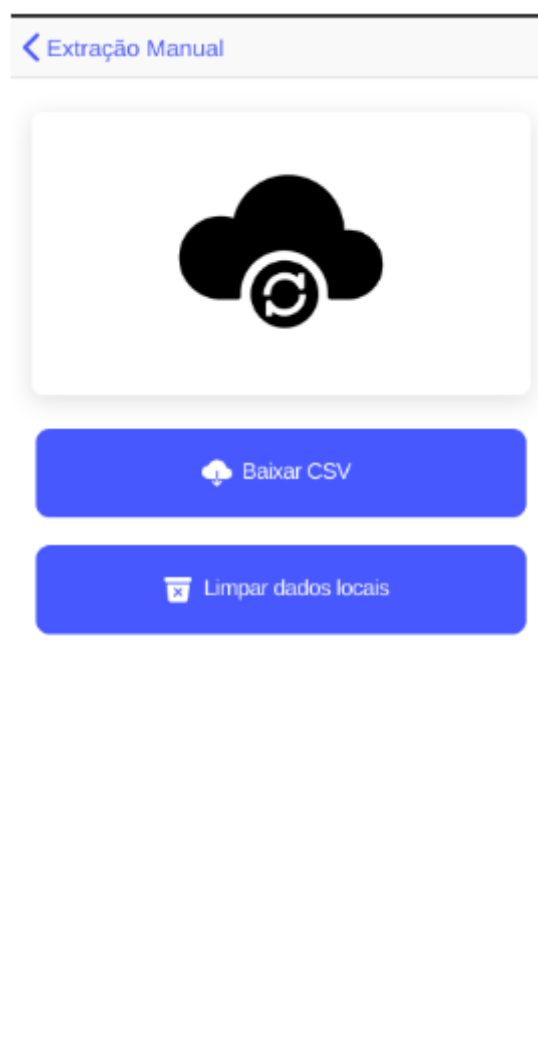
Umidade Alta (%):  
90.25

✓ Salvar Configurações

Para a integração do dispositivo com um servidor remoto, informações como endereço do servidor, porta, e outros são de suma importância. Devido à característica efêmera de muitas destas configurações, há a necessidade de um meio pelo qual o usuário possa atualizar ou editar as informações. Esta tela auxilia neste processo, permitindo de forma intuitiva, a inserção, criação e remoção de variáveis do ambiente, posteriormente utilizadas pelo dispositivo.



## Tela de extração manual de medições de temperatura e umidade

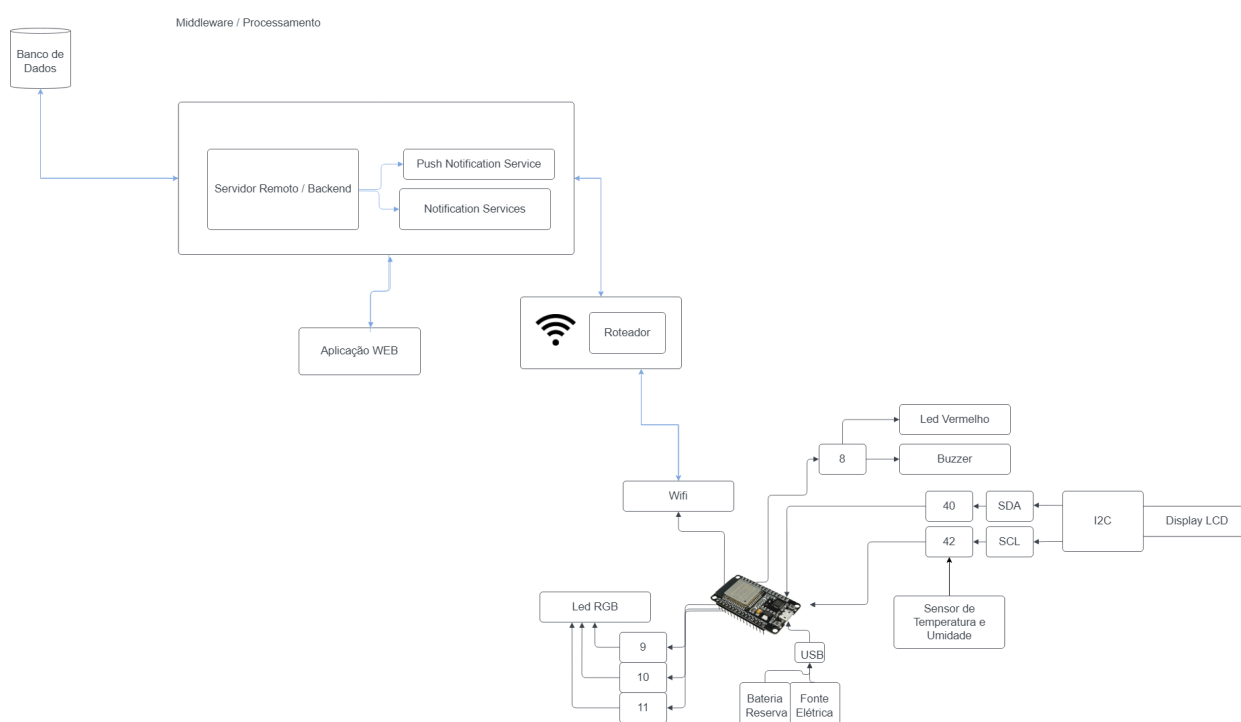


A presença das estufas de eucaliptos, em uma região rural, expõe a solução a diversos riscos como indisponibilidade da rede de internet, e queda de energia. Como a comunicação e transmissão dos dados capturados pelo dispositivo é feita por meio da internet, estes riscos podem levar à inoperabilidade do equipamento. Para contornar esse problema, usam-se métodos de extração manual dos dados, os quais devem estar alocados no buffer do ESP32.

Esta tela permite acesso aos registros de dados de medição.  
Dessa forma, o usuário pode exportar os dados, para uso posterior, sem a necessidade de conectividade com a internet.

## 2. Arquitetura da solução

### 2.1. Arquitetura versão 1



Componente	Descrição da função/características/requisitos
Sensor de	Medir temperatura do ambiente de minuto em minuto.

temperatura	
Sensor de Umidade	Medir umidade relativa do ar do ambiente de minuto em minuto.
ESP32	Armazenamento de dados de configuração de rede. Conexão bluetooth com dispositivos móveis para primeira conexão na rede wi-fi. Conexão com rede wifi para upload de dados gerados pelo sensor. <i>Buffering</i> dos dados gerados minuto a minuto pelos sensores em caso de falha na conexão com a rede.
LED verde	Indica que está conectado ao wifi.
LED vermelha	Indica que está desconectado do wifi.
Display	Mostra mensagens simples de erro e código para conexão bluetooth que será usada para configuração inicial da rede wifi.

## 2.1.1 Camada de Percepção

A camada de percepção, composta pelo controlador central, sensores e atuadores correspondentes, é responsável pela captação e leitura dos dados provenientes dos sensores, assim como acionamento dos atuadores de acordo com regras previamente estabelecidas.

O display serve como auxiliar na comunicação e setup dos dispositivos, tanto como configuração de wifi e possível conexão bluetooth se existente, sendo desligado após um tempo pré-determinado, dessa forma, estendendo a vida útil do componente.

## 2.1.2 Camada de Rede

Para fazer a transmissão dos dados capturados pela unidade controladora, é necessário estabelecer métodos de comunicação para com o servidor responsável. No caso do ESP 32, utiliza-se o módulo WIFI acoplado, para conectá-lo à roteadores de internet, presentes no local, para que estes então, possam redirecionar e permitir o acesso do dispositivo à internet. Além disso, para configuração e *setup* dos dispositivos, pode-se utilizar o modo Hotspot do módulo de wifi, para acesso e comunicação com interfaces internas.

## 2.1.3 Camada Middleware

Nesta camada, são estabelecidos meios para o recebimento e agregação dos dados aferidos pelos dispositivos remotos. As informações são recebidas por rotas de um servidor,

para então serem guardadas em um banco de dados especificado ou outro meio de persistência de dados. Conecta-se também, nesta seção, serviços externos de suma importância para a aplicação, como Push Notification Services, e serviços de status.

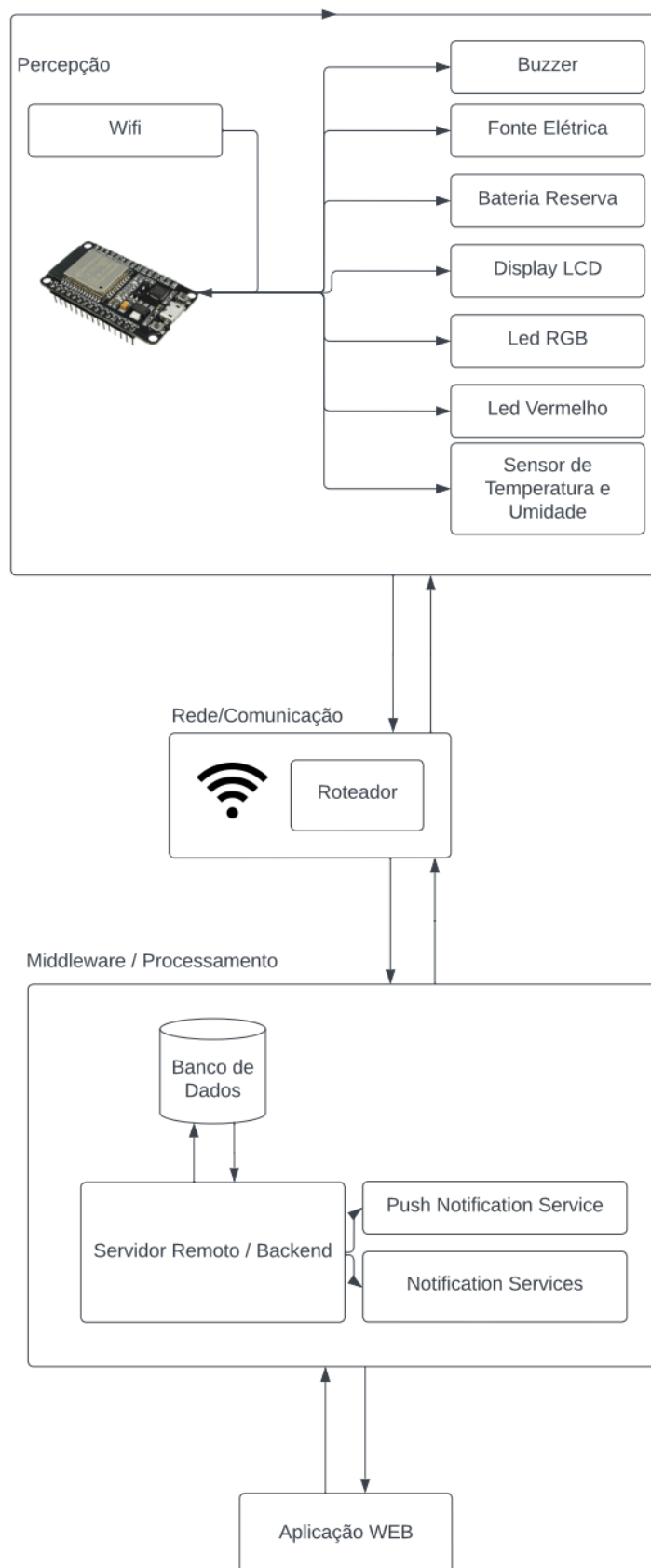
## 2.1.4 Camada de Processamento

Na camada de processamento, os dados previamente recebidos são processados a partir de regras de funcionamento. Limites de temperatura e definições de métricas para medição, são aplicadas às informações coletadas, para então gerar sinais de ação, tais como notificações push ou notificações de status do sistema e dispositivos. Ademais, sintetiza-se estas informações, de acordo com períodos ou intervalos de tempo, para a possível geração de gráficos, buscando o entendimento dos dados.

## 2.1.5 Aplicação Web

Nesta última camada, é definido o meio de acesso, visualização e controle de ações, baseados nos dados produzidos pelos dispositivos IOT. Compreende-se como aplicação Web, uma plataforma online, que permite à múltiplos usuários, a visualização dos dados de temperatura e umidade, por estufa, dentro de um período determinado por este usuário, em forma de gráficos. Ainda, esta camada permite a execução e confirmação de ações que afetam diretamente ou indiretamente as estufas, como por exemplo, a abertura ou fechamento das janelas de uma estufa. Por fim, os serviços de notificação push e de status, presentes no servidor, utilizam a aplicação para avisar riscos de alta temperatura, e indicações de abertura de janelas,desse modo, ajudando na prevenção de perdas de mudas.

## 2.2. Arquitetura versão 2



Componente / Conexão	Descrição da função	Tipo:  entrada / saída / atuador / alimentação
ESP32 / S3	Armazenamento de dados de configuração de rede. Conexão bluetooth com dispositivos móveis para primeira conexão na rede wi-fi. Conexão com rede wifi para upload de dados gerados pelo sensor. <i>Buffering</i> dos dados gerados minuto a minuto pelos sensores em caso de falha na conexão com a rede.	
Led RGB	O led RGB é um conjunto de três LEDs encapsulados das cores: vermelho, verde e azul. Será utilizado para mostrar o status das mudas, vermelho para estado crítico e verde para estado normal e amarelo para estado de alerta.	Saída
Display LCD 16x2 1602A	LCD PANEL 16 caracteres por duas linhas (16x2). Esse display será utilizado para exibir informações importantes, como temperatura e umidade. Também, poderá auxiliar no entendimento de possíveis falhas, ao exibir códigos de erro padronizados e status de conectividade wifi.	Saída
Buzzer	Durante o processo de medição de temperatura, devido à riscos locais na infraestrutura da região, há grandes chances de indisponibilidades na rede e/ou energia ocorrerem. Como elemento de segurança, caso os métodos principais de alerta de temperatura e umidade, falharem, o buzzer entra como um dispositivo sonoro, indicando o status emergencial de uma estufa.	Saída
Led vermelho	O Led vermelho tem a função de sinalizar quando a conexão de wifi falhar.	Saída

	Medir a temperatura e umidade relativa do ar a cada minuto.	
Sensor de Temperatura e Umidade (AHT10)		Entrada
Fonte Elétrica	Fornece energia para o ESP32.	Alimentação
Bateria Reserva	Fornece energia para o ESP32, em caso de queda de energia.	Alimentação

## 2.2.1 Camada de Percepção

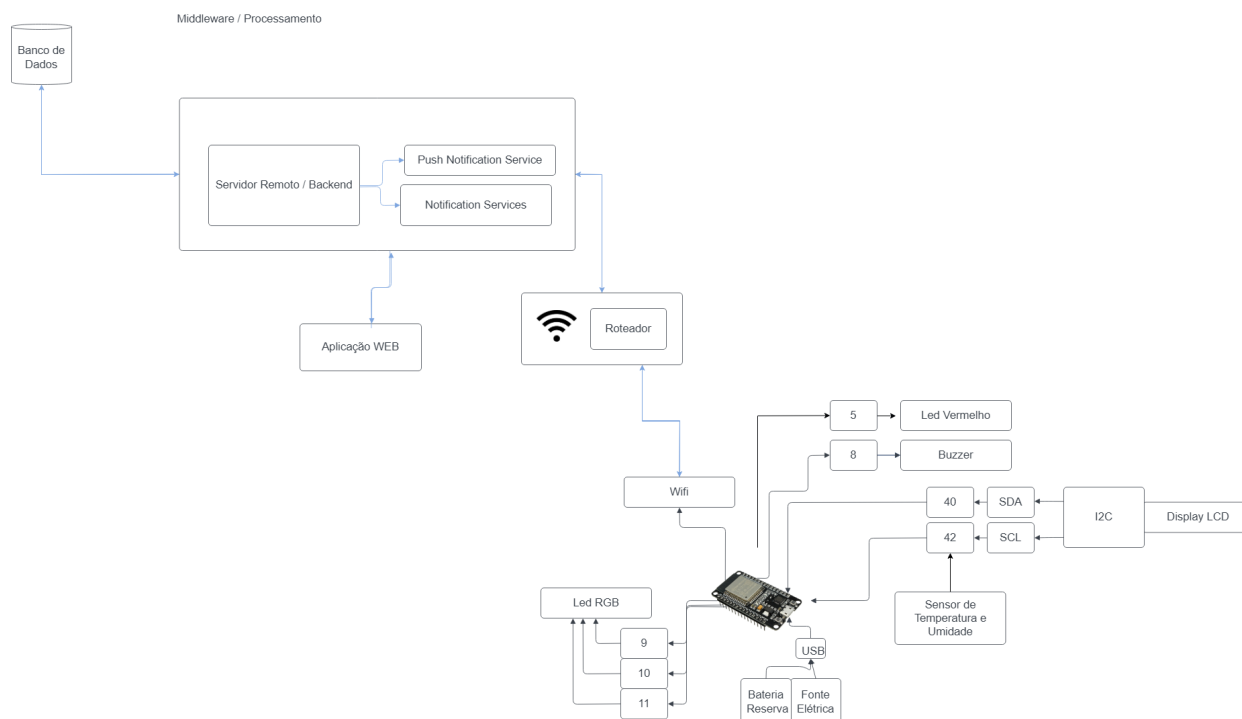
Na camada de percepção, as mudanças feitas se referem a adição de um led RGB para a indicação das condições climáticas do ambiente, antigamente, a ideia era a utilização de um led verde e um vermelho, mas a troca por um único led se torna mais objetiva e intuitiva para o usuário. Além disso, o acréscimo do buzzer, que será tocado continuamente em situações de queda de energia e/ou internet quando as condições não são favoráveis até que elas voltem a ser, sendo uma forma de indicar aos operadores que mudanças devem ser feitas mesmo quando não é possível o acesso pelo aplicativo. Também, a junção dos sensores de temperatura e umidade num único componente, uma vez que o modelo escolhido como o melhor para o projeto funciona de tal maneira.

## 2.3. Arquitetura versão 3

Na terceira sprint, iniciamos o desenvolvimento da integração entre o dispositivo físico e as plataformas do back-end e front-end da aplicação. Com isso, diversas mudanças foram feitas, de forma a executar a comunicação dos dados, assim como atualização e produção de alertas.

As seguintes implementações foram feitas:

- Conexão automática com rede wifi pré-determinada no dispositivo
- Criação de rotas para recebimento das informações do dispositivo
- Instanciação do banco de dados e modelos relacionais



No diagrama acima, as setas azuis representam as conexões sem fio via wifi e as setas pretas conexões com fio. As portas 9, 10, 11 e 8 estão sendo usadas para output. As portas 40 e 42 estão sendo usadas para comunicação do I2C com o display e o sensor de temperatura e umidade.

### 2.3.1 Conexão Automática com rede Wifi

Para conectar a placa na rede, com a internet, foi utilizada a biblioteca "Wifi.h". Considerando também, que a comunicação primordial é centrada no envio de dados do dispositivo para um servidor externo, utiliza-se a placa de rede wifi no modo cliente. Após a importação da biblioteca, são definidas em duas variáveis, ssid e password, as informações de acesso da rede Wifi alvo. Todo este processo é centrado na função setup, do ESP32, onde ele permanece, até que uma conexão seja estabelecida com sucesso.



### 2.3.2 Rotas para recebimento das informações do dispositivo e comunicação HTTP

Na etapa loop, do ESP32, informações de temperatura e umidade são capturadas pelos sensores ligados à placa, e logo após são enviados para um servidor previamente especificado. Este destino é definido em uma variável chamada "serverName". Para se comunicar com o servidor, é feita a importação da biblioteca "HTTPClient.h". Logo após, esta classe é instanciada, passando o endereço destino, e o cliente wifi. Assim, além de a placa conseguir se comunicar com a internet, é efetuado o envio dos dados adquiridos, para um computador remoto.

As rotas do back-end definidas são compreendidas em:

- Medidas: Conjunto de rotas para o recebimento e listagem de medidas de temperatura e umidade
- Notificação: Conjunto de rotas para a listagem, gatilho e remoção de alertas de temperatura e umidade, a partir de regras previamente estabelecidas
- Dispositivos: Conjunto de rotas para listagem e habilitação de dispositivo ESP32 - estufas, no banco de dados

Dessas, o dispositivo utiliza diretamente a rota medidas, para adição das informações coletadas. Logo após, será implementada a rota dispositivos, para o cadastro automático de cada ESP32 no banco de dados.

### 2.3.3 Instanciação do banco de dados e modelos relacionais

Para a persistência dos dados, recebidos pela placa, foi estabelecido um sistema de banco de dados remoto, em PostgreSQL. Para cada rota da aplicação, foi constituída uma tabela, com uma série de colunas dependentes. Considera-se presente as 3 tabelas:

- Medidas: escopo de responsabilidade fechado aos dados de temperatura e umidade da solução.
- Notificação: escopo de responsabilidade fechado à aplicação das regras de negócio, em cima dos dados de medida de um dispositivo, para a geração de alertas.
- Dispositivo: escopo de responsabilidade fechado ao cadastro e modificação de um dispositivo ESP32, também referenciado como estufa.

Entende-se também, que a persistência de dados neste meio, é feito por meio do back-end, pelo qual a placa se comunica e fornece os dados necessários.

Posicione aqui a evolução dos seus diagramas, aprimorando a versão inicial dos blocos e incluindo as soluções de interação com módulos externos (por exemplo, sistema de posicionamento). O diagrama e a tabela devem:

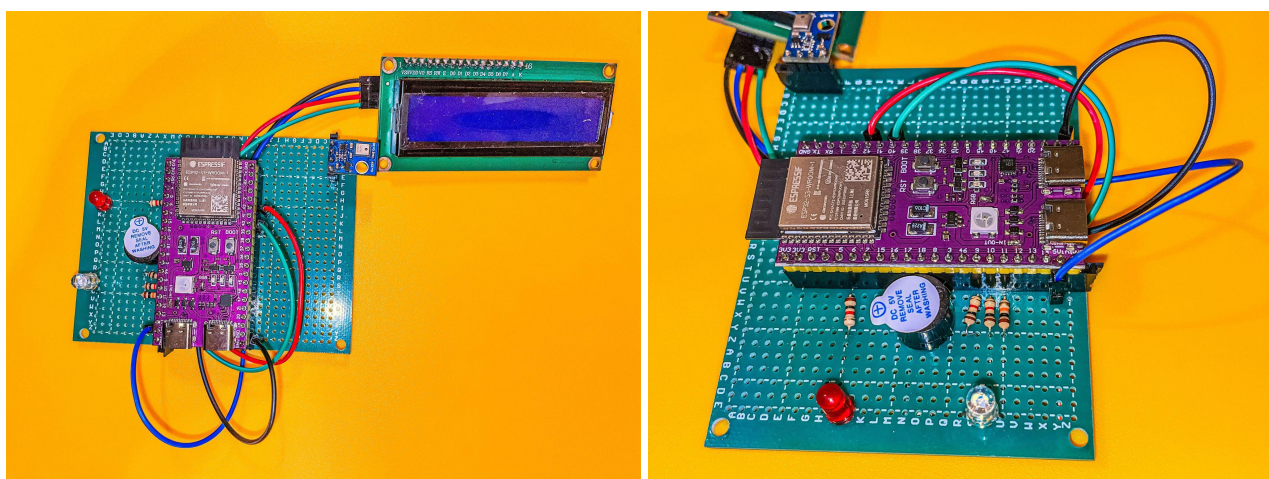
1. Além do já incluído nas versões anteriores, mostrar a interação indireta (wifi) entre os elementos externos e o seu funcionamento

Componente / Conexão	Descrição da função	Tipo: entrada / saída / atuador / alimentação
ESP32 / S3	Armazenamento de dados de configuração de rede. Conexão bluetooth com dispositivos móveis para primeira conexão na rede wi-fi. Conexão com rede wifi para upload de dados gerados pelo sensor. <i>Buffering</i> dos dados gerados minuto a minuto pelos sensores em caso de falha na conexão com a rede.	—
Led RGB	O led RGB é um conjunto de três LEDs encapsulados das cores: vermelho, verde e azul. Será utilizado para mostrar o status da temperatura e umidade, vermelho para estado crítico e verde para estado normal e amarelo para estado de alerta. Conforme o valor lido pelo sensor, são acionadas as portas de cada cor para que a led acenda na cor necessária de acordo com os intervalos passados pelo usuário.	Saída (portas 9, 10 e 11)
Display LCD 16x2 1602A	LCD PANEL 16 caracteres por duas linhas (16x2). Esse display será utilizado para exibir informações importantes, como temperatura e umidade. Também, poderá auxiliar no entendimento de possíveis falhas, ao exibir códigos de erro padronizados e status de conectividade wifi. Conectado na ESP32 utilizando o I2C.	Saída(SDA 40, SCL 42)
Buzzer	Durante o processo de medição de temperatura, devido à riscos locais na infraestrutura da região, há grandes chances de indisponibilidades na rede	Saída(8)

	e/ou energia ocorrerem. Como elemento de segurança, caso os métodos principais de alerta de temperatura e umidade falham, o buzzer entra como um dispositivo sonoro, indicando o status emergencial de uma estufa.	
Led vermelho	O Led vermelho tem a função de sinalizar quando a conexão de wifi falhar.	Saída(5)
Sensor de Temperatura e Umidade (AHT10)	Medir a temperatura e umidade relativa do ar a cada minuto.	Entrada(42)
Fonte Elétrica	Fornece energia para o ESP32.	Alimentação(USB)
Bateria Reserva	Fornece energia para o ESP32, em caso de queda de energia.	Alimentação(USB)

## OBS:

O produto pode ser feito utilizando uma placa com circuito impresso, com as soldagens feitas da seguinte forma:



## 2.4. Fluxo de erros

O fluxo de erros tem como objetivo esclarecer quais os caminhos e possíveis erros que podem ocorrer dentro da arquitetura da solução, além de auxiliar como eles serão exibidos.

Pensando nisso, desenvolvemos uma nomenclatura para cada erro [ <ESCOPO> <ID> <GRAVIDADE> ].

Definição de Códigos:

Definição de Escopo:

H - Erros relacionados ao Hardware

S - Erros relacionados ao software

C - Erros relacionados a conectividade do dispositivo

ID do código de erro:

- 00
- 01
- 02
- 03

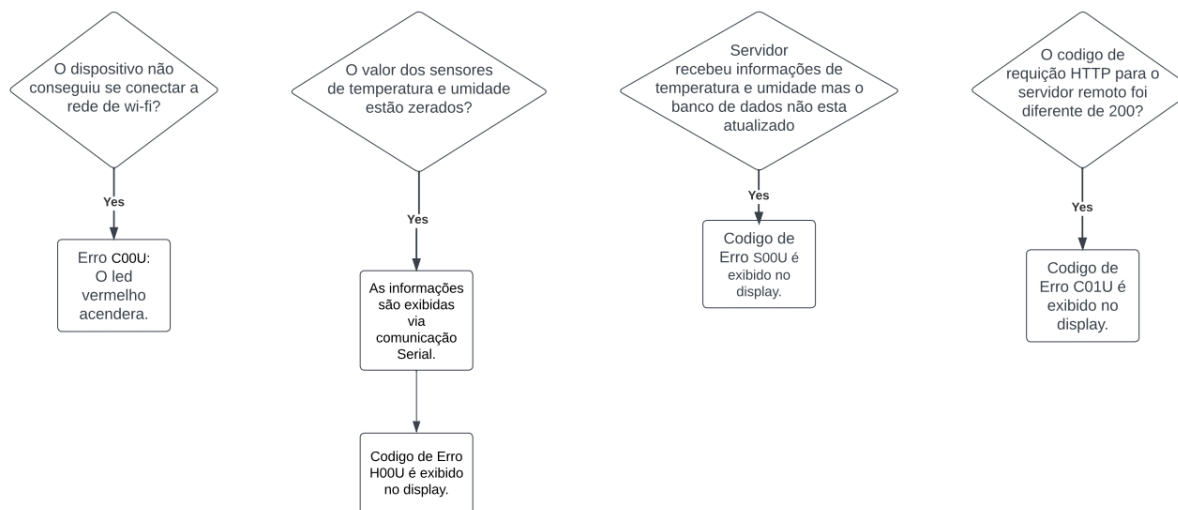
Definição de Gravidade:

- L - Low
- M - Medium
- U - Urgent

CÓDIGO COMPLETO	ESCOPO	ID	GRAVIDADE	DESCRIÇÃO
H00U	H	00	U	Os sensores não estão capturando os dados de temperatura e umidade.
S00U	S	00	U	O servidor não consegue se comunicar com o banco de dados.
C00U	C	00	U	O dispositivo não conseguiu obter uma conexão com o wifi.
C01U	C	01	U	O dispositivo não conseguiu se

				comunicar com o servidor remoto.

Tabela de definição dos erros



Fluxograma dos erros

## 3. Situações de uso

### 3.1. Entradas e Saídas por Bloco

#	bloco	componente e de	leitura da	componente de saída	leitura da saída	Descrição
---	-------	-----------------	------------	---------------------	------------------	-----------

		entrada	entrada			
1	Medidor de temperatura local	Medidor de temperatura AHT10	$< 45^{\circ}\text{C}$	Display LCD 16x2 1602A	alteração dos caracteres no display cristalino	após medir a temperatura, ela será representada no display
2	Medidor de umidade relativa do ar	Medidor de umidade AHT10	$< 100\%$	Display LCD 16x2 1602A	alteração dos caracteres no display cristalino	após medir a umidade, ela será representada no display
4	Conexão Wi-Fi à rede	ESP32 / S3	disconnected	Led vermelho	Led vermelho acende	Quando o equipamento estiver desconectado à internet, o led vermelho ficará aceso
5	Sinalizador das condições naturais	medidor de temperatura e umidade relativa do ar	$28^{\circ}\text{C} < T < 36^{\circ}\text{C}$	Led RGB	Led RGB acende em verde	Quando as condições de temperatura estão favoráveis, o led verde ficará aceso.
			$70\% < U < 85\%$		Led RGB acende em verde	Quando as condições de umidade estão favoráveis, o led verde ficará aceso.
			$< 28^{\circ}\text{C}$ ou $> 36^{\circ}\text{C}$		Led RGB acende em vermelho	Quando a temperatura está muito baixa ou muito alta, o led vermelho ficará aceso.

			< 70% ou > 85%		Led RGB acende em vermelh o	Quando a umidade está muito baixa ou muito alta, o led vermelho ficará aceso.
6	Sinalizador de temperatura desfavorável	Sensor de temperatura	> 36 °C	Buzzer	Alta e constant e emissão de som	Quando a temperatura estiver muito alta, o buzzer irá fazer um som contínuo.

## 3.2. Interações

#	configuração do ambiente	ação do usuário	resposta esperada do sistema
1	É necessário um computador ou celular para se conectar à plataforma web da solução.	Usuário adiciona as estufas na plataforma web através do sinal wi-fi de cada dispositivo de medição e coleta de dados.	A interface do sistema adiciona um card referente a cada estufa registrada e retorna quando foi a última medição e quais são os valores de temperatura e umidade

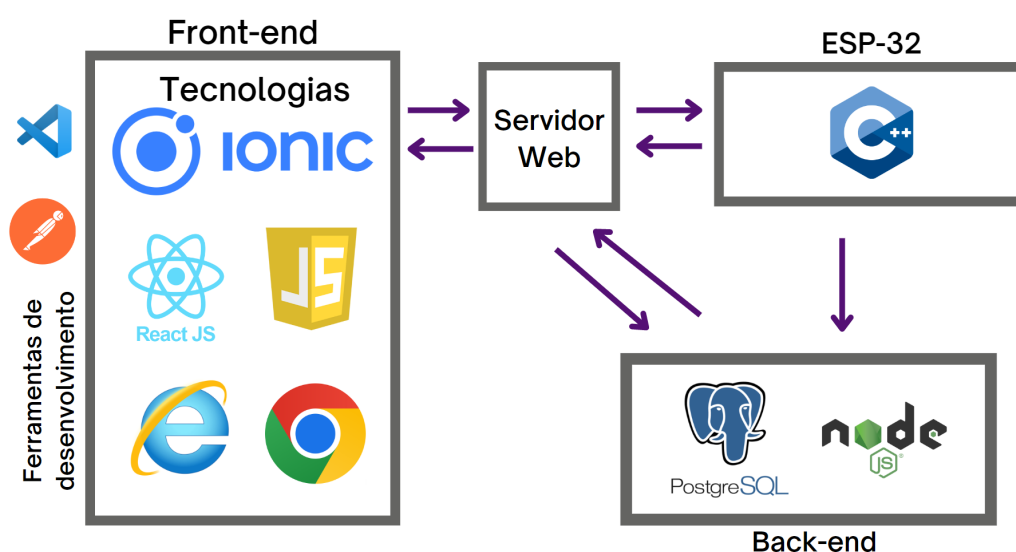
			registrados.
2	É necessário um computador ou celular para se conectar à plataforma web da solução.	O usuário clica no botão de notificações e no canto superior direito da interface.	A interface do sistema acessa o banco de dados e retorna as notificações de cada estufa com seus respectivos alertas de umidade e/ou temperatura .
3	É necessário um computador ou celular para se conectar à plataforma web da solução.	O usuário clica nos cards das estufas registradas no banco de dados do sistema.	A interface do sistema retorna um gráfico com os valores de temperatura e umidade das últimas medições, e recebe um retorno do status de abertura das janelas laterais e zenitais
4	É necessário um computador ou celular para se conectar à plataforma web da solução.	O usuário clica no card de status de abertura das janelas na página de status de cada estufa.	A interface do sistema acessa o banco de dados e retorna o histórico de abertura das janelas laterais e zenitais.
5	É necessário um computador ou celular para se conectar ao aplicativo nativo da solução.	O usuário abre o aplicativo nativo e clica no botão de configurar wifi	O usuário altera o nome e a senha da rede a ser conectada e salva, dessa forma, o sistema passa a se conectar a outro wifi.
6	É necessário um computador ou celular para se conectar ao aplicativo nativo da solução.	O usuário abre o aplicativo nativo e clica no botão de configurações gerais	O usuário faz alterações de nome da estufa conectada, quantidade de estufas, intervalo de tempo entre medições e salva no ESP32.
7	É necessário um computador ou celular para se conectar ao aplicativo nativo da solução.	O usuário abre o aplicativo nativo e clica no botão de extração manual.	O usuário receberá um arquivo com as últimas medições feitas que estão salvas no sistema para o caso de queda do wifi.
8	É necessário um computador ou celular para se conectar ao aplicativo nativo da	O usuário abre o aplicativo nativo e clica no botão de resetar configurações de fábrica.	O usuário irá fazer com que o dispositivo retorne ao padrão do sistema, apagando as últimas medições e as configurações salvas



			da memória do ESP32.
	solução.		
9	É necessário que o usuário possua o sistema final entregue junto com uma fonte .	O usuário liga o sistema à fonte.	O sistema liga e se conecta automaticamente com a plataforma web da solução, o display passa a exibir a temperatura e a umidade do ambiente, os sensores começam a enviar os dados coletados para o ESP32, que os envia para um banco de dados e os leds começam a funcionar de acordo com o estado do sistema.

## 4. Front-end, API e Banco de Dados

### 4.1. Tecnologias utilizadas



Nome	O que é	Em que foi utilizado	Versão
Visual Studio Code	Editor de código redefinido e otimizado para criar e depurar aplicativos modernos da Web e da nuvem.	Para a criação de todos os códigos presentes no frontend e backend.	Versão 1.67.2
Ionic + React Js	Framework para desenvolvimento mobile.	Para a construção do frontend da aplicação.	Versão 6.3.6
Postman	Ferramenta que dá suporte à documentação das requisições feitas pela API.	Para documentação e execução de testes de APIs e requisições em geral.	Versão 9.18.3
JavaScript	Linguagem de programação que é uma das principais tecnologias da World Wide Web.	Para o desenvolvimento do frontend e backend.	Versão ES6
C++	Linguagem de programação extensão da linguagem C.	Para o desenvolvimento do código fonte do microcontrolador.	Versão C++20
PostgreSQL	Mecanismo de banco de dados relacionais.	Para a montagem do banco de dados.	Versão 15.1
NodeJS	Ambiente de Execução Javascript, server-side, responsável pelo backend da aplicação.	Para o desenvolvimento do backend.	Versão 16.15.0 LTS

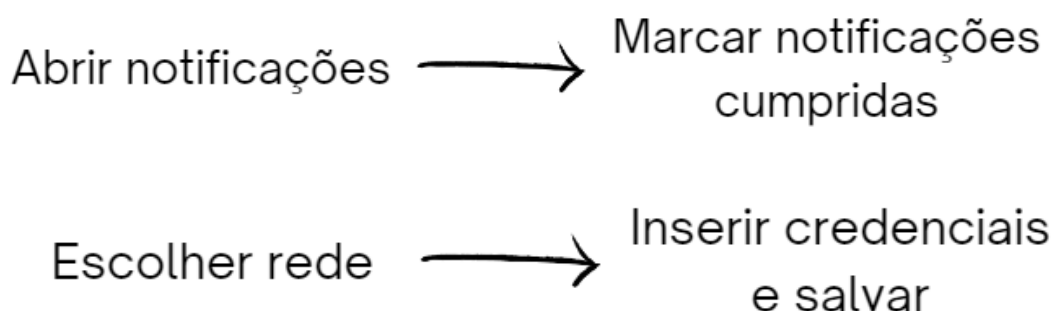
## 4.2. Front-end

### 4.2.1. Ferramenta utilizada

Para a construção do front-end, o framework Ionic utilizando React foi escolhido, principalmente, por oferecer a chance ao parceiro de modificar o design facilmente, mas

também, facilitando o esforço dos próprios desenvolvedores do projeto, já que o Ionic possui a capacidade de criar *layouts* de componentes que serão utilizados no design. Consequentemente, em caso da necessidade da troca de algo referente a um componente, ele será modificado diretamente em seu arquivo *layout* e não em todas as diferentes páginas em que é utilizado.

#### 4.2.2. Visão Geral



### 4.3. API's

As API's descrevem um conjunto de funções que tem por finalidade fazer uma ponte entre o front-end e o back-end, em que o usuário faz alguma solicitação através do front-end e a API recebe tal comando e faz a operação determinada com o banco de dados no back-end.

#### 4.3.1. API de dispositivos

A API de dispositivos descreve quais são os comandos estabelecidos para gerenciar as informações dos dispositivos do sistema no banco de dados.

```
1 // Required for accessing the database
2 const { PrismaClient, Prisma } = require("@prisma/client");
3 const prisma = new PrismaClient();
4
5 // Required for route handling
6 const express = require("express");
7 const router = express.Router();
8
```

Tal trecho é necessário para o acesso ao banco de dados da solução e para a manipulação das rotas.

```

11 // [GET] requests
12
13 router.get("/list", async (request, response) => {
14   // returns all the dispositivos that are in the database
15   const dispositivos = await prisma.dispositivo.findMany();
16   const responseObj = { data: dispositivos, error: false };
17   return response.json(responseObj);
18 });
19
20 router.get("/list/detailed", async (request, response) => {
21   // returns all the dispositivos that are in the database
22   const dispositivos = await prisma.dispositivo.findMany({
23     include: {
24       Medidas: {
25         orderBy: {
26           id: "desc",
27         },
28         take: 1,
29       },
30     },
31   });
32   const responseObj = { data: dispositivos, error: false };
33   return response.json(responseObj);
34 });

```

Nesse trecho, são descritas as solicitações “GET”. As solicitações “GET” são aquelas em que é feita a requisição no front-end e a API busca tais dados no banco e os retorna para o usuário. Dito isso, a primeira requisição retorna todos os dispositivos que estão no banco de dados. Já a segunda requisição retorna os mesmos dispositivos porém com dados adicionais de medidas (de temperatura e umidade da última medição do sistema).

```

38 router.post("/add", async (request, response) => {
39   const dispositivoMac = request.body.mac;
40   const dispositivoName = request.body.nome || dispositivoMac;
41   // checks if the current mac exists on the database
42   const dispositivoExists = await prisma.dispositivo.findUnique({
43     where: {
44       mac: dispositivoMac,
45     },
46   });
47
48   // Returns an error if the dispositivo already exists
49   if (dispositivoExists) {
50     response.statusCode = 500;
51     return response.json({
52       message: "O Dispositivo já existe",
53       data: {},
54       error: true,
55     });
56   }
57 }

```

```

58 // creates the dispositivo
59 const dispositivo = await prisma.dispositivo.create({
60   data: {
61     estufa: dispositivoName,
62     mac: dispositivoMac,
63   },
64 });
65
66 // returns the newly created dispositivo
67 response.statusCode = 200;
68 return response.send({
69   message: "Dispositivo Adicionado com sucesso!",
70   data: dispositivo,
71   error: false,
72 });
73 });
74

```

Nesse trecho do código, é descrita a primeira solicitação “POST” da API de dispositivos. As solicitações “POST” são aquelas em que o usuário faz uma alteração do banco de dados no back-end através do front-end. Nesse caso, tal rota permite ao usuário adicionar novos dispositivos no banco de dados.

```

75 router.post("/edit", async (request, response) => {
76   const dispositivoId = request.body.id;
77   const dispositivoName = request.body.nome;
78
79   // checks if the current mac exists on the database
80   const dispositivoExists = await prisma.dispositivo.findUnique({
81     where: {
82       id: dispositivoId,
83     },
84   });
85

```

```
86 // Returns an error if the dispositivo already exists
87 if (dispositivoExists) {
88     // updates the dispositivo by id
89     const updateDispositivo = await prisma.dispositivo.update({
90         where: {
91             id: dispositivoId,
92         },
93         data: {
94             estufa: dispositivoName,
95         },
96     });
97
98     response.statusCode = 200;
99     return response.json({
100         message: "Dispositivo editado com sucesso",
101         data: updateDispositivo,
102         error: false,
103     });
104 }
105
106 response.statusCode = 500;
107 return response.json({ message: "Erro ao editar dispositivo", error: true });
108 });
109
110 module.exports = router;
```

Nesse trecho do código, é descrita outra solicitação "POST". Nesse caso, tal rota busca editar um dispositivo que esteja no banco de dados a partir de uma requisição feita no front-end.

#### 4.3.2. API de medidas

A API de medidas descreve quais são os comandos estabelecidos para gerenciar as informações das medidas de temperatura e umidade realizadas pelo sistema, contidas no banco de dados.

```

1  // Required for accessing the database
2  const { PrismaClient, Prisma } = require("@prisma/client");
3  const prisma = new PrismaClient();
4
5  // Required for route handling
6  const express = require("express");
7  const { MetricValidator } = require("../utils/metrics");
8  const { Notificacao } = require("../utils/notificacao");
9
10 // Router Controllers
11
12 const router = express.Router();
13
14 // controller
15

```

Tal trecho é necessário para o acesso ao banco de dados da solução e para a manipulação das rotas.

```

16 // GET requests
17 router.get("/list", async (request, response) => {
18   // returns all the medidas that are in the database
19   const medidas = await prisma.medidas.findMany();
20   const responseObj = { data: medidas, error: false };
21   return response.json(responseObj);
22 });
23
24 router.get("/list/dispositivo/:id", async (request, response) => {
25   // returns the medidas from a dispositivo that are in the database
26   const idDispositivo = Number(request.params.id);
27   const medidas = await prisma.medidas.findMany({
28     where: {
29       | dispositivoId: idDispositivo,
30     },
31   });
32   const responseObj = { data: medidas, error: false };
33   return response.json(responseObj);
34 });
35

```

Tal trecho descreve as solicitações “GET” da API de medidas. Nesse caso, a primeira rota tem a função de retornar todas as medidas que estão no banco de dados. Já a segunda tem a função de retornar as medidas de um determinado dispositivo presente no banco de dados.

```

36 // POST requests
37 router.post("/add", async (request, response) => {
38   const dispositivoId = Number(request.body.dispositivoId);
39   const medidasTemperatura = request.body.temperatura;
40   const medidasUmidade = request.body.umidade;
41   const medidasDatetime = request.body.datetime;
42
43   // creates the medida
44   const medidas = await prisma.medidas.create({
45     data: {
46       dispositivoId: dispositivoId,
47       temperatura: medidasTemperatura,
48       umidade: medidasUmidade,
49       datetime: medidasDatetime,
50     },
51   });
52
53   const validator = new MetricValidator(medidasTemperatura, medidasUmidade);
54   const statusList = validator.verifyRules();
55
56   statusList.forEach(async (statusCode) => {
57     const notificacao = new Notificacao(dispositivoId, medidas.id, statusCode);
58     const createNotification = await notificacao.createNotification();
59     if (createNotification == true) {
60       console.log("Notificação Gerada!!");
61     }
62   });
63
64   // returns the newly created medidas
65   response.statusCode = 200;
66   return response.send({
67     message: "Medidas Adicionadas com sucesso!",
68     data: medidas,
69     error: false,
70   });
71 });
72
73 module.exports = router;
74

```

Tal trecho de código descreve a solicitação “POST” da API de adicionar as medições feitas pelo sistema no banco de dados. Nesse caso, as medições são adicionadas com temperatura, umidade e horário da medição.

### 4.3.3. API de notificações

A API de notificações descreve quais são os comandos estabelecidos para gerenciar as informações das notificações contidas no banco de dados.



```

1  //const express = require("express");
2
3  const { PrismaClient, Prisma } = require("@prisma/client");
4  const prisma = new PrismaClient();
5
6  const express = require("express");
7  const router = express.Router();
8
9  // Router Controllers
10
11 //const router = express.Router();
12
13 // controller
14

```

Tal trecho é necessário para o acesso ao banco de dados da solução e para a manipulação das rotas.

```

15 // GET requests
16 router.get("/list/active", async (request, response) => {
17   const notificacoes = await prisma.notificacao.findMany({
18     where: {
19       status: 0,
20     },
21     include: {
22       dispositivo: {
23         select: {
24           estufa: true,
25         },
26       },
27       gatilho: {
28         select: {
29           temperatura: true,
30           umidade: true,
31         },
32       },
33     },
34   });
35   const responseObj = { data: notificacoes, error: false };
36   return response.json(responseObj);
37   //response.send({ message: "hello world! - list" });
38 });
39

```

Tal trecho descreve as solicitações “GET” da API de notificações. Nesse caso, a rota tem a função de retornar todas as notificações que estão no banco de dados.

```

41 // requisição POST para adicionar uma notificação
42 router.post("/add", async (request, response) => {
43   //definindo as constantes pelo banco de dados (CONFERIR)
44   const notificacaoDispositivoId = request.body.id;
45   const notificacaoStatus = request.body.status;
46   const notificacaoTipo = request.body.tipo;
47   const notificacaoGatilho = request.body.gatilho;
48
49   // criando a notificação
50   const notificacoes = await prisma.notificacao.create({
51     data: {
52       dispositivoId: notificacaoDispositivoId,
53       status: notificacaoStatus,
54       gatilhoId: notificacaoGatilho,
55       tipo: notificacaoTipo,
56     },
57   });
58

```

```

59 // retorna a nova notificação com uma mensagem informando que a notificação foi adicionada
60 response.statusCode = 200;
61 return response.send({
62   message: "Notificação adicionada com sucesso!",
63   data: notificacoes,
64   error: false,
65 });
66 });

```

Tal trecho de código descreve a solicitação “POST” da API de adicionar uma nova notificação no banco de dados. Nesse caso, as notificações variam de tipo a partir das condições de temperatura e umidade do ambiente.

```

67 //requisição POST para atualizar o status das notificações (VER SE é isso mesmo)
68 router.post("/status", async (request, response) => {
69   //response.send({ message: "hello world! - status" });
70   const notificacaoId = request.body.id;
71   const notificacaoName = request.body.status;
72
73   // confere se a notificação já existe
74   const notificacaoExists = await prisma.notificacao.findUnique({
75     where: {
76       id: notificacaoId,
77     },
78   });
79
80   if (notificacaoExists) {
81     // atualiza a notificação pelo id dela
82     const updateNotificacoes = await prisma.notificacao.update({
83       where: {
84         id: notificacaoId,
85       },
86       data: {
87         status: notificacaoName,
88       },
89     });

```

```

90   //caso a notificação seja alterada com sucesso
91   response.statusCode = 200;
92   return response.json({
93     message: "Status editado com sucesso",
94     data: updateNotificacoes,
95     error: false,
96   });
97 }
98 //caso não consiga alterar o status da notificação uma mensagem aparecerá.
99 response.statusCode = 500;
100 return response.json({ message: "Erro ao editar status", error: true });
101 });
102
103 module.exports = router;
104

```

Tal trecho de código descreve a solicitação “POST” da API de editar uma notificação no banco de dados. Nesse caso, a API vai verificar se a notificação existe através de seu id e atualizá-la.

## 4.4. Banco de Dados

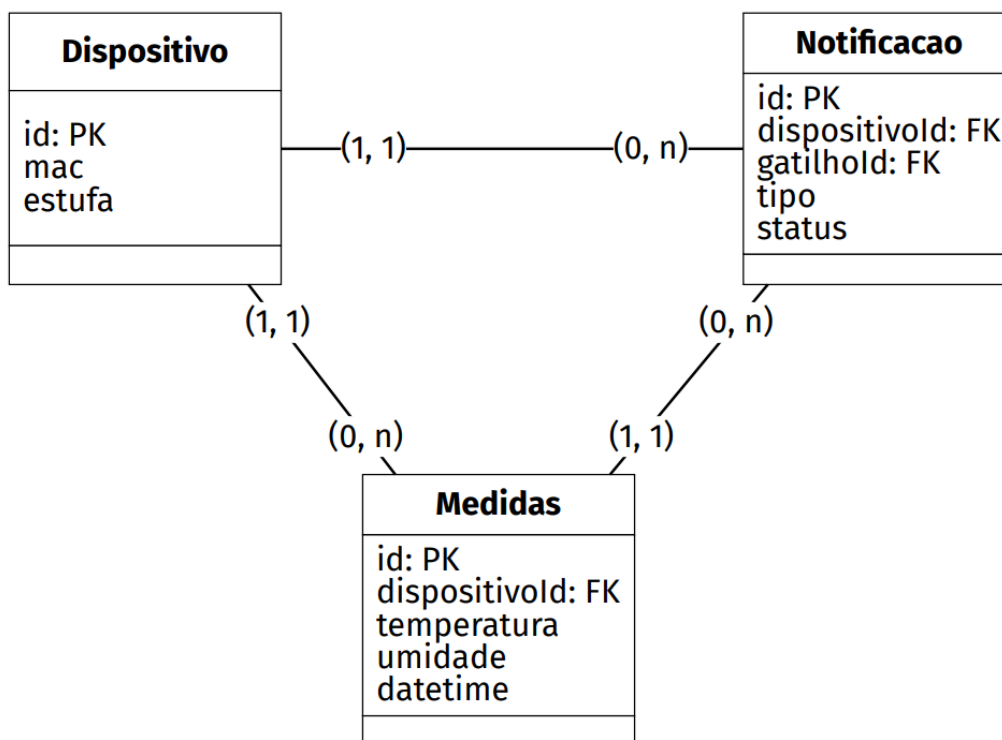
Os bancos de dados são utilizados para armazenar dados captados e extrair informações.

O modelo do banco de dados que adotamos foi o modelo relacional. Esse modelo permite que os dados sejam armazenados de forma relacionada criando codependência entre certas tabelas dependendo da necessidade da solução.

### 4.4.1. Ferramenta utilizadas

Para a construção do banco de dados utilizamos o PostgreSQL por conta da alta escalabilidade, ser uma das ferramentas mais utilizadas no mercado e por ter uma ótima performance com relação a alta quantidade de dados, assim sendo capaz de suportar o alto fluxo de dados recebidos do ESP32.

### 4.4.2. Modelo lógico



Legenda:

(1, 1): um para um.

(0, n): nenhum para muitos.

PK : Primary Key ou Chave primária.

FK: Foreign Key ou Chave estrangeira.

**Entidades:**

- Dispositivo
- Medidas
- Notificacao

**Dispositivo**

Esta entidade é responsável por armazenar dados de um novo dispositivo ESP32. Possui como identificador, o MAC address do dispositivo, o qual deve ser único, não permitindo a inserção de novos ESP32s caso estes venham a apresentar a mesma identificação. Também, para facilitar a localização e reconhecimento de cada um dos microcontroladores, há a presença de uma terceira coluna, chamada estufa, a qual recebe um nome definido no cadastro do dispositivo.

**Medidas**

A entidade medidas é responsável por armazenar os dados recebidos de um determinado dispositivo que é identificado pela chave estrangeira “dispositivold” que é recebida da tabela dispositivos. Os dados que essa entidade recebe são os dados de temperatura, umidade e datetime(o horário que as medidas foram tiradas).

**Notificações**

A entidade notificações tem como responsabilidade primordial, o armazenamento e controle de estado de notificações, e alertas, referentes a temperatura e umidade de uma medição. Com isso, existem duas chaves estrangeiras: “dispositivold” e “gatilhold”. A primeira constrói o relacionamento entre o alerta e uma estufa, identificando o dispositivo de origem. A segunda, aponta para uma medida de temperatura e umidade, da tabela medidas, ocasionadora do alerta. Dessa forma, só é permitida a criação destes registros, se um dispositivo e medida estiverem presentes e válidos, garantindo a validade do alerta gerado.

## Anexos

[Manual de instruções](#)

[Figma](#)