
PLATAFORMA PARA MONITORAMENTO E ANÁLISE DE TEMPERATURA E UMIDADE

Integrantes:

Guilherme Luís Rodrigues Silva

Jorge Luiz Patrocínio dos Santos

Yago Raphael de Melo Mouro

Rafael Victor Redoval de Souza

24/05/2025

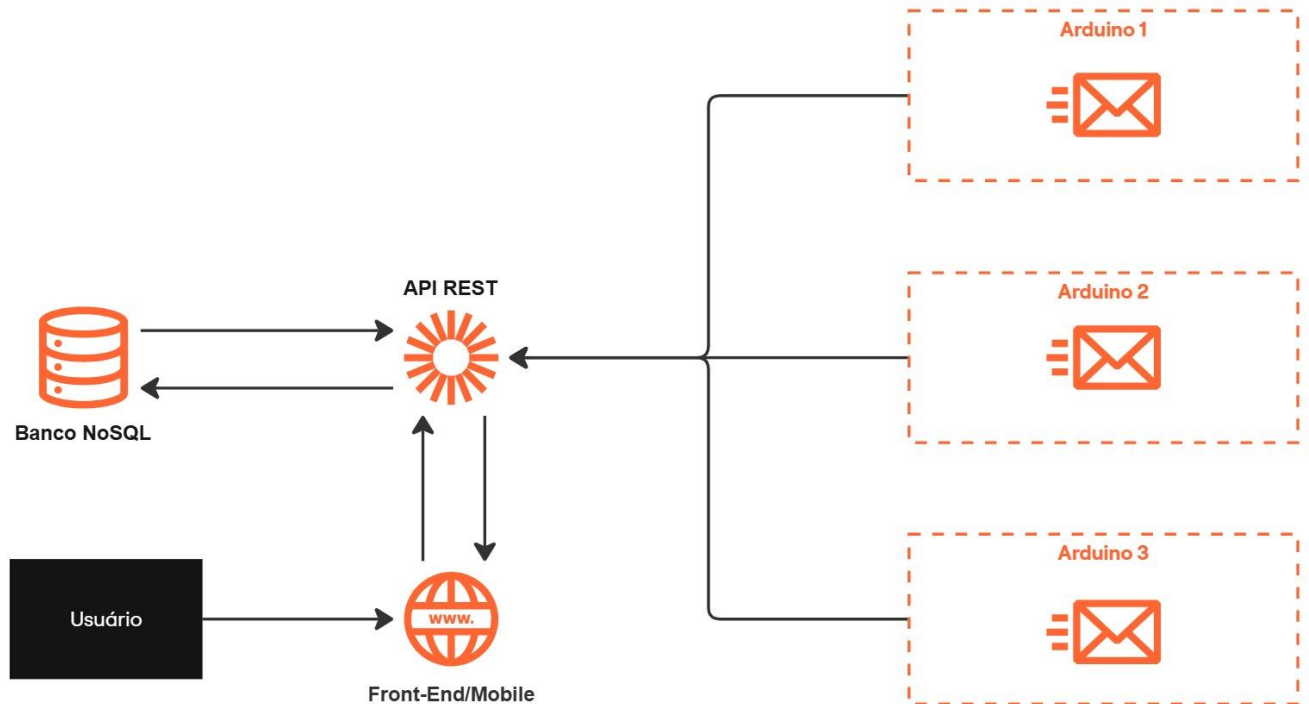
PAPÉIS E RESPONSABILIDADES DOS INTEGRANTES:

- **Guilherme Luís Rodrigues Silva:**
 - Front-End: React.js, Redux, Recharts, Biblioteca de testes
- **Jorge Luiz Patrocínio dos Santos:**
 - Back-End: Node.js, Express, MongoDB, Swagger
- **Yago Raphael de Melo Mouro:**
 - Protótipo Arduino: NodeMCU ESP8266, Sensor DHT11
- **Rafael Victor Redoval de Souza:**
 - Mobile: React Native

GUILHERME LUÍS RODRIGUES SILVA Front-End	JORGE LUIZ PATROCÍNIO DOS SANTOS Back-End	YAGO RAPHAEL DE MELO Mouro Arduino	RAFAEL VICTOR REDOVAL DE SOUSA Mobile
Framework Principal: <ul style="list-style-type: none">• React.js	Ambiente de Desenvolvimento da API: <ul style="list-style-type: none">• Node.js• Express	Construção do Protótipo: <ul style="list-style-type: none">• Placa Wi-Fi: Node MCU ESP8266• Lolin• Sensor: DHT11	Framework Principal: <ul style="list-style-type: none">• React Native
Principais Bibliotecas: <ul style="list-style-type: none">• Redux (G. Estado)• Recharts (Gráficos)	Banco de Dados: <ul style="list-style-type: none">• MongoDB - NoSQL	Linguagem: <ul style="list-style-type: none">• C/C++	Principais Bibliotecas: <ul style="list-style-type: none">• Redux (G. Estado)• Recharts (Gráficos)
Funções: <ul style="list-style-type: none">• Exibição das informações do Arduino de forma legível.	Documentação: <ul style="list-style-type: none">• Swagger• OpenAPI 3.0	Funções: <ul style="list-style-type: none">• Coleta dos dados através do sensor DHT11.• Envio das informações via Wi-Fi para API	Funções: <ul style="list-style-type: none">• Exibição das informações do Arduino de forma legível no app sem quebrar o layout.

VISÃO GERAL DO PROJETO:

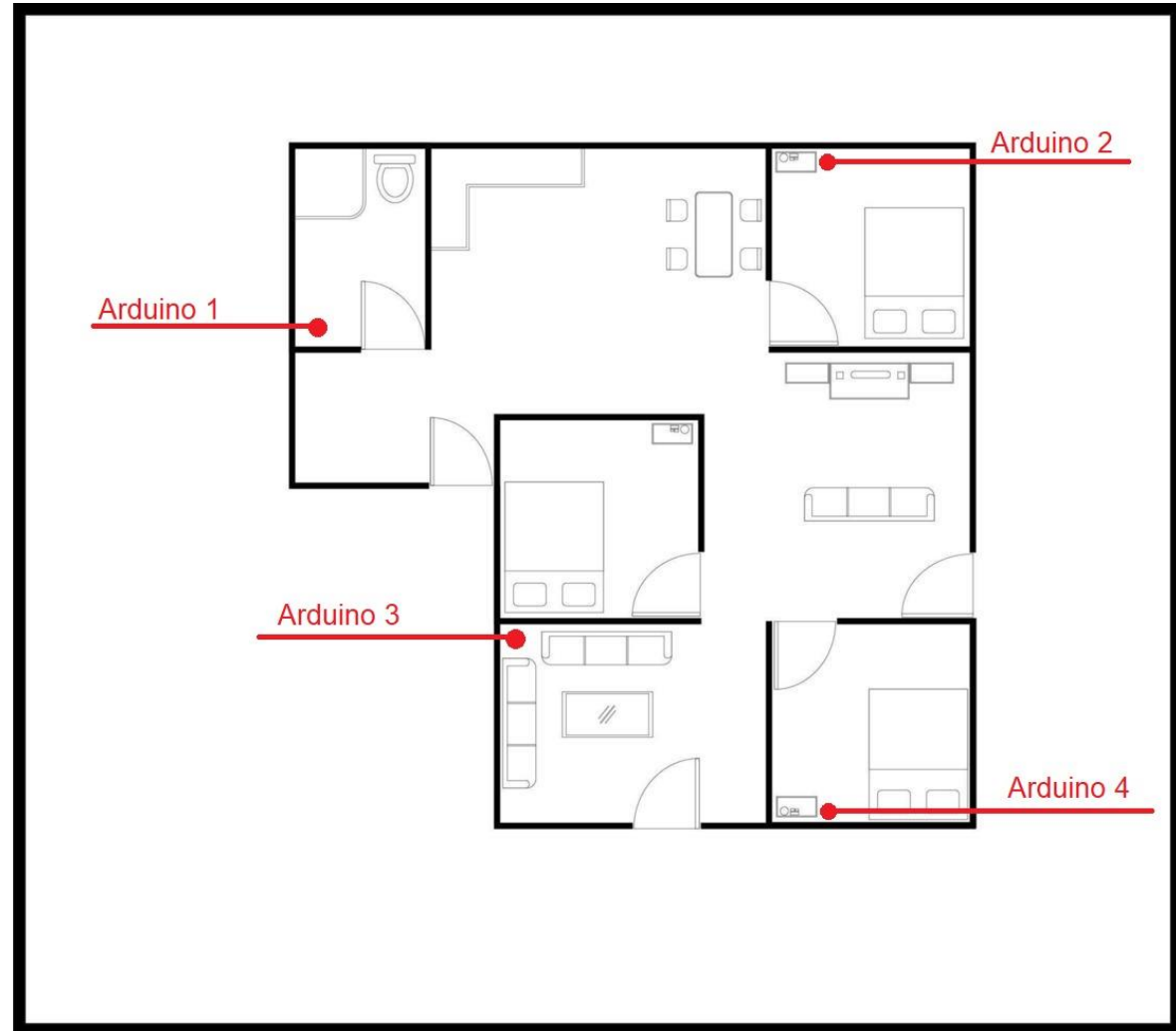
A ideia do nosso projeto é um Dashboard web e app mobile que permite o usuário monitorar temperatura e umidade em tempo real, integrando múltiplos Arduinos via Wi-Fi a uma API REST.

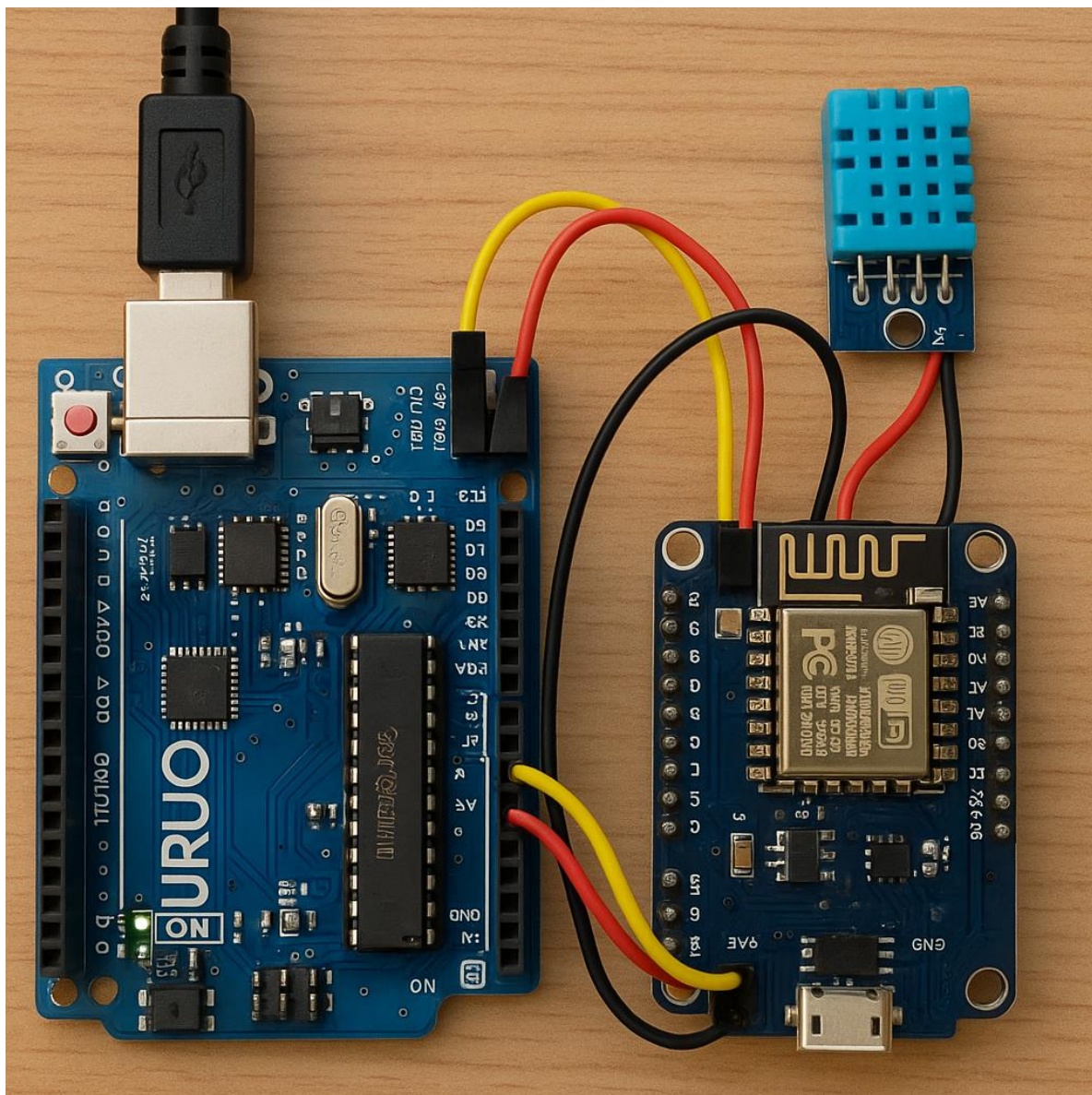


REGRAS DE NEGÓCIO:

Foi definido que o projeto deve ter as seguintes regras de negócios:

1. Identificação Obrigatória dos dispositivos.
2. Faixas válidas de temperaturas.
3. Atualização contínua de dados.
4. Classificação visual por cores do Arduino.
5. Visualização pública sem login.
6. Armazenamento com timestamp.
7. Responsividade da Interface.





PROTÓTIPO E COMPONENTES DO IOT:

Os componentes que irão compor o Arduino são:

1. Placa de Desenvolvimento Microcontrolada ATmega328P – Arduino Uno.
2. Placa de Desenvolvimento Wi-Fi NodeMCU ESP8266.
3. Sensor Digital de Temperatura e Umidade DHT11.



INTEGRAÇÃO IOT AO API REST:

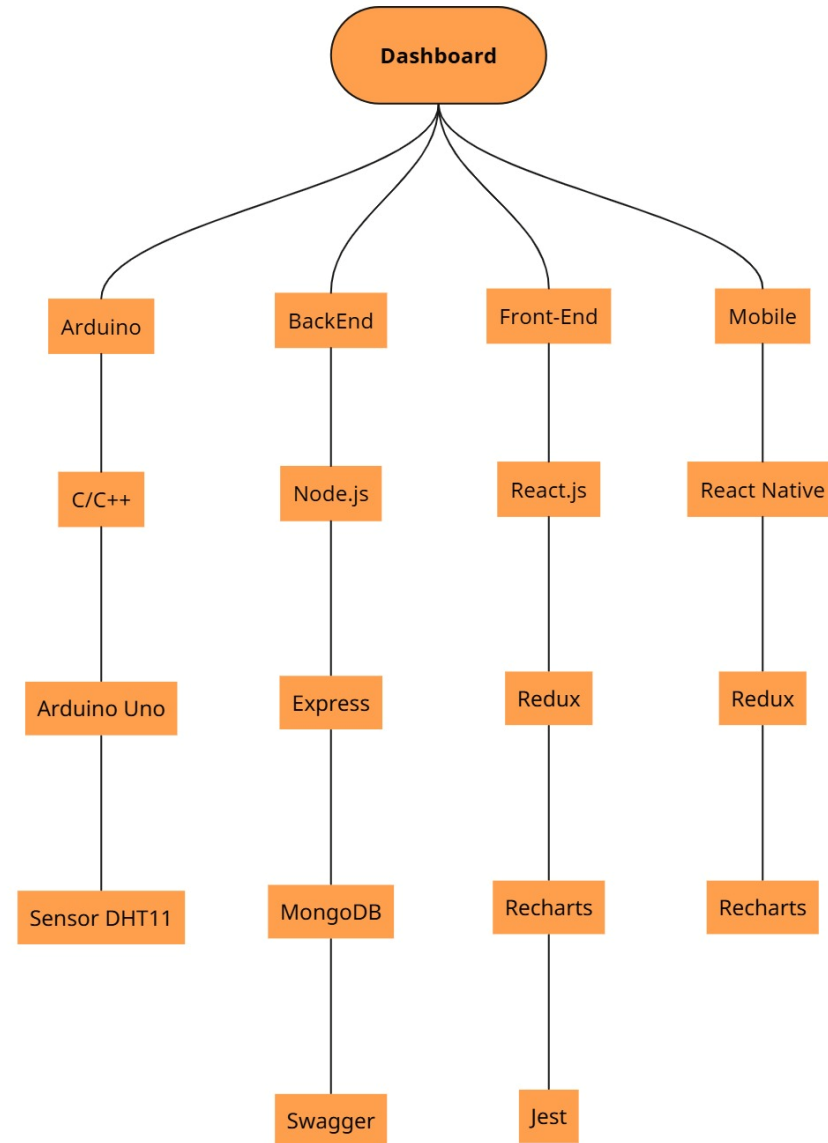
A integração do IOT com a nossa API irá ocorrer da seguinte forma:

1. O Arduino Uno, em conjunto com o módulo Wi-Fi NodeMCU ESP8266, será responsável por enviar pacotes no formato JSON com os seguintes campos:
 - {humidty, location, temperature, timestamp_TTL}
 2. A API REST, desenvolvida em Node.js com Express, receberá esses dados por meio de requisições HTTP POST. Após a validação dos campos, as informações serão armazenadas no banco de dados MongoDB para posterior exibição no dashboard web e mobile.
-

TECNOLOGIAS E FRAMEWORKS UTILIZADOS:

As tecnologias e frameworks que iremos utilizar em nosso projeto pode ser colocada como:

- **Arduino:** C/C++, Arduino Uno, ESP8266, Sensor DHT11.
- **Back-End(API-REST):** Node.js, Express, MongoDB(NoSQL), Swagger.
- **Front-End WEB:** React.js, Redux, Recharts, Jest.
- **Front-End Mobile:** React-Native, Redux, Recharts.



ENDPOINTS DA API REST:

A API REST do back-end irá possuir os seguintes Endpoints:

1. POST /api/data

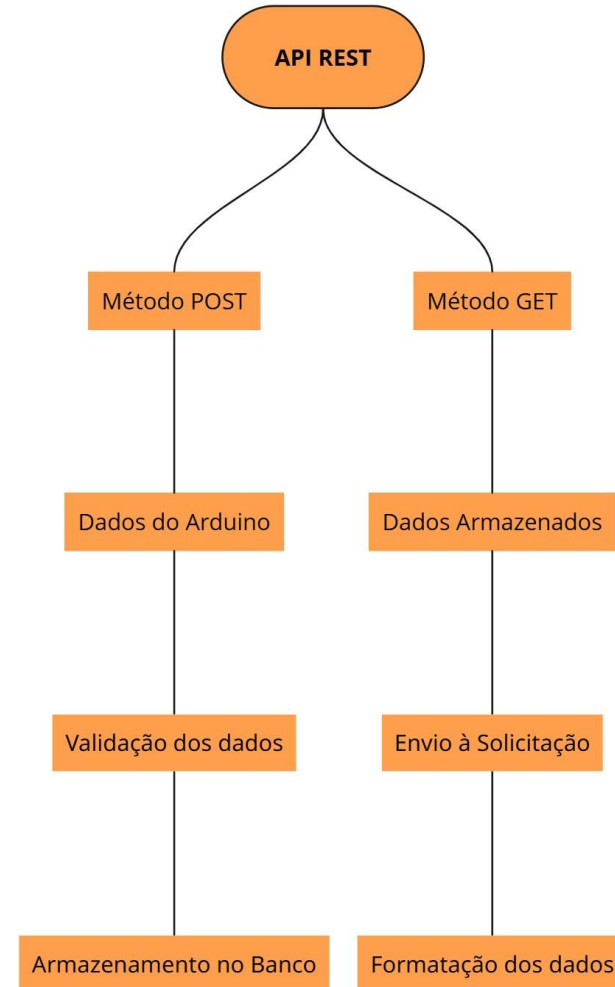
- Recebe as leituras do Arduino
- Payload: {humidty, location, temperature, timestamp_TTL}

2. GET /api/data/latest

- Retorna o último valor por dispositivo

3. GET /api/data?start={ISO}&end={ISO}

- Retorna histórico no intervalo especificado



PROTOTIPAÇÃO DAS TELAS – WEB:

- A prototipação das telas foram realizadas conforme a elicitação de requisitos do projeto:

1. Home com Dashboard:

- Foi criado um dashboard com cards que podem ser expandidos para exibição das informações coletada pelo arduino

2. Configuração de Tempo dos Arduinos

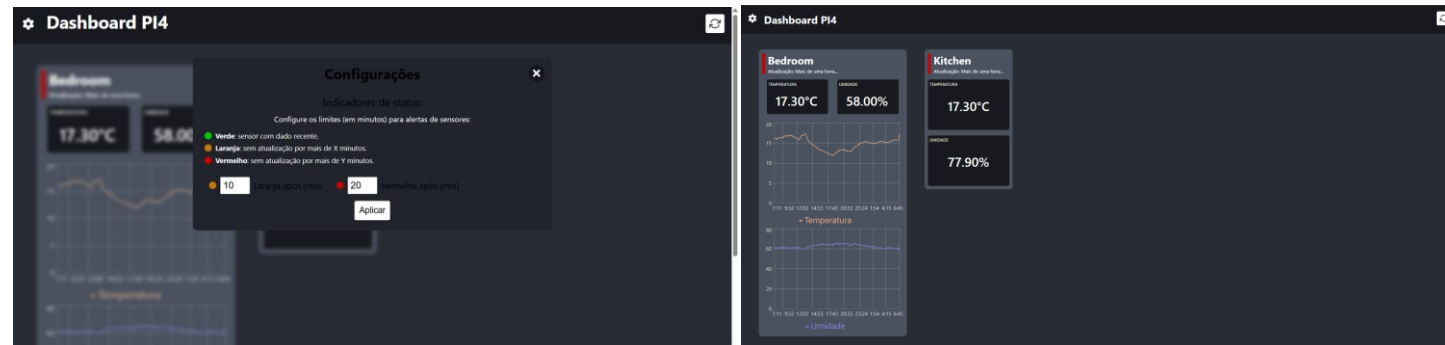
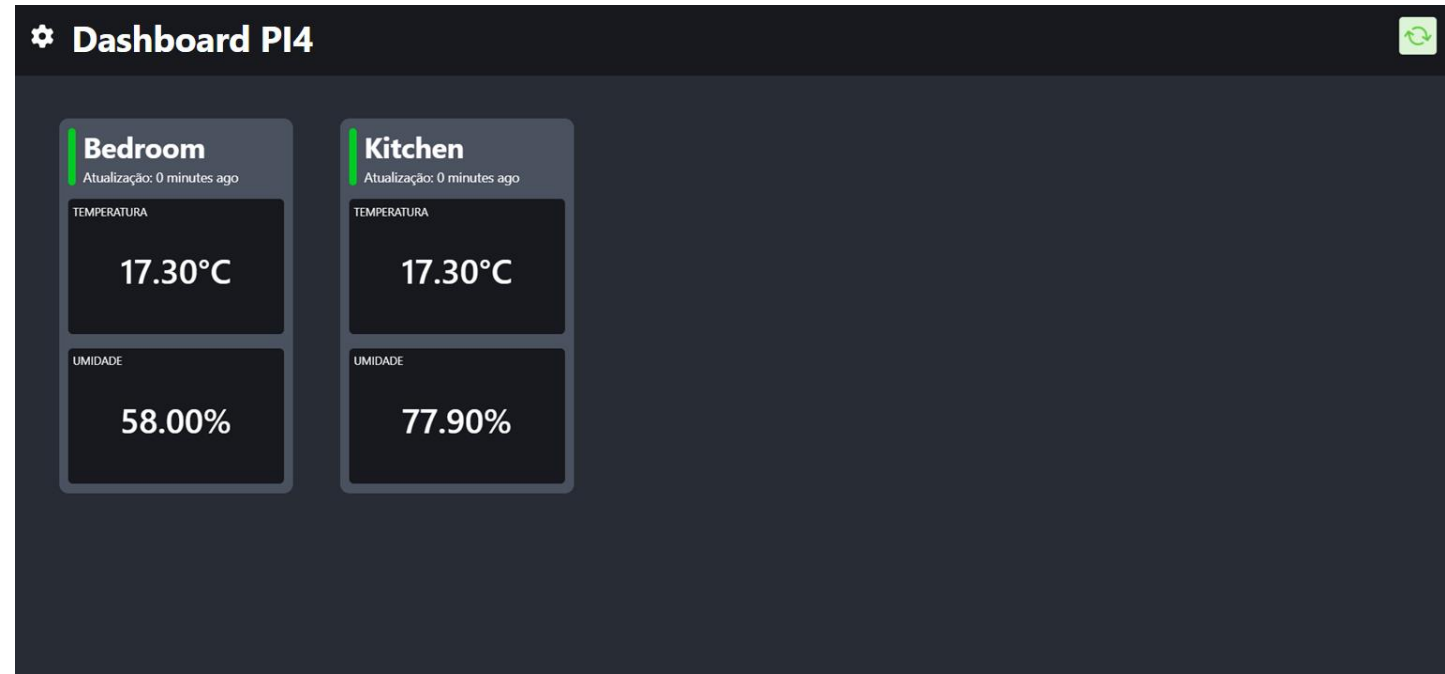
- É possível alterar o tempo de exibição das cores indicativas do Arduino

3. Exibição de Múltiplos Arduinos conectados

- O dashboard comporta mais de um Arduino simultaneamente

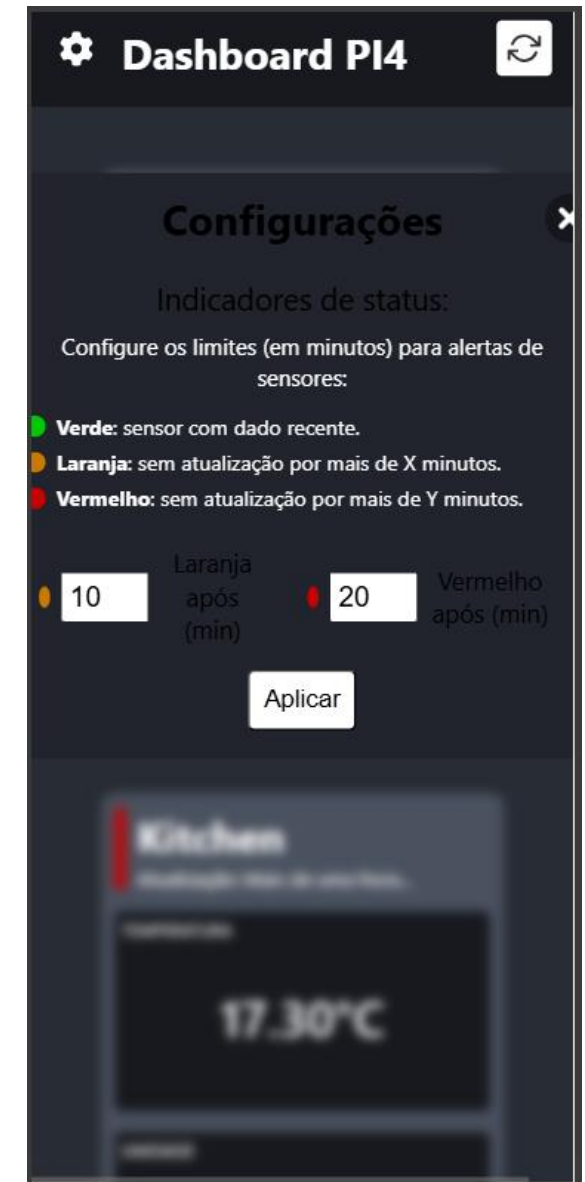
4. Atualização dos Arduinos

- É possível atualizar manualmente a coleta de dados feita pelo Arduino.



PROTOTIPAÇÃO DAS TELAS – MOBILE:

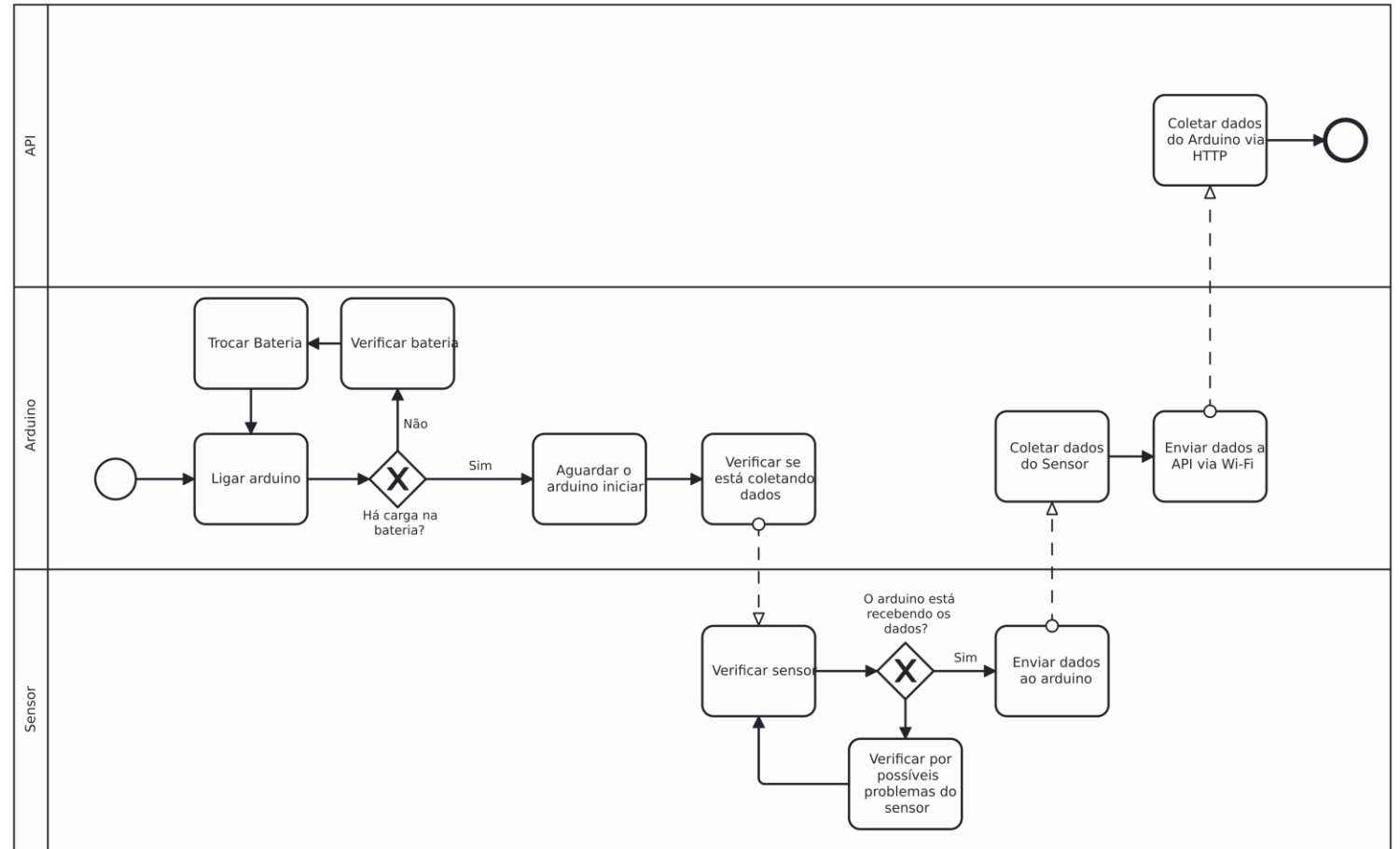
- A prototipação das telas do Mobile foram realizadas baseada na prototipação de tela do modelo WEB:
1. Home Mobile:
 - Foi criado um dashboard dinâmico com cards que podem ser expandidos para exibição das informações coletada pelo Arduino.
 2. Design responsive, navegação simples e feedback visual:
 - É possível alterar o tempo de exibição das cores indicativas do Arduino



DIAGRAMAS DE PROCESSOS – BUSINESS PROCESS MODEL AND NOTATION

1. Fluxo Arduino:

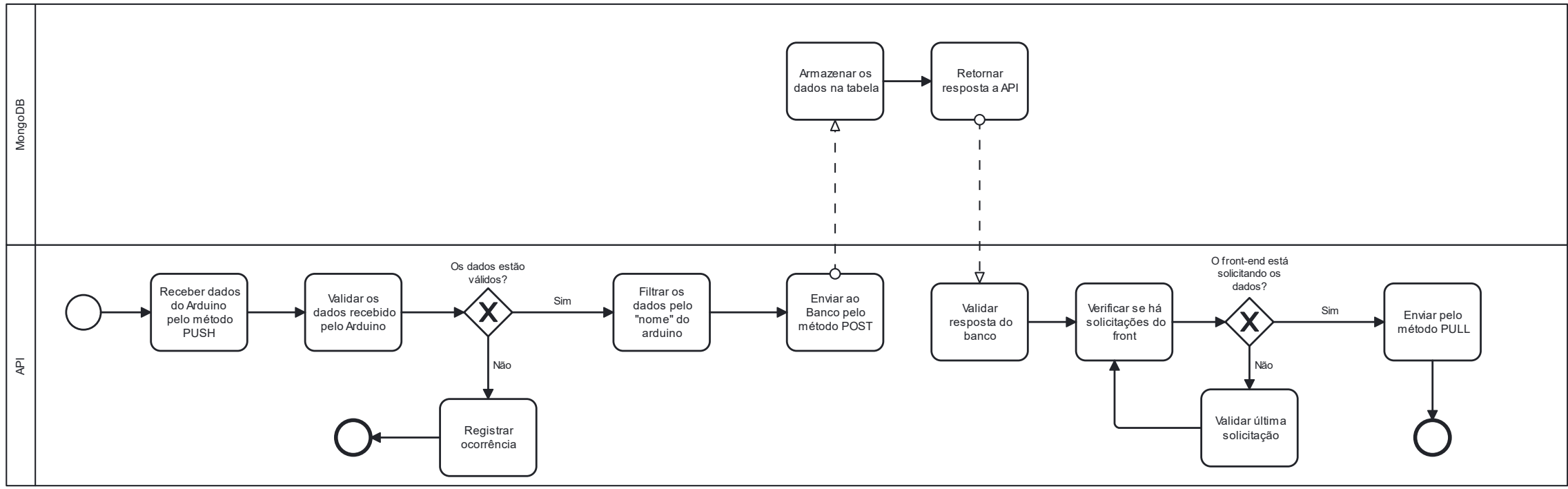
- O diagrama ao lado representa o fluxo de tarefas que cada Arduino deve executar. Nele, há três piscinas representando:
 - Sensor: DHT11.
 - Arduino: Modelo UNO.
 - API: API-REST.



DIAGRAMAS DE PROCESSOS – BUSINESS PROCESS MODEL AND NOTATION

1. Fluxo Back-End:

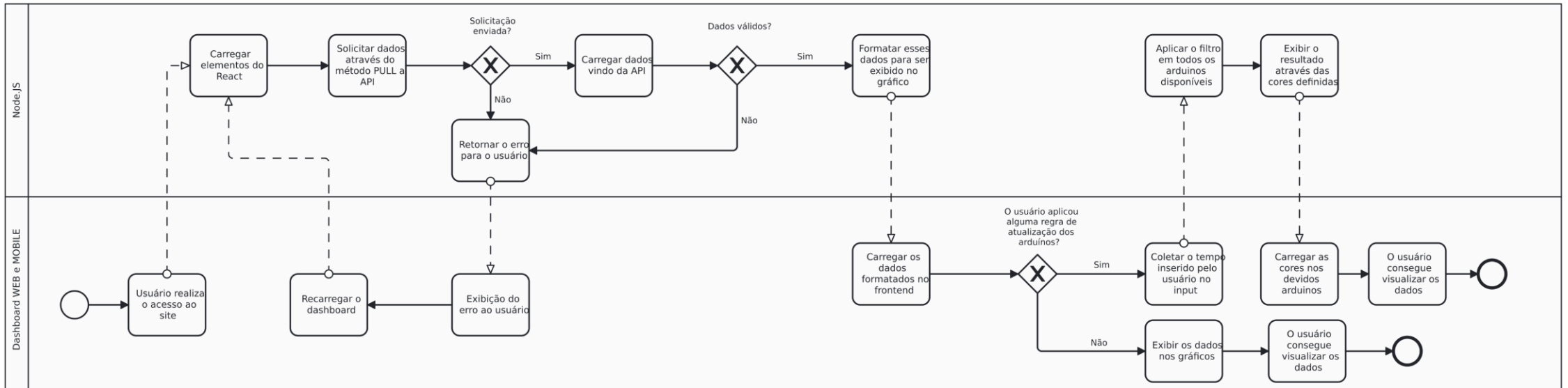
- O diagrama abaixo representa o fluxo das responsabilidades do Back-End do projeto. Ele é composto por duas piscinas, que representam:
 - MongoDB: Baseado em NoSQL.
 - API: Node.js, Express e documentada com Swagger.



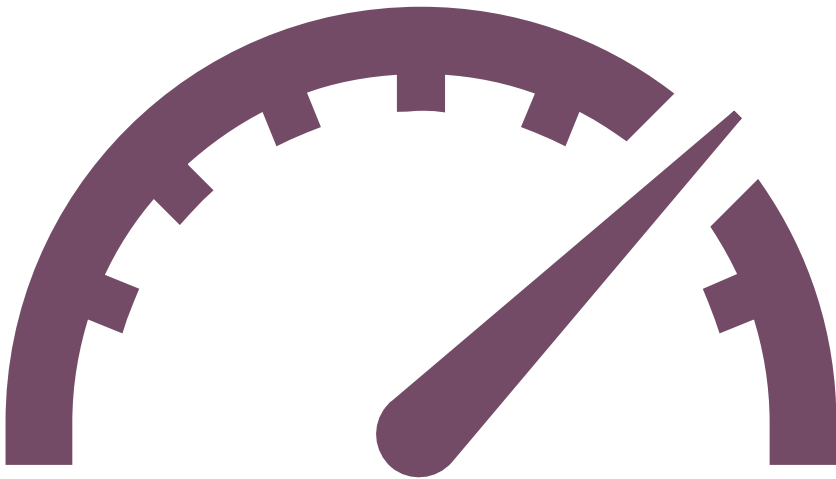
DIAGRAMAS DE PROCESSOS – BUSINESS PROCESS MODEL AND NOTATION

1. Fluxo Front-End:

- O diagrama abaixo representa o fluxo de responsabilidades do Front-End do projeto. Ele é composto por duas piscinas, que representam:
 - Interface(WEB, Mobile): Baseado em React.js, Redux, Rechart, Jest.
 - Linguagem: Node.js.



CONCLUSÃO DO PROJETO:



A proposta do nosso projeto é o desenvolvimento de um dashboard intuitivo e acessível, que possibilita ao usuário acompanhar em tempo real os dados de temperatura e umidade por meio de uma interface Web e também via aplicativo mobile. O grande diferencial da nossa solução está na praticidade de uso e na capacidade de análise das informações apresentadas. Diferente de soluções mais limitadas, nosso sistema permite o monitoramento simultâneo de múltiplos dispositivos Arduino, oferecendo uma visão mais ampla e eficiente para o usuário final.