

Projeto de PI - SmartPlant

1. Nome dos Integrantes

- Paulo Henrique Borges de Andrade Filho (Responsável pelo backend - Node.js e integração IoT)
- Nathan Bizinoto (Responsável pelo frontend - Next.js e integração IoT)
- Vinicius Rodrigues (Responsável pelo mobile dashboard e integração IoT)

2. Nome do Projeto

SmartPlant - Monitoramento Inteligente de Plantas com IoT

3. Regras de Negócio e Telas

- O sistema deve coletar automaticamente os dados de temperatura e umidade do solo via sensor IoT.
- Os dados são enviados em tempo real para a API através de uma conexão Wi-Fi.
- Caso a umidade esteja abaixo de um limite configurável, um alerta é registrado.
- Os usuários podem visualizar os dados históricos e em tempo real através de um painel web ou aplicativo mobile.
- O sistema permitirá login e cadastro de usuários para personalização e visualização dos dados vinculados à planta.

Telas (Website e Mobile):

- - Tela de login/cadastro
- - Dashboard com dados de sensores em tempo real (gráficos de temperatura e umidade)
- - Histórico de dados por data
- - Configuração de limites de alerta

4. Integração com IoT

Será utilizado um microcontrolador ESP32 com sensor DHT11 ou DHT22 para coletar os dados ambientais.

O dispositivo enviará os dados via HTTP em formato JSON para a API desenvolvida em Node.js.

5. Tecnologias Utilizadas

Front-End (Website): Next.js + Tailwind CSS

Back-End (API): Node.js + Express + Prisma (PostgreSQL)

Mobile (Dashboard): React Native + Expo

IoT (Dispositivo): C++ com Arduino IDE, Bibliotecas: WiFi.h, HTTPClient.h, DHT.h

6. APIs e Endpoints

- Endpoints a serem desenvolvidos:
 - - POST /sensor: recebe os dados de temperatura e umidade do ESP32
 - - GET /dados/:plantald: retorna dados históricos da planta
 - - POST /login: autentica o usuário
 - - POST /cadastro: cria novo usuário
 - - POST /alerta-config: configura limite de alerta

7. Integração IoT com APIs

O ESP32 se conecta à internet e envia periodicamente os dados dos sensores para o endpoint /sensor utilizando uma requisição POST com JSON. Esses dados são processados pela API e armazenados no banco de dados. O frontend e o mobile consomem esses dados via endpoints REST para exibição em tempo real.

8. Divisão das Atividades

Cada integrante é responsável por uma etapa do projeto. No documento final entregue no Teams, o nome de cada integrante estará associado à sua respectiva atividade.

- - IoT: Nathan Bizinoto, Paulo Henrique de Andrade e Vinicius Rodrigues
- - Backend API: Paulo Henrique de Andrade
- - Frontend Website: Nathan Bizinoto
- - Mobile Dashboard: Vinicius Rodrigues