

Manual de Instruções

ETANÓIS
Atvos

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
01/12/23	Etanóis	0.1.0	Criação do documento
02/12/23	Etanóis	0.1.1	Preenchimento das seções 1 e 2
04/12/23	Etanóis	0.1.2	Preenchimento das seções 3, 4 e 5
06/12/23	Etanóis	0.1.3	Preenchimento da seção 6 e revisão do documento
10/12/23	Etanóis	1.0.0	Revisão geral e correção do documento

Índice

1.Introdução	4
1.1. Solução	4
Benefícios da solução	4
1.2. Arquitetura da Solução	5
2. Componentes e Recursos	6
2.1. Componentes de hardware	6
2.2. Componentes externos	10
2.3. Requisitos de conectividade	13
3. Guia de Montagem	13
4. Guia de Instalação	15
5. Guia de Configuração	15
6. Guia de Operação	16
7. Troubleshooting	17

1. Introdução

1.1. Solução

A equipe Etanóis propõe uma solução abrangente baseada em IoT para a Atvos, visando monitorar equipamentos e insumos agrícolas.

A proposta inclui rastreamento em campo, controle nos almoxarifados, revisão logística, eliminação de processos manuais e abertura/fechamento digital de ordens de serviço.

A solução utiliza uma plataforma na nuvem, tags de rastreamento e dispositivos IoT.

Justificativas destacam a abordagem completa, rastreamento em tempo real, prevenção de desvios, eficiência operacional, histórico para decisões estratégicas, sustentabilidade, diferenciação competitiva e integração com outros sistemas.

Em suma, a proposta busca melhorar a eficiência, reduzir perdas e fortalecer a posição da Atvos no setor agrícola.

Benefícios:

- Melhor rastreabilidade de equipamentos, peças e insumos.
- Redução significativa de desvios e roubos.
- Aumento da eficiência operacional.
- Eliminação de processos manuais demorados.
- Facilitação da abertura e fechamento digital de Ordens de Serviço.
- Dados históricos para tomada de decisões estratégicas.
- Contribuição para a sustentabilidade ao evitar desperdícios e reduzir a pegada de carbono.
- Fortalecimento da posição competitiva da Atvos no mercado.

1.2 Arquitetura da Solução

A proposta de solução para a empresa parceira busca revolucionar a gestão de insumos e equipamentos através da integração da tecnologia IoT (Internet das Coisas).

A arquitetura é composta por diversos componentes interligados, abrangendo desde dispositivos físicos no campo até a interface do usuário, passando por servidores em nuvem e servidores de aplicação.

Cada componente desempenha uma função específica para atender aos requisitos estabelecidos, sejam eles funcionais ou não funcionais. A representação visual desta arquitetura pode ser observada na ao lado.

Para o bom funcionamento do protótipo e solução WEB é recomendável a utilização de ambiente Linux nativo ou máquina virtual, para o manual e passos de instalação serão pensados para ambientes Unix.

- IDE (Ambiente de Desenvolvimento Integrado): Uma ferramenta completa que combina editor de código, depurador e outras funcionalidades para simplificar e otimizar o processo de desenvolvimento de software, proporcionando um ambiente integrado e eficiente (Arduino IDE).

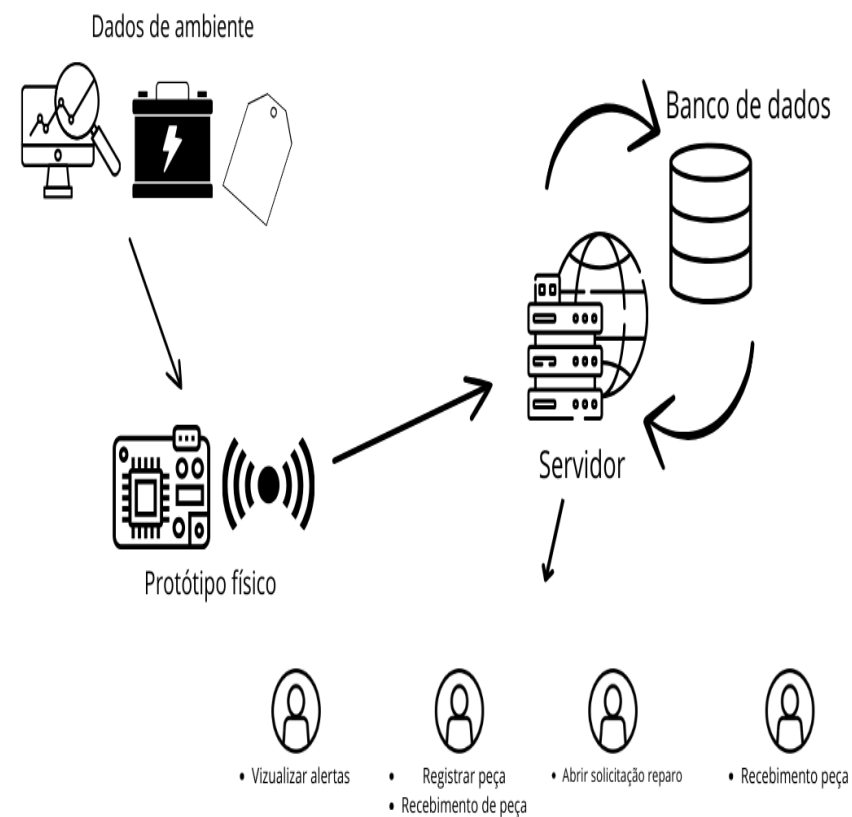
- Backend (Lado do Servidor): Refere-se à parte do sistema que lida com a lógica de negócios, processa dados e executa operações no servidor. Geralmente, é responsável pela manipulação dos dados, interações com o banco de dados e outras tarefas relacionadas ao processamento no lado do servidor. O backend da solução foi realizado com *Express*, siga os seguintes passos para executar o backend:
 - Entre no repositório e na pasta “backend”
 - Execute o comando “npm i”
 - Em seguida, execute o comando “npm start”

- Frontend (Interface do Usuário): É a camada visível para o usuário final, responsável por exibir e coletar informações. O frontend visa proporcionar uma experiência interativa e amigável ao usuário no navegador, o frontend da presente solução foi feito em *React*, para executar o frontend, siga os seguintes passos:

- Abrir o repositório na pasta frontend.
- Executar o comando “yarn install”.
- E por fim, executar o comando “yarn dev”.

- Banco de Dados: Um sistema que armazena e gerencia dados de forma organizada e estruturada. Os bancos de dados são essenciais para armazenar informações de aplicativos, permitindo consultas eficientes, atualizações e recuperação de dados de maneira rápida e segura. O banco de dados do projeto foi feito com *Postgress*.



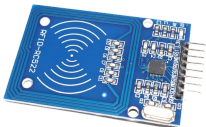

- ElephantSQL: Um serviço de hospedagem de banco de dados PostgreSQL na nuvem, oferecendo uma solução pronta para uso e gerenciamento eficiente de bancos de dados relacionais, com foco na escalabilidade e confiabilidade (indicado expressamente para o ambiente de desenvolvimento, para produção outros serviços em nuvem deve ser utilizados).







2. Componentes e Recursos

2.1. Componentes de hardware



A seguir estão listados os componentes de hardware necessários para a montagem do dispositivo IoT.

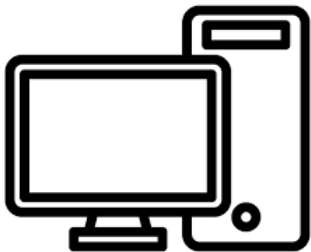

Representação	Componente	Descrição	Tipo: entrada/saída/ atuador
	ESP32 WROOM 32U	O microcontrolador ESP32-S3 desempenha um papel crucial no sistema, estabelecendo a conexão vital com os sensores. Ele capta dados com alta velocidade de transmissão e os armazena no servidor e no banco de dados. Equipado com a maioria dos principais componentes de um computador, o microcontrolador atua como o cérebro do sistema, controlando todas as operações. Sua utilização garante a implementação de um ambiente seguro e o processamento eficiente das informações.	Entrada e saída
	Antena Articulada	A antena articulada é um componente essencial do sistema, projetada para facilitar a comunicação sem fio entre o microcontrolador ESP32-S3 e o servidor ou dashboard.	Entrada
	Leitor de Etiquetas RFID-MFRC522	O sensor RFID (Radio-Frequency Identification) é um dispositivo que utiliza tecnologia de radiofrequência para identificação e rastreamento de objetos equipados com etiquetas RFID. No contexto do sistema IoT, o sensor RFID é empregado para captar informações de ativos equipados com tags RFID, possibilitando a identificação única e o rastreamento eficiente desses elementos.	Entrada
	Tags RFID	A tag RFID (Radio-Frequency Identification) é um pequeno dispositivo que contém um chip eletrônico e uma antena, sendo utilizado para identificação única de objetos ou ativos.	Saída

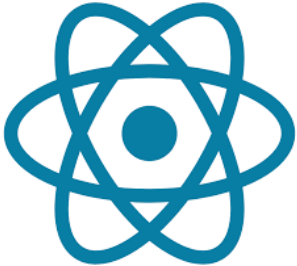

	LED 5 Milímetros	<p>O LED (Light Emitting Diode), ou Diodo Emissor de Luz, é um componente eletrônico que emite luz quando uma corrente elétrica passa por ele. No contexto do sistema IoT, os LEDs são utilizados como indicadores visuais para fornecer feedback sobre o status do sistema ou eventos específicos.</p>	Saída
	Buzzer TMB12A05	<p>O Buzzer é um dispositivo acústico que converte sinais elétricos em ondas sonoras audíveis. No contexto do sistema IoT, o Buzzer é utilizado para gerar alertas sonoros em resposta a eventos específicos.</p>	Saída
	Botão	<p>O botão, no contexto do sistema IoT, é um componente de entrada física que permite a interação do usuário com o sistema. Ele funciona como um interruptor, geralmente sendo pressionado para iniciar ou interromper determinadas operações. No contexto específico do projeto, o botão pode ser utilizado para acionar solicitações, iniciar processos ou realizar ações específicas no sistema.</p>	Entrada
	Jumpers	<p>Responsáveis por realizar conexões entre os componentes de hardware.</p>	n/a



2.2 Componentes externos

A seguir estão listados os componentes externos que fazem parte da solução.

Representação	Componente	Descrição	Tipo: entrada/saída/ atuador
	Interface Web	Utilizando a interface WEB, o parceiro terá acesso a uma visualização da situação atual de cada dispositivo. Ela receberá informações atualizadas do banco de dados	Saída
	Roteador Wi-fi	O que fará a conexão do dispositivo com a internet, dessa forma, permitindo que ele envie dados para o banco de dados.	Atuador

	<p>Computador</p>	<p>O computador é responsável por configurar os dispositivos e fazê-los funcionar.</p>	<p>Atuador</p>
	<p>Platform IO</p>	<p>O PlatformIO é um ecossistema de código aberto que simplifica o desenvolvimento de firmware embarcado. Com suporte multiplataforma, integração a IDEs, gerenciamento de bibliotecas e automação de compilação, oferece uma solução unificada e eficiente.</p>	<p>Atuador</p>

	<p>React</p>	<p>React é uma biblioteca JavaScript para construir interfaces de usuário interativas, destacando-se pela criação eficiente de componentes reutilizáveis e atualização automática da interface em resposta a alterações de dados.</p>	<p>Atuador</p>
	<p>JavaScript</p>	<p>Linguagem de programação utilizada para executar operações básicas na interface web, como a navegação entre telas e a obtenção de dados do banco de dados.</p>	<p>Atuador</p>

	<p>ElephantSQL</p>	<p>ElephantSQL é um serviço em nuvem para bancos de dados PostgreSQL, oferecendo solução gerenciada. Ideal para projetos que demandam conveniência, escalabilidade e robustez sem preocupações de infraestrutura.</p>	<p>Atuador</p>
	<p>VS Code</p>	<p>O VSCode, editor de código da Microsoft, é leve, modular e suporta diversas linguagens. Reconhecido por sua interface amigável e ampla extensibilidade.</p>	<p>Atuador</p>

2.3. Requisitos de conectividade

Os requisitos de conectividade envolvem as redes, protocolos de rede e especificações do back-end para o funcionamento dos dispositivos. Na primeira conexão é possível citar a energia, que precisa estar conectada ao ESP 32 para que ele possa distribuir a corrente para a protoboard e assim, conseguir ligar todos os componentes do microcontrolador.

Por outro lado, existe a conexão da rede Wifi, a qual o ESP32 se conecta para conseguir mandar as informações coletadas para o banco de dados e para que toda a interface web possa funcionar com êxito.

3. Guia de Montagem

Este guia tem como objetivo fornecer instruções detalhadas e passo-a-passo para a montagem física dos dispositivos IoT que compõem a solução proposta.

Abordaremos os principais componentes mencionados na seção 2, utilizando diagramas, fotografias e descrições precisas para garantir uma montagem eficiente e sem contratempos.

Ao seguir este guia, você terá uma visão clara do processo de montagem, incluindo as ferramentas necessárias e os detalhes essenciais para a implementação bem-sucedida da solução.

Materiais Necessários:

- ESP32
- Antena Articulada
- Protoboard
- Botão
- Buzzer
- Leitor de RFID e Cartões RFID (3 unidades)
- LEDs (3 unidades)
- Jumpers (45 unidades)
- Cabo Micro USB

Ferramentas Necessárias:

- Cabo Micro USB (dados e alimentação)
- Computador
- Kit de ferramentas básico

Passos:

- Preparação do ESP32:
 - Conecte o ESP32 à case, observando a orientação correta dos pinos e sua disposição.
 - Conecte a antena articulada ao ESP32, garantindo uma conexão firme e com maior potência.

- Instalação do Leitor de RFID:

- Conecte o leitor de RFID à case, seguindo as instruções do fabricante.
- Conecte o leitor de RFID ao ESP32, utilizando as portas designadas pela tabela abaixo.

RFID 1:

PORTA	SDA /CS	SCK	MOS I	MIS O	RQ	GND	RST	3.3V
PINO	21	18	23	19	N/A	GND	22	3.3V

RFID 2:

PORTA	SDA /CS	SCK	MOS I	MIS O	RQ	GND	RST	3.3V
PINO	13	18	23	19	N/A	GND	32	3.3V

- Conexão do Buzzer:

- Conecte o botão e o buzzer à case, garantindo uma disposição organizada, respeitando a pinagem pré-estabelecida.

- Conecte o botão e o buzzer ao ESP32, utilizando as portas especificadas (por exemplo, porta Y para o botão e porta Z para o buzzer).

PORTA	POSITIVO	NEGATIVO
PINO	33	GND

- Configuração dos LEDs:

- Conecte os LEDs à case, prestando atenção à polaridade [Tutorial como saber a polaridade do LED](#).
- Conecte os LEDs ao ESP32, utilizando as portas apropriadas

PORTA	NEGATIVO (AMBOS)	AZUL POSITIVO	VERDE POSITIVO
PINO	resistor de 100Ω	27	14

- Conexão dos Botões:

- Conecte o botão à case, garantindo uma disposição organizada, respeitando a pinagem pré-estabelecida.

- Conecte o botão ao ESP32, utilizando as portas especificadas

Botão 1:

PORTA	NEGATIVO	POSITIVO
PINO	GND	26

Botão 2:

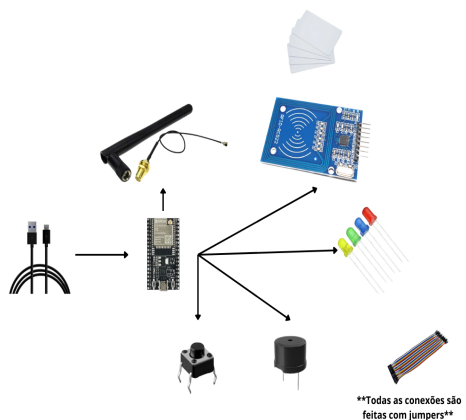
PORTA	NEGATIVO	POSITIVO
PINO	GND	25

- Incorporação dos Cartões RFID:
 - Aproxime os cartões RFID no leitor conforme necessário, seguindo as orientações do fabricante.
- Verificação das Conexões:
 - Certifique-se de que todas as conexões estão firmes na protoboard case.
 - Conecte o cabo Micro USB ao ESP32 e, em seguida, ao computador.

- Programação e Teste:
 - Programe o ESP32 com o código correspondente, considerando as portas específicas mencionadas anteriormente.
 - Realize testes para verificar o funcionamento adequado do protótipo.

Observações:

- Siga a orientação correta dos pinos nos componentes.
- Consulte o diagrama de conexões fornecido na seção 3 para referência.
- Em caso de problemas, verifique as conexões e revise o código utilizado.



Este guia tem como propósito orientar o processo de instalação dos dispositivos IoT no ambiente físico designado.

Ao seguir este documento, você obterá instruções detalhadas sobre como conectar os dispositivos à rede, respeitando limites, alcances e propriedades específicas para garantir um desempenho ótimo.

Além disso, serão abordadas as etapas para instalação de softwares nos dispositivos, proporcionando uma implementação suave e funcional da solução proposta.

Utilize as informações, imagens e passos apresentados aqui para assegurar uma instalação eficiente e adequada às necessidades do seu contexto.

4. Guia de Instalação

- Posicionamento: Identifique áreas estratégicas para a instalação. Considere limitações físicas e obstáculos potenciais.
- Energia: Conecte o ESP32 à energia usando um cabo Micro USB.

- Wi-Fi: Configure o ESP32 para se conectar à rede Wi-Fi específica.
- Verificação: Teste a conexão Wi-Fi para garantir a conectividade adequada.

- Montagem: Fixe o ESP32 em suportes adequados. Posicione os leitores RFID de forma acessível.
- Testes Preliminares: Verifique o reconhecimento de tags RFID para garantir o funcionamento inicial.
- Configuração do Software: Instale o firmware necessário no ESP32. Configure os parâmetros do sistema conforme as necessidades específicas.
- Verificação: Realize testes nos LEDs, botões e buzzers para garantir seu funcionamento correto.

Observações:

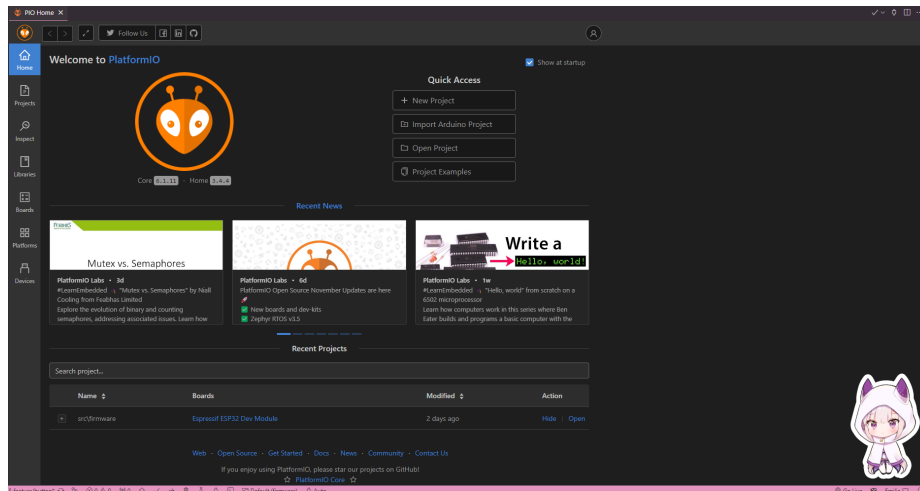
- Priorize a segurança física do sistema.
- Proteja o equipamento contra condições adversas.
- Evite exposição não autorizada aos componentes.
- Revise periodicamente a conexão Wi-Fi e a integridade do sistema.
- Mantenha registros detalhados de instalação e configuração para referência futura.

IDE (Ambiente de execução):

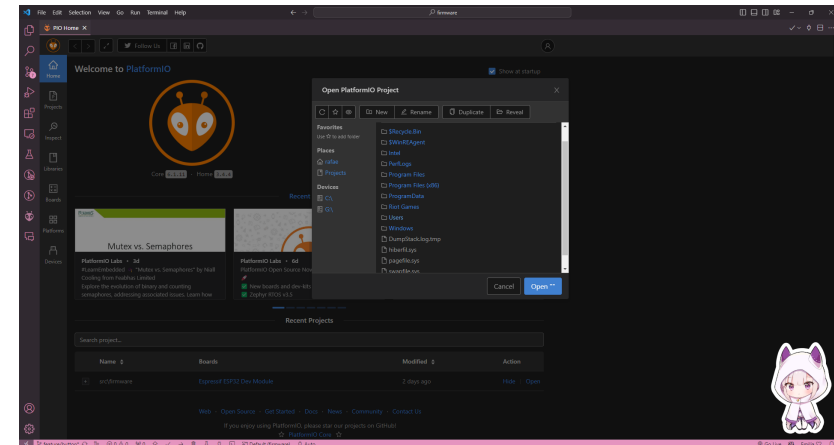
O PlatformIO no VSCode é uma plataforma de desenvolvimento integrado que simplifica a programação de microcontroladores, incluindo o ESP32. Ao oferecer um ambiente amigável com editor de código, compilador e monitor serial, o PlatformIO permite que desenvolvedores escrevam códigos em C/C++, compilem-nos e os enviem diretamente para o ESP32 ao conectar a placa ao computador. Essa funcionalidade de upload facilita a implementação de projetos eletrônicos, permitindo a interação entre o ESP32 e os componentes conectados, tornando o processo de desenvolvimento mais acessível a uma ampla variedade de entusiastas e profissionais, agora com a conveniência do ambiente do Visual Studio Code.

- Faça download do vscode, especificado no site oficial [Download oficial vscode](#)
- Faça download da extensão do PlatformIO, um suite de ferramentas para desenvolvimentos de embarcados [Download oficial PlatformIO](#)

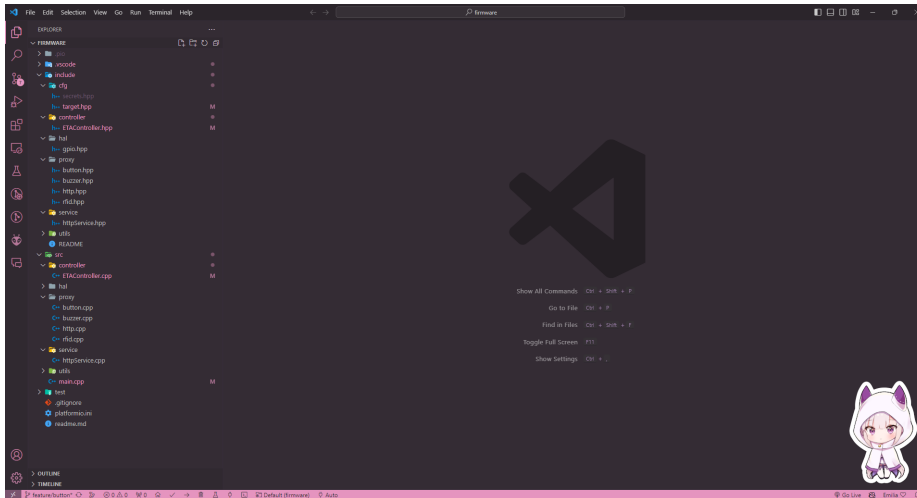
Após o Download da plataforma, um ícone de formiga aparecerá nas suas extensões do vscode, clique no ícone para ser redirecionado para esta tela:



Clique em “import project” e selecione a pasta com o arquivo de firmware

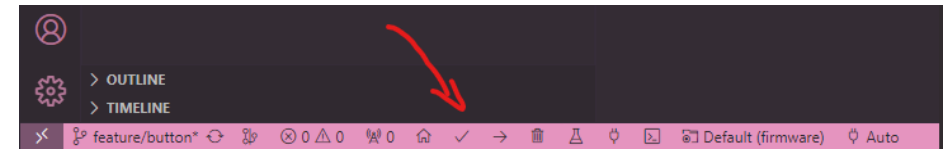


Agora com o projeto aberto, indicado pelos arquivos no seu explorador de arquivos à esquerda:

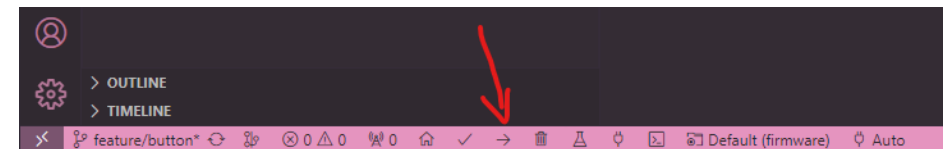


Atente-se, para o próximo passo é vital que o circuito esteja conectado fisicamente ao computador via micro USB para dados e energia.

Clique no ícone de flecha no rodapé do aplicativo, esta função pré-compila o código a fim de buscar quaisquer falhas:



Após todas estas etapas finalizadas basta apertar o “run” para realizar o upload do código em c++ para o protótipo (mantenha o esp32 conectado até o fim do “run”):



Feito todos os passos corretamente o seu protótipo estará completamente funcional.

Aplicação WEB(BACKEND):

- Para instalar o Node.js no Linux, siga os passos de instalação de acordo com a documentação oficial do Node.js em [Documentação Oficial NODE](#).
- Após a instalação bem sucedida do Node.js é necessário a instalação do npm, gerenciador pacotes do Node para poder baixar dependências externas, no seu terminal digite o seguinte comando:

```
sudo apt install npm -y
```

- Após a instalação dessas duas ferramentas, basta navegar para a pasta “backend/”, na raiz do projeto, nela estão contidos todos os arquivos referentes ao servidor da aplicação.
- Para ambientes de testes é necessário a criação de um servidor em nuvem para este propósito será utilizado a plataforma [ElephantSQL](#), acesse o link e siga o passo a passo. [ElephantSQL](#), salve o link gerado na área de transferência da sua máquina. Execute o seguinte comando: (para executar este comando certifique-se de estar no diretório “/grupo4/src/backend”)

```
echo "ELEPHANTSQL=seu+link+aqui" > .env
```

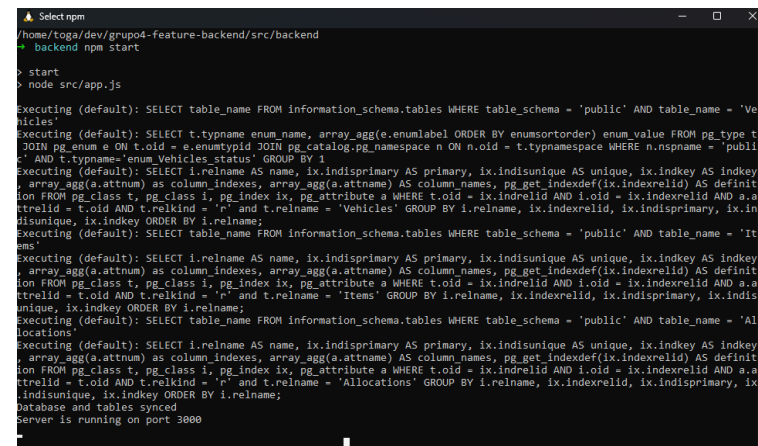
- Instale as dependências do projeto, ainda dentro de (“/grupo4/src/backend”), execute:

```
npm install
```

- Execute o backend localmente:

```
npm start
```

Caso todos os passos tenham sido seguidos corretamente o seu terminal estará semelhante ao mostrado ao lado:



```

Select npm
/home/toga/dev/grupo4-feature-backend/src/backend
+ backend npm start
> start
> node src/app.js

Executing (default): SELECT table_name FROM information_schema.tables WHERE table_schema = 'public' AND table_name = 'Vehicles'
Executing (default): SELECT t.typname enum_name, array_agg(e.enumlabel ORDER BY enumsortorder) enum_value FROM pg_type t JOIN pg_enum e ON t.oid = e.enumtypid JOIN pg_catalog.pg_namespace n ON n.oid = t.typnamespace WHERE n.nspname = 'public' AND t.typname = 'enum Vehicles status' GROUP BY 1
Executing (default): SELECT i.relname AS name, ix.indisprimary AS primary, ix.indisunique AS unique, ix.indkey AS indkey, array_agg(a.attname) AS column_indexes, array_agg(a.attname) AS column_names, pg_get_indexdef(ix.indexrelid) AS definition FROM pg_class t, pg_class i, pg_index ix, pg_attribute a WHERE t.oid = ix.indexrelid AND i.oid = ix.indexrelid AND a.attrelid = t.oid AND t.relkind = 'r' and t.relname = 'Vehicles' GROUP BY i.relname, ix.indexrelid, ix.indisprimary, ix.indisunique, ix.indkey ORDER BY i.relname;
Executing (default): SELECT table_name FROM information_schema.tables WHERE table_schema = 'public' AND table_name = 'Items'
Executing (default): SELECT i.relname AS name, ix.indisprimary AS primary, ix.indisunique AS unique, ix.indkey AS indkey, array_agg(a.attname) AS column_indexes, array_agg(a.attname) AS column_names, pg_get_indexdef(ix.indexrelid) AS definition FROM pg_class t, pg_class i, pg_index ix, pg_attribute a WHERE t.oid = ix.indexrelid AND i.oid = ix.indexrelid AND a.attrelid = t.oid AND t.relkind = 'r' and t.relname = 'Items' GROUP BY i.relname, ix.indexrelid, ix.indisprimary, ix.indisunique, ix.indkey ORDER BY i.relname;
Executing (default): SELECT table_name FROM information_schema.tables WHERE table_schema = 'public' AND table_name = 'Allocations'
Executing (default): SELECT i.relname AS name, ix.indisprimary AS primary, ix.indisunique AS unique, ix.indkey AS indkey, array_agg(a.attname) AS column_indexes, array_agg(a.attname) AS column_names, pg_get_indexdef(ix.indexrelid) AS definition FROM pg_class t, pg_class i, pg_index ix, pg_attribute a WHERE t.oid = ix.indexrelid AND i.oid = ix.indexrelid AND a.attrelid = t.oid AND t.relkind = 'r' and t.relname = 'Allocations' GROUP BY i.relname, ix.indexrelid, ix.indisprimary, ix.indisunique, ix.indkey ORDER BY i.relname;
Database and tables synced
Server is running on port 3000
  
```

Aplicação WEB(Frontend):

Para o prosseguimento bem sucedido desta seção é crucial que a etapa Aplicação(Backend) tenha sido executada com êxito. Pois para o prosseguimento da instalação das dependências supõe-se que o interpretador de código Node.js e o gerenciador de pacotes npm estejam instalados corretamente. Quais dúvidas consulte [Documentação oficial Node.js](#).

- Execute o seguinte comando no seu terminal para instalar um outro gerenciador de pacotes “yarn”, que foi escolhido para o desenvolvimento da plataforma.

```
npm install -global yarn
```

- Agora execute o seguinte comando no seu terminal para instalar as dependências necessárias para o funcionamento do projeto

```
yarn install
```

- Agora basta apenas executar o projeto com o comando:

```
yarn dev
```

Caso a instalação e principalmente a execução do projeto for bem sucedida o seu terminal apresentará a seguinte mensagem:

```

yarn
→ FrontendEtanois git:(feature/frontend) yarn dev
yarn run v1.22.21
$ vite

VITE v5.0.3 ready in 354 ms
→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help

```

5. Guia de Configuração

- Acesso ao Servidor Embarcado:

- Insira o endereço IP do ESP32 no navegador, por exemplo, <http://192.168.1.1>.
- Faça login utilizando as credenciais padrão ou as personalizadas previamente configuradas no dispositivo.

- Configuração de Rede:
 - Navegue para a seção "Configurações" no painel do servidor embarcado.
 - Insira detalhes precisos da rede, incluindo SSID (nome da rede) e senha.
 - Se necessário, configure configurações avançadas, como endereço IP estático.
- Registro de Dispositivos:
 - Acesse a funcionalidade de registro de dispositivos no painel.
 - Adicione tags RFID ao sistema, escaneando ou inserindo manualmente.
 - Associe cada tag a ativos específicos, proporcionando uma associação clara e organizada.

- Parâmetros do Sistema:

- Ajuste os intervalos de leitura RFID para otimizar o desempenho do sistema.

- Configure alertas para notificações específicas, definindo limites ou condições que acionaram os alertas.

- Verificação de Configuração:

- Realize um teste de reconhecimento de tags RFID recém-registradas para garantir a funcionalidade adequada.

- Confirme que os alertas estão sendo gerados corretamente, verificando a caixa de histórico de alertas no painel.

- Backup de Configuração:

- Salve todas as configurações realizadas no servidor embarcado.

- Mantenha uma cópia de segurança das configurações em um local seguro, como um servidor externo ou um dispositivo de armazenamento.

- *Observações:

Certifique-se de que o ESP32 e o dispositivo de acesso estejam conectados à mesma rede Wi-Fi para garantir a comunicação adequada.

- Siga rigorosamente as orientações de segurança fornecidas no manual do sistema.

- Faça anotações detalhadas de todas as configurações realizadas, incluindo datas e detalhes específicos.

- Realize testes periódicos para garantir o desempenho contínuo do sistema, especialmente após atualizações ou alterações significativas.

6. Guia de Operação

- Tela Inicial:

- Visualize ativos e seus estados atuais de forma clara e organizada.

- Acesse opções de leitura e ações diretamente a partir da tela inicial, proporcionando uma experiência de usuário eficiente.

- Leitura RFID:

- Para realizar a leitura RFID, aproxime a tag do leitor de forma cuidadosa.

- Observe atentamente a confirmação na tela para garantir que a leitura tenha sido bem-sucedida.

- Verifique atualizações do ativo associado à tag, visualizando detalhes atualizados imediatamente após a leitura.

- Monitoramento em Tempo Real:

- Acesse gráficos detalhados de leituras, proporcionando insights visuais sobre o comportamento dos ativos.

- Obtenha dados em tempo real para tomada de decisões informada e eficaz.

- Ações com Atuadores:

- Ative LEDs e o Buzzer diretamente pela interface, facilitando o controle remoto dos dispositivos.

- Confirme mudanças nos dispositivos monitorados, garantindo feedback visual e sonoro de ações realizadas.

- Configurações Adicionais:

- Ajuste os intervalos de leitura conforme as necessidades específicas do sistema.

- Configure alertas personalizados para notificações imediatas de eventos críticos.

- Imprecisões de Localização:

- Esteja ciente da possibilidade de desvios na localização dos ativos.

- Utilize informações adicionais, como dados de sensores ambientais, para compensar possíveis imprecisões.

- Observações:

- Mantenha atualizações frequentes no sistema para garantir o desempenho contínuo.

- Esteja atento às notificações geradas pelo sistema, priorizando ações imediatas quando necessário.

- Relate qualquer anomalia detectada imediatamente ao responsável pelo sistema ou equipe técnica.

- Consulte guias de solução de problemas fornecidos para abordar questões com eficácia.

- Mantenha as tags RFID em bom estado, realizando inspeções regulares e substituindo tags danificadas.

- Faça leituras em ambientes estáveis para garantir resultados precisos e consistentes.

Durante a operação dos dispositivos IoT, é crucial estar ciente de que a precisão da localização pode ser afetada por variáveis ambientais e interferências.

Para contornar eventuais imprecisões, sugerimos as seguintes abordagens:

Verificação de Obstáculos:

Antes da instalação, realize uma verificação minuciosa do ambiente. Obstáculos físicos, como paredes densas e grandes estruturas metálicas, podem interferir na precisão da localização. Evite instalar os dispositivos em locais com obstruções significativas.

Ajustes na Instalação:

Faça ajustes na disposição dos dispositivos, se necessário, para otimizar a linha de visão e minimizar interferências.

Reposicionar os dispositivos em áreas mais abertas pode melhorar a precisão da localização.

Atualizações de Firmware:

Mantenha os dispositivos com o firmware mais recente.

Atualizações de firmware podem incluir melhorias nos algoritmos de localização e correções para aprimorar a precisão.

Calibração Ocasional:

Realize calibrações periódicas, especialmente se houver mudanças significativas no ambiente.

A calibração pode ajudar a ajustar os parâmetros de localização para garantir resultados mais precisos.

Uso de Pontos de Referência:

Em situações onde a precisão é crucial, utilize pontos de referência adicionais para validar a localização.

Estabelecer pontos fixos conhecidos pode ajudar a corrigir imprecisões temporárias.

Lembramos que, embora os dispositivos se esforcem para fornecer localizações precisas, condições externas podem impactar o desempenho.

7. Troubleshooting

Liste as situações de falha mais comuns da sua solução (tais como falta de conectividade, falta de bateria, componente inoperante etc.) e indique ações para solução desses problemas.

#	Problema	Possível solução
1	Falta de Conectividade Wi-Fi	<ul style="list-style-type: none"> - Verifique a configuração Wi-Fi no ESP32. - Verifique se há interferências na rede.. - Atualize as configurações de rede conforme necessário.
2	Falta de Energia no Dispositivo	<ul style="list-style-type: none"> - Substitua ou recarregue as baterias. - Verifique a conexão do cabo de alimentação. - Certifique-se de que a fonte de energia está funcionando.
3	Componente Inoperante (LED, Buzzer, RFID, etc.)	<ul style="list-style-type: none"> - Verifique a conexão física do componente ao ESP32. - Substitua o componente

		se estiver danificado. <ul style="list-style-type: none"> - Atualize o firmware do ESP32, se aplicável. - Verifique as configurações de pinagem no código.
4	Erros na Leitura RFID	<ul style="list-style-type: none"> - Garanta que as tags RFID estão em boas condições. - Ajuste a distância entre a tag e o leitor RFID. - Considere calibrar o leitor para ambientes específicos. - Verifique a integridade do código de leitura no ESP32.
5	Problemas de Compatibilidade com Displays	<ul style="list-style-type: none"> - Atualize os drivers e bibliotecas para o display. - Verifique a integridade física do cabo de conexão.