

# Manual do Usuário

**PIRELLI**  
**Pirelli**

## Controle do Documento

### Histórico de revisões

Data	Autor	Versão	Resumo da atividade
06/06/2023	Henrique Godoy	1.0	Criação do Manual do usuário

# Índice

1. Componentes e Recursos (sprint 4)	3
1.1. Componentes externos	3
1.2. Requisitos de conectividade	3
2. Guia de Montagem (sprint 4)	4
3. Guia de Instalação (sprint 4)	5
4. Guia de Configuração (sprint 4)	5
5. Guia de Operação (sprint 4)	6
6. Troubleshooting (sprint 4)	7

# 1. Componentes e Recursos (sprint 4)

## 1.1. Componentes externos

Liste aqui componentes como computadores, tablets e/ou celulares que deverão fazer parte da sua solução, bem como eventuais serviços em nuvem, softwares de edição de código ou outras aplicações utilizadas.

Para garantir a utilização eficiente da solução, é altamente recomendado fazer uso dos seguintes Componentes/Serviços:

Nome	Função
ESP32-S3-WROOM-1	Responsável por executar o código da solução
Laptop ou Desktop	Responsável pela visualização da aplicação web
Jumper Macho-Macho e Fêmea -Fêmea	Responsável pela conexão dos componentes através do circuito eletrônico

Cabo Usb-C	Responsável pela conexão física entre microcontrolador e notebook
Resistor 1K e 10K	Responsável por evitar queima dos componentes dentro do circuito eletrônico
Protoboard 400 furos ou mais	Responsável pela organização do circuito eletrônico
LCD	Responsável pela exibição dos resultados derivados do microcontrolador
Antena WiFi	Responsável por transmitir e receber os sinais de WiFi de outros dispositivos
Banco de Dados MongoDB	Responsável pelo armazenamento da aplicação
Github	Plataforma responsável pela colaboração de código fonte
CodeSandbox	Responsável por rodar o código na nuvem da aplicação
React	Linguagem responsável pelo código da interface front-end

C++	Linguagem responsável pelo código executado no microcontrolador
Arduino IDE 1.8+	Responsável pela escrita do código executado no microcontrolador
Visual Studio Code 1.73+	Responsável pela escrita do código da interface backend e frontend

	possibilidade de conexão a redes sem fio
HTTPS	Protocolo que realiza a conexão entre navegador e servidor
MAC	Protocolo que permite a conexão de múltiplos dispositivos a uma rede compartilhada
TCP/IP	Protocolo de controle de transmissão

## 1.2. Requisitos de conectividade

Liste aqui as redes, protocolos de rede e eventuais especificações de back-end, necessários para o funcionamento dos dispositivos.

Exemplo de uso de imagem em coluna única:

A fim de ser possível a comunicação entre as diferentes camadas da arquitetura da solução e o backend, são imprescindíveis os requisitos a seguir

Conectividade	Descrição
MQTT	Protocolo que realiza a conexão entre ESP32 e Backend
WiFi	Tecnologia que traz a

## 2. Guia de Montagem

(sprint 4)

Descreva passo-a-passo como montar fisicamente os dispositivos IoT de sua solução, mencionando os componentes da seção 2.

Utilize diagramas e fotografias para ilustrar o processo de montagem (você pode ser bem didático e explicar até quais as ferramentas necessárias). Utilize exatamente os mesmos nomes/modelos de componentes listados na seção 2.

Para o uso adequado da solução a montagem deve ser feita de acordo com estas instruções tanto para a montagem do protótipo de cadastro do manutentor, quanto para o protótipo acoplado ao tablet

### Guia de Montagem Protótipo Manutentor

Checagem dos componentes **necessários**:

Montagem esp32 na protoboard

Montagem **leitor** rfid na protoboard

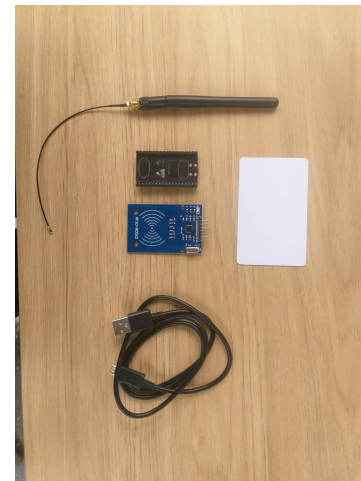
Montagem leitor lcd

Conexão com jumpers

### Guia de Montagem Protótipo acoplado ao tablet

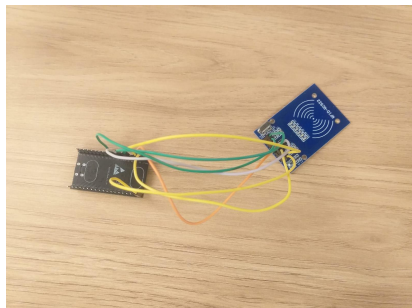
Passo 1:

Checagem dos componentes necessários, organizar a antena, o esp32 e leitor RFID



Passo 2:

Conexão da antena no esp32



Conexão do esp32 com o leitor rfid seguindo este padrão:

SDA → 21

SCK → 18

MOSI → 23

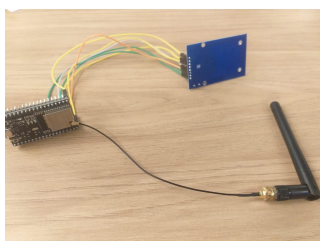
MISO → 19

RST → 22

GND → GND

RST → 22

Passo 3:



# 3. Guia de Instalação

(sprint 4)

Descreva passo-a-passo como instalar os dispositivos IoT no espaço físico adequado, conectando-os à rede, de acordo com o que foi levantado com seu parceiro de negócios.

Não deixe de especificar propriedades, limites e alcances dos dispositivos em relação ao espaço destinado.

Especifique também como instalar softwares nos dispositivos.

Utilize fotografias, prints de tela e/ou desenhos técnicos para ilustrar o processo de instalação.

A solução desenvolvida pelo grupo requisita o uso de dois tipos de esp-32, o primeiro será “protótipo acoplado ao tablet”, que chamaremos de ESP(T), enquanto que o segundo será o “protótipo cadastral”, que servirá para o registro dos colaboradores na aplicação WEB, nomearemos esse como ESP(C).

## Instalação softwares:

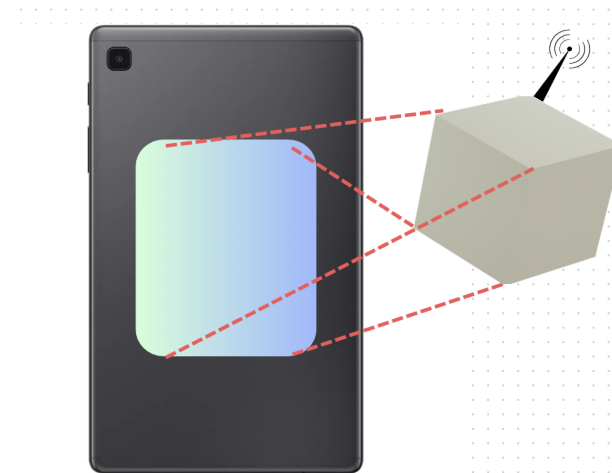
para a instalação dos softwares e conexão com a rede é necessário seguir o seguinte passo a passo:

- Instalação da arduino IDE

- Utilização do código desenvolvido para a solução, que está disponível no github
- Alteração da rede e senha de acordo com wi-fi da fábrica
- Conectar através de um cabo o ESP no computador que estará utilizando o código (de acordo com o tipo de esp, tendo em vista que está disponível o código de ESP(T) e o de ESP(C)) do grupo
- Compilar o código

Após esse pequeno procedimento descrito a solução já estará disponível com o software desenvolvido pelo grupo.

**Instalação ESP(T):** Para a instalação do ESP(T) no espaço físico, basta acoplar a caixa, que irá proteger o ESP, na parte traseira dos tablets.

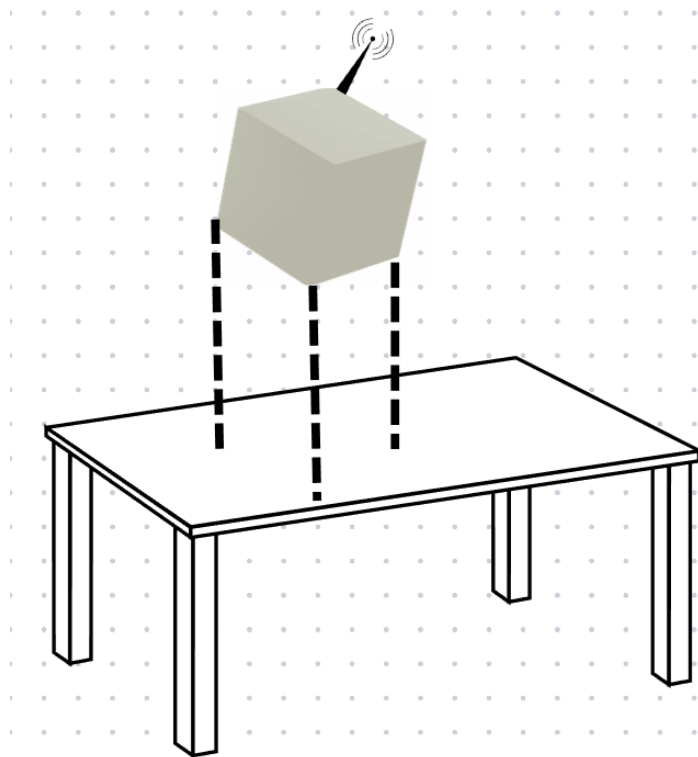


Após esse processo, a instalação do ESP(T) estará concluída.



### Instalação ESP(C):

Para a instalação do ESP(C) no espaço físico, pode-se colocá-lo em qualquer superfície plana, após isso conectá-lo ao computador, em seguida seguir o procedimento de uso do ESP(C).



## 4. Guia de Configuração (sprint 4)

Para conectar o rastreador à solução web é um requisito ter seguido todos os passos da seção 3 com sucesso. Uma vez feito isso, ligue o aparelho e ele irá automaticamente se conectar a aplicação web, aparecendo um novo tablet na página do mapa, como também na seção dos tablets.

Colocar uma imagem aqui do front com um novo tablet

## 5. Guia de Operação

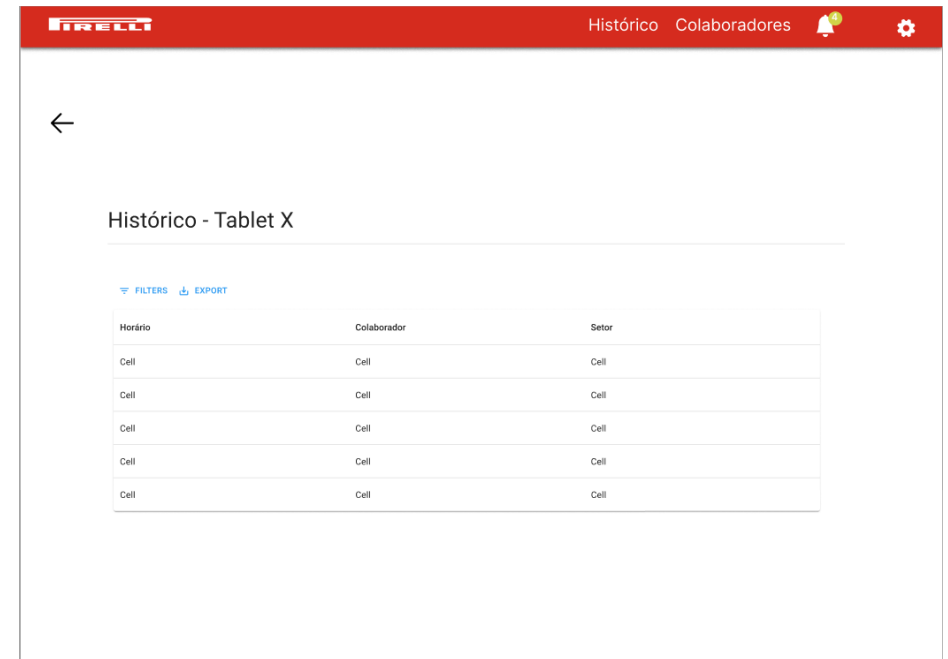
(sprint 4)

Descreva os fluxos de operação entre interface e dispositivos IoT. Indique o funcionamento das telas, como fazer leituras dos dados dos sensores, como disparar ações através dos atuadores, como reconhecer estados do sistema.

Indique também informações relacionadas à imprecisão das eventuais localizações, e como o usuário deve contornar tais situações.

Utilize fotografias, prints de tela e/ou desenhos técnicos para ilustrar os processos de operação.

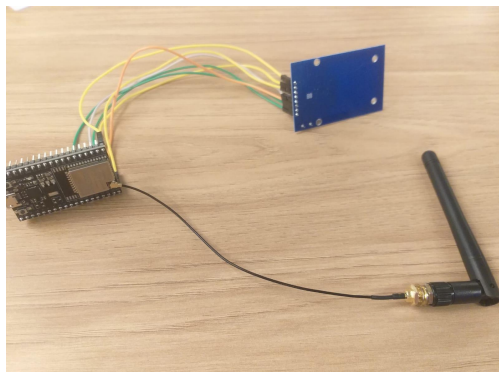
Para entender o fluxo de operações da nossa solução, iremos seguir uma linha cronológica para facilitar a visualização. O nosso dispositivo IOT acoplado ao Tablet, gera informações a cada intervalo X de tempo, criando um conjunto de informações como localização atual, potência do sinal dos 3 roteadores mais próximos, endereço mac dos 3 roteadores mais próximos. Essas informações são enviadas ao nosso banco de dados pela conexão com a internet da nossa peça Esp32, localizada no dispositivo IOT. Após isso, o nosso front end, através de integração com o back end, requisita as informações de Histórico e as torna visíveis para o usuário na plataforma web.



Horário	Colaborador	Setor
Cell	Cell	Cell
Cell	Cell	Cell
Cell	Cell	Cell
Cell	Cell	Cell
Cell	Cell	Cell

Partindo para outra interação, focaremos agora na troca da posse do tablet entre os **funcionários, nesse cenário**, será necessário o leitor RFID no tablet que irá interagir com os cartões RFID de cada funcionário. Em cada tablet, teremos um leitor RFID que é capaz de receber informações de um cartão RFID e envia essas informações para o nosso banco de dados, da mesma forma que o processo explicado anteriormente, assim, o nosso código backend entende que o tablet não está mais na posse do funcionário X, e assim que houver uma nova leitura, é configurada uma troca, onde agora o funcionário Y é quem tem a posse do tablet. Dessa forma, na próxima atualização do Histórico na plataforma, o colaborador atrelado ao

tablet já vai estar atualizado. Aqui vemos o leitor RFID conectado ao esp32.



Atualmente, a localidade do tablet é dada pelo nome do setor onde ele se encontra, o que pode ser observado na imagem do último processo. Isso garante uma maior assertividade na localidade, mas por outro lado, fornece uma informação mais abrangente do que uma coordenada estimada da localização do dispositivo.

## 6. Troubleshooting

(sprint 4)

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	Conexão falha no MQTT, interrompendo o envio de dados dos dispositivos IOT para o nosso backend.	Mecanismos de resolução para erros inesperados, isso inclui fazer com que o dispositivo tente se reconectar assim que for desconectado, garantindo que o dispositivo fique o mínimo possível de tempo sem conexão com a solução.
2	O sistema de verificação das notificações se mostrou falho por em algumas ocasiões, alterar o estado da notificação como pendente ou resolvido somente no frontend.	O código será revisado logicamente para seu funcionamento correto.
3	Conexão falha do	Atualmente, o esp apresenta um

	dispositivo IOT com o Wi-fi, mecanismo utilizado para dar a localidade do tablet.	código lógico que o faz tentar reconectar sempre que se desconectar do wi-fi, dessa forma, o provável é que após um tempo o dispositivo se conectará novamente sem intervenção.
4	Perda da conexão Wifi em localidades da fábrica devido a interferência eletromagnética	Sempre que o esp se desconectar da rede, ele tenta se conectar novamente.
5		