

Manual de Instruções

NOME DO PROJETO
Nome do Parceiro

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
21/11/2022	Moises Cazé, Elias Biondo, João Gonzalez, Felipe Silberberg, Ueliton Rocha, Camila Anacleto	3.0	Criação do documento

Índice

1. Introdução	3
1.1. Solução	3
1.2. Arquitetura da Solução	3
2. Componentes e Recursos	4
2.1. Componentes de hardware	4
2.2. Componentes externos	4
2.3. Requisitos de conectividade	4
3. Guia de Montagem	5
4. Guia de Instalação	6
5. Guia de Configuração	7
6. Guia de Operação	8
7. Troubleshooting	9
8. Créditos	10

1. Introdução

1.1. Solução (sprint 3)

Na ausência de um sistema de controle e monitoramento de processos. O cliente não consegue mapear o percurso feito pelo manobrista desde o recebimento do veículo até o estacionamento e vice-versa, bem como o tempo estimado para a realização dos trajetos de ida e volta. O gerente, por não ter um controle de produtividade, não possui uma maneira de gerenciar os manobristas e controlá-los. A solução é um sistema de controle que vincula o veículo ao manobrista que o conduz até a vaga na qual foi estacionado. Visando aumentar a qualidade e eficiência dos serviços oferecidos, permite aos gerentes controlar e monitorar os manobristas de acordo com sua produtividade, e aos clientes ter uma fila de espera dos veículos, bem como uma estimativa de tempo até o recebimento destes veículos.

1.2. Arquitetura da Solução (sprint 3)

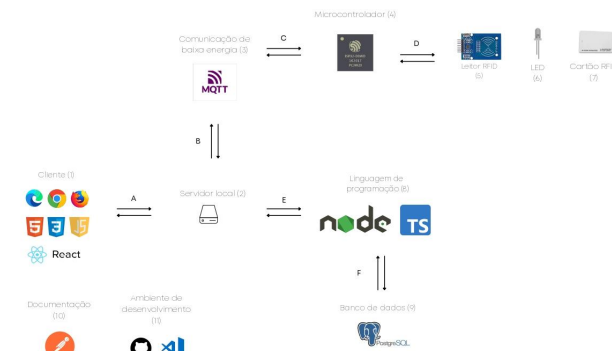


Tabela de Itens					
ID	Objeto	Sub-itens	Versão	Descrição	Motivo de utilização
1	Cliente	Navegadores	-	Programas que permitem seus usuários acessarem sites na Internet.	Acesso à plataforma web criada.
		HTML	5	Linguagem de marcação.	Construção da interface do usuário.
		CSS	3	Linguagem de estilização.	Construção da interface do usuário.
		ES (JS)	6	Linguagem de programação.	Construção da interface do usuário.
		React	17	Biblioteca para construção de interface do usuário.	Construção da interface do usuário.
2	Servidor	Servidor Local	-	Servidor local hospedado para a realização de serviços de maior performance, como processamento de imagens, tratamento de dados etc.	Redução da carga de processamento do microcontrolador.
3	CBE	MQTT	5	Protocolo de comunicação de baixa energia.	Transmissão de dados entre periféricos e servidores.
4	Microcontrolador	ESP32	S3	Módulo de alta performance e de baixa energia para aplicações.	Localização dos veículos.
5	Periférico	Lector RFID	-	Sensor de reconhecimento do cartão RFID	Conexão e transação de dados do cartão RFID com microcontrolador, recebe os dados do manobrista que está se vinculando.
7		LED	-	Diodo emissor de luz.	Feedback para os usuários.
8		Cartão RFID	-	Identificação por radiofrequência (método de identificação automática através de sinais de rádio).	Identificação de manobristas.
10	LP	NodeJS	16.06	Executor de código Javascript.	Criação de API para plataforma web.
11	Banco de Dados	TypeScript	4.8.4	Linguagem de programação.	Adição de tipos à aplicação Javascript.
12	Documentação	PostgreSQL	12	Sistema de gerenciamento de banco de dados relacional.	Armazenamento de dados.
13	AD	Postman	10.0.33	Plataforma de API para desenvolvedores.	Debug da aplicação.
		GitHub	-	Plataforma para versionamento de código.	Consolidação de desenvolvimento entre o time.
		VSCode	1.72	Ambiente de desenvolvimento integrado.	Produtividade e codificação.

Tabela de relacionamentos				
Identificação	Relação	Descrição	Exemplo de Uso	
A	1-2	Agentes de modificação do sistema. Envio, recebimento e consulta de dados na nuvem.	A notificação de uma nova ordem de serviço será enviada pelo servidor para o cliente.	
B	2-3	Transmissão de dados tratados para os periféricos da aplicação.	Serão acionados, no dispositivo, os atuadores de som e luminosidade.	
C	3-4	Transmissão de dados com baixo custo de energia. Envio e recebimento de ordens.	Toda a comunicação e comandos que serão executados no microcontrolador.	
D	4-5 e 4-7	Lector RFID integrado no microcontrolador recendo dados do cartão RFID.	O manobrista aproximará sua tag RFID ao leitor e estará vinculado.	
D	4-6	Visibilidade do sistema, exibição de status etc.	O led estará aceso continuamente na cor verde quando não associado à algum manobrista, e vermelho quando estiver.	
E	2-8	Transmissão das ordens recebidas para as estruturas de controles responsáveis.	Envio dos dados do manobristas para o servidor, onde serão tratados de acordo.	
F	8-9	Criação, leitura, atualização e deleção de dados. Controle do banco como um todo.	Adição de novas ordens de serviço, manobristas, tudo que for registrado.	

2. Componentes e Recursos

(sprint 3)

2.1. Componentes de hardware

Componentes	Utilidade	Quantidade
ESP32-S3-WRO OM-1	Serve para controlar sistemas eletrônicos não digitais	1
RFID-RC522	Permite que os usuários identifiquem e rastreiem de maneira automática e exclusiva os ativos	1
Cabos Jumper	Transmitir sinais analógicos e digitais	9
Resistores	Limita o fluxo da corrente elétrica	6
Leds LDR	Controle de iluminação	2
Buzzer	Geração de sinais sonoros	2
Cabo USB	Comunicação entre os equipamentos	1

2.2. Componentes externos

- Painel para que os manobristas verifiquem os veículos a estacionar e outro para o gestor captar os dados do cliente e enviar para os manobristas que precisam atender o cliente, seja estacionando ou retirando o veículo;
- Vscod: O Visual Studio Code, que é um editor de código fonte que auxilia na criação do código de software, inclusive nas fases de testes;
- Servidor com Node.js: Node.js é um ambiente de execução Javascript, linguagem padrão de manipulação de páginas HTML. Sua escolha é embasada em uma característica muito peculiar: sua alta escalabilidade, uma vez que a execução single-thread permite criar um Event Loop com requisições que não demandam output;
- Banco de dados PostgreSQL: O Postgres é um gerenciador de Bancos de Dados muito conhecido e usado no mundo do Desenvolvimento Web. Seus Bancos de Dados são relacionais, muito similar ao que vemos no MySQL, por exemplo;

- **Documentação com Postman:** O Postman é uma ferramenta que dá suporte à documentação das requisições feitas pela API. Ele possui ambiente para a documentação, execução de testes de APIs e requisições em geral;

2.3. Requisitos de conectividade

- **Protocolo I2C:** O modo de funcionamento do protocolo I2C é baseado na interação entre elementos seguindo a hierarquia mestre/escravo, ou seja, quando temos vários dispositivos se comunicando segundo esta premissa, pelo menos um destes deve atuar como mestre e os demais serão escravos. A função do mestre consiste em realizar a coordenação de toda a comunicação, pois, ele tem a capacidade de enviar e requisitar informações aos escravos existentes na estrutura de comunicação, os quais, devem responder às requisições citadas.
- **Rede Wifi:** O WiFi funciona através de ondas de rádio, assim como as TVs, aparelhos de rádio e celulares. A antena do roteador é a responsável por captar e emitir os sinais, bem como decodificá-los. E é assim que os aparelhos conseguem trocar informações.
- **FTM:** protocolo de comunicação baseado na estimativa da distância entre dois aparelhos, através do tempo que um sinal demora para propagar de um aparelho para o outro.
- **API:** consiste em um conjunto de normas que possibilita a comunicação entre plataformas através de uma série de

padrões e protocolos. Por meio desta, softwares e aplicativos, capazes de se comunicar com outras plataformas, podem ser criados.

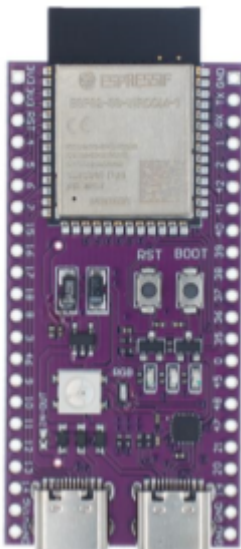
- **Protocolo MQTT:** foi criado com o objetivo de oferecer um baixo consumo de rede, banda e também dos demais recursos de software. O formato utilizado no MQTT é de Cliente/Servidor. Para funcionar, o Protocolo MQTT utiliza um modelo de Publish/Subscribe onde permite que o cliente faça postagens e/ou capte informações enquanto o servidor irá administrar esse envio e o recebimento dos respectivos dados. Ou seja, em um MQTT haverá um publicador onde será responsável por publicar as mensagens em um determinado tópico onde um assinante irá inscrever-se neste tópico para poder acessar a mensagem. Como não há uma conexão direta entre o assinante e o publicador, para que essas mensagens aconteçam, o protocolo MQTT irá precisar de um gerenciador de mensagens chamado de Broker.
- **Protocolo SPI:** é uma interface de comunicação simples de 4 fios, outra característica do SPI é que não existe o conceito de transferência de propriedade da “barra de dados” ou seja não existe um endereço específico para o master e ou slave. O SPI opera em modo full duplex, isto significa que os dados são transferidos em ambas as direções e ao mesmo tempo e isso faz com que sua velocidade de troca de dados seja bem mais rápida, superior a 10 Mhz em comparação com outros sistemas.

3. Guia de Montagem

Lista de Componentes:

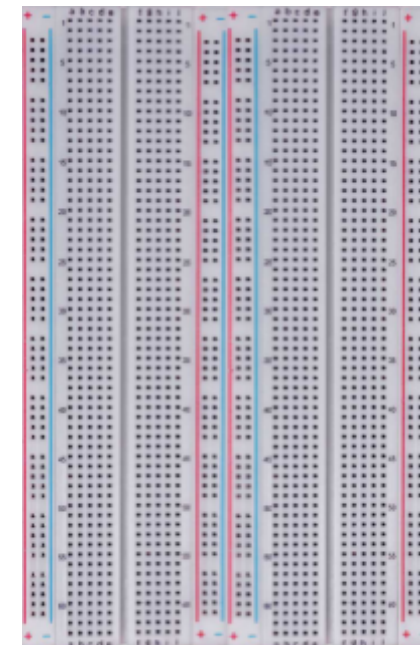
- 1 microcontrolador ESP32-S3;
- 2 leds RGB - cátodos comuns;
- 6 resistores 220r (vermelho, vermelho, marrom, dourado);
- 2 buzzers;
- 1 leitor RFID;
- 11 jumpers Dupont macho x macho.

1. O protagonista deste fluxo é um ESP32-S3, microcontrolador de baixa potência e baixo custo. Destina-se especialmente a fornecer versatilidade, robustez e confiabilidade em um grande número de aplicações.



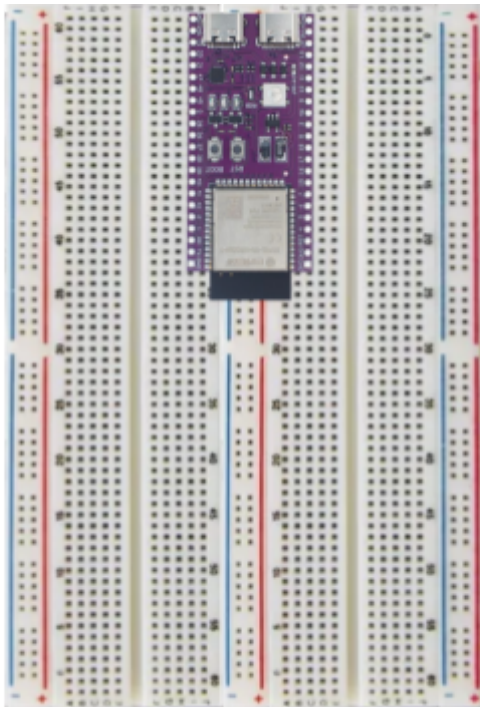
Fonte: 11nq.com/esp32-image

2. O suporte usado para a montagem dos circuitos deste protótipo é chamado protoboard. Uma placa na qual o microcontrolador (neste caso, o ESP) é integrado para que seja feita dentre os periféricos uma conexão que é possível através da transmissão de correntes elétricas provenientes do próprio microcontrolador. Segue o anexo abaixo:



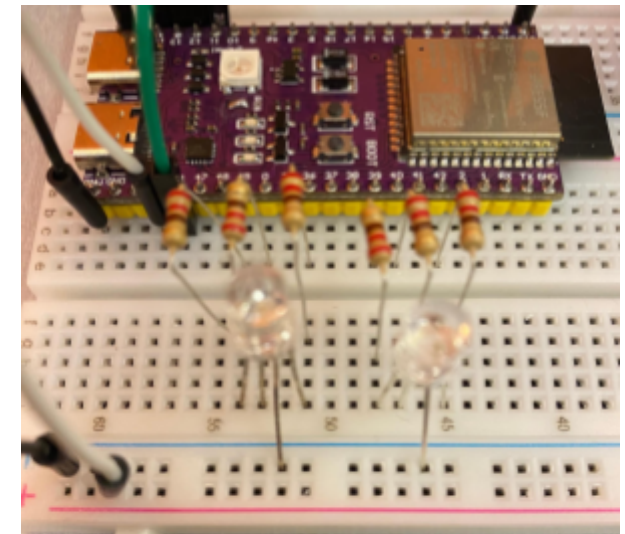
Fonte: 11nq.com/protoboard-image

3. O ESP32 deve ser inserido com as entradas USB próximas ao final da placa, com a primeira linha horizontal que se encontra mais abaixo (onde se encontram os 2 pinos GND) posicionada na fila de número 61 do protoboard. A linha do topo, que, por sua vez, possui os pinos GND e 3U3, deve estar posicionada na linha de número 40. Segue abaixo uma demonstração:



ESP32 conectada à protoboard

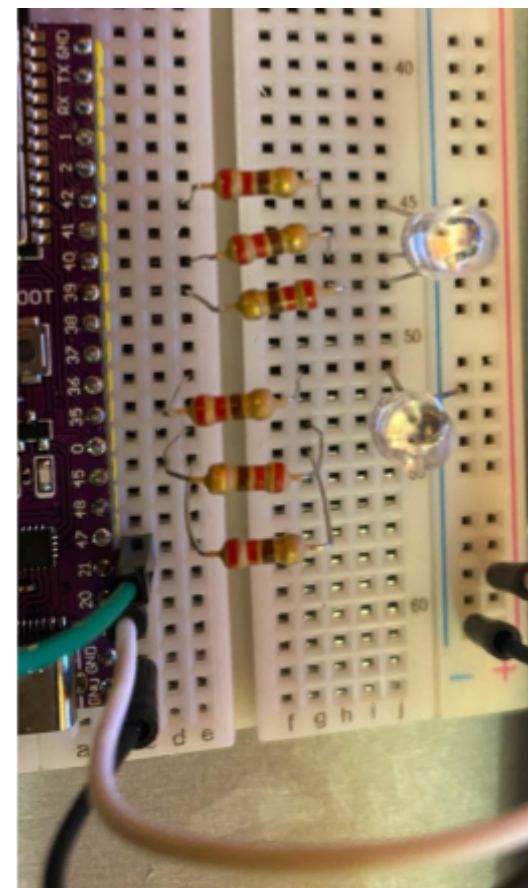
4. Antes de mostrar como os leds RGB devem estar posicionados, vamos a uma breve explicação sobre a estrutura deles: um cátodo comum possui 4 pinos, onde 3 são respectivos às cores vermelho, verde e azul e um, à corrente negativa de energia. Da esquerda à direita, são os pinos: vermelho, negativo (o maior de todos), verde e, por último, o da cor azul. A fila vertical posicionada à direita do sinal de negativo e a linha vertical azul, uma vez conectada a um pino GND do ESP, pode receber pinos de vários terminais negativos, o mesmo para o positivo. portanto, vamos conectar os 2 terminais negativos dos leds (pernas maiores) na última fila negativa à direita da protoboard, nos pontos 46 e 52, conforme ilustrado na imagem à direita.



Leds RGB conectados

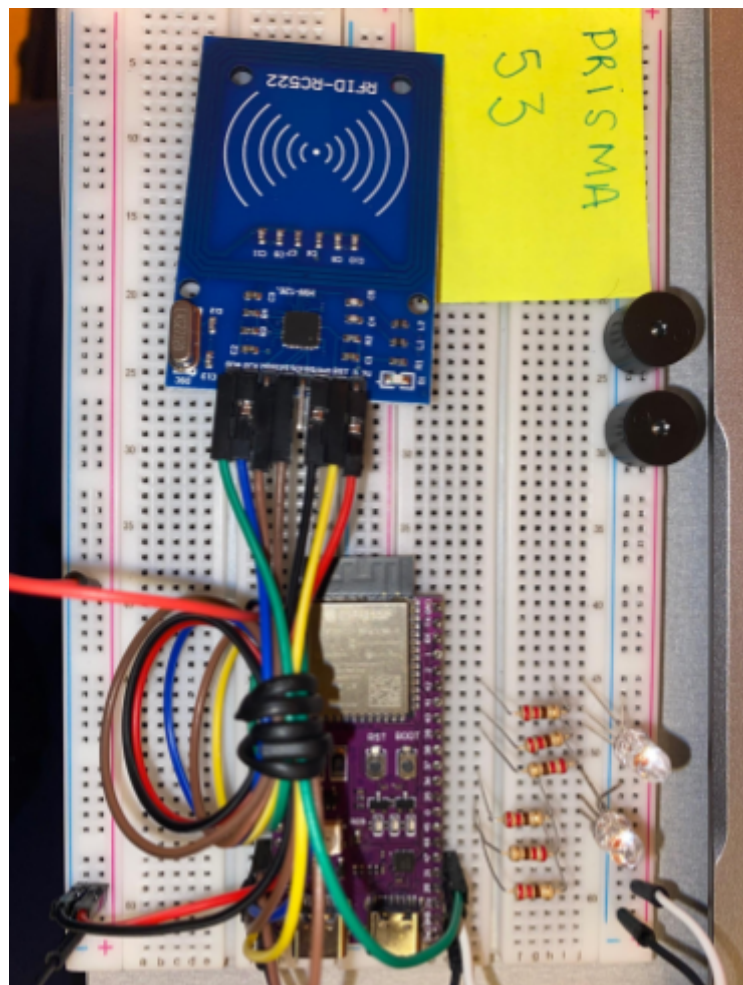
5. Todos os 6 resistores estão conectados aos pinos positivos (que representam as cores) dos leds. Segue a ordem: O led RGB que já possui seu pino no ponto de número 52 da fila negativa se encontrará no seguinte estado: pino azul no ponto 54, verde no 53 e vermelho no 52. Deve-se inserir um resistor na linha horizontal onde se encontra o pino azul (ponto 54) que conecta à porta 45 do ESP, outro na linha onde fica o pino verde (53) que conecta à porta 0, e, por último, um no vermelho (52) que conecta na porta 36. Quanto ao segundo led, devem ser feitas as seguintes conexões com resistores: vermelho (45) na porta 42 do ESP, verde (47) na porta 40 e azul (48) na porta 39. Segue anexo desta parte do circuito já conectada.

6. Por fim, as filas positivas e negativas da protoboard que não estejam conectadas a porta nenhuma devem ser conectadas a portas 3V3 e GNDs, respectivamente, para que a energia seja passada do microcontrolador aos periféricos conectados nelas.



Resistores nos pontos positivos do led RGB

Uma vez montado todo o circuito, a protoboard se encontrará conforme ilustrado na figura abaixo:



Circuito completamente montado

4. 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.

5. Guia de Configuração

(sprint 4)

Descreva passo-a-passo como configurar os dispositivos IoT utilizando os equipamentos devidos (ex. smartphone/computador acessando o servidor embarcado ou a página na nuvem).

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

6. Guia de Operação

(sprint 5)

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.

7. Troubleshooting

(sprint 5)

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		
2		
3		
4		
5		

8. Créditos

(sprint 5)

Seção livre para você atribuir créditos à sua equipe e respectivas responsabilidades