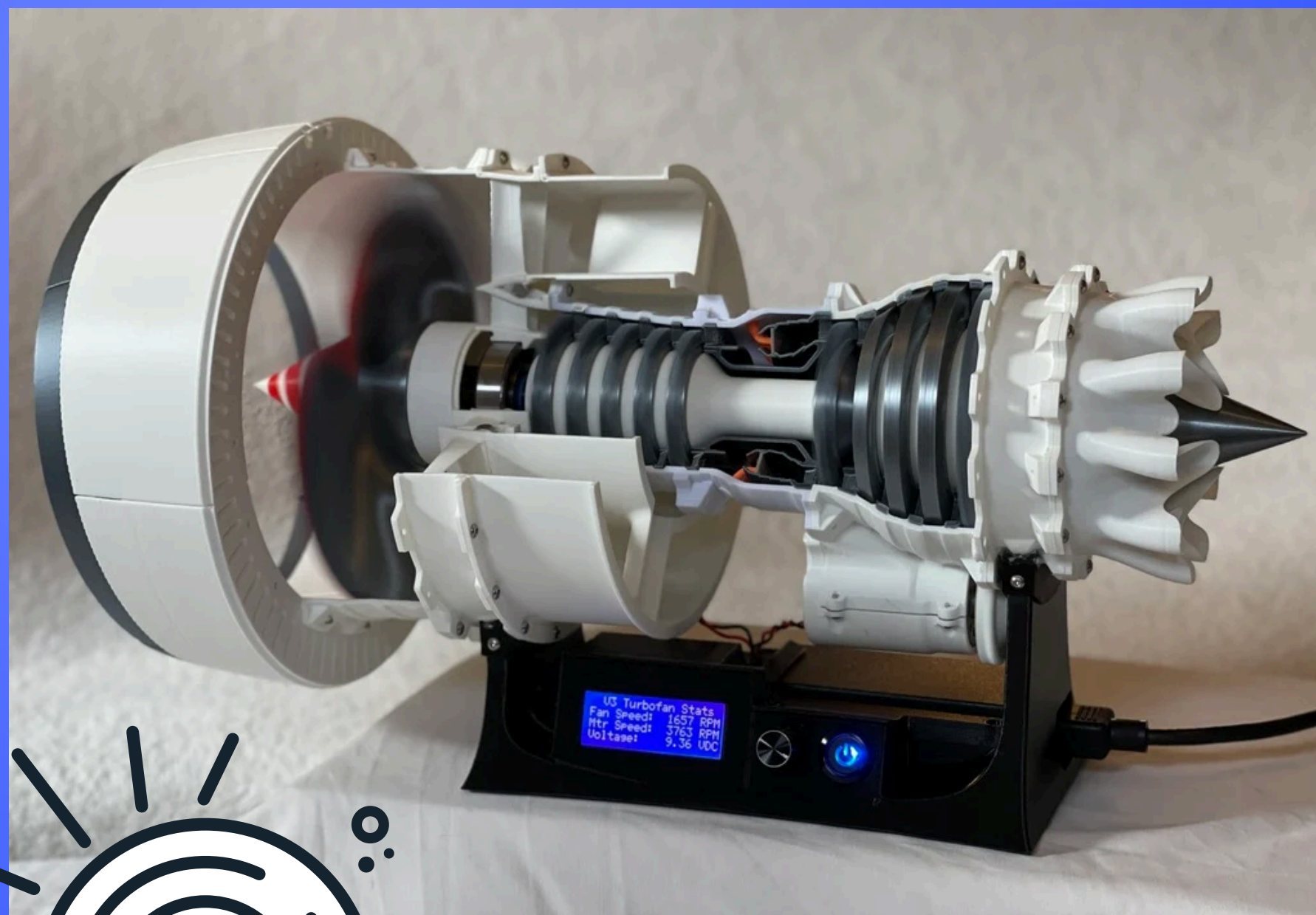


# Projeto: Motor Eléctrico com Controle Web

## 1. Definição do Problema





# Introdução

O projeto visa desenvolver um motor cuja rotação possa ser controlada e monitorada em tempo real por meio de um aplicativo web.



# Importância

- Controle de RPM fundamental para performance de motores
- Permite regular consumo de energia / combustível
- Impacto ambiental positivo: menos emissões com rotações adequadas
- Aprimoramento de habilidades práticas em sistemas embarcados

# Motivação

- Desenvolver uma aplicação com sistemas embarcados
- Aprender controle de motores e aplicações IoT
- Explorar a integração entre hardware e software





## OBJETIVOS PRINCIPAIS

- Controlar velocidade de rotação do motor via aplicativo web
- Medir e visualizar métricas de rotação em tempo real
- Permitir acesso público controlado via WiFi
- Demonstrar integração hardware-software

## OBJETIVOS ESPECÍFICOS

- Controlar velocidade do motor via PWM no ESP32
- Medir RPM com sensor de rotação (Hall Effect ou óptico)
- Desenvolver interface web para controle e métricas
- Criar comunicação bidirecional (WebSocket ou MQTT) entre App e ESP32



# Funcionalidades



O projeto vai permitir:

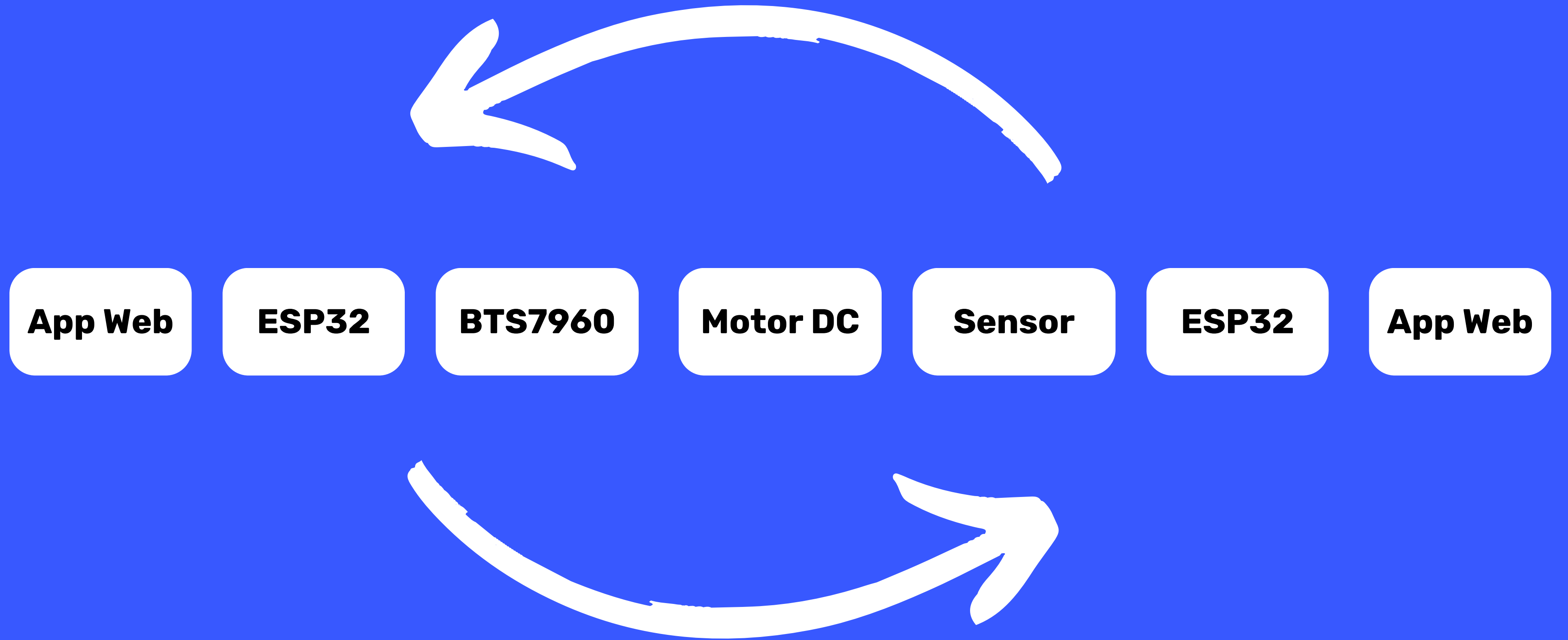
- Controle de velocidade do motor em tempo real via aplicativo web.
- Visualização de métricas do motor, como RPM, potência estimada, consumo de energia e tempo de operação.
- Acesso público controlado via Wi-Fi, incluindo sistema de fila e QR Code para acesso rápido.
- Controle bidirecional do motor (rotação horária e anti-horária) com proteção integrada do driver.
- Registro e histórico das métricas para análise posterior de desempenho e eficiência energética.



# Arquitetura Utilizada

- Microcontrolador (MCU): ESPC6-32 N4.
- Driver do motor: BTS7960 H-Bridge (aplica PWM e protege o motor).
- Motor: DC GA25-370 (169 RPM, 6V).
- Sensor de rotação: Hall Effect (KY-003) ou sensor óptico para medir RPM.
- Barramento de comunicação:
- Entre App Web e ESP32: Wi-Fi usando WebSocket (ou MQTT) para comunicação bidirecional em tempo real.
- Entre ESP32 e driver/motor: sinais PWM e controle digital de direção.





# Metodologia e Organização do Grupo



## Metodologia de desenvolvimento:

- Hardware: cascata (Waterfall) → montagem, testes e integração do motor, driver e sensor.
- Software: SCRUM → desenvolvimento do frontend, backend, comunicação e interface de métricas.

## Organização do time:

- Equipe Hardware: controla motor, sensores, driver e alimentação, faz ajustes físicos e testes de integração.
- Equipe Software: desenvolve aplicativo web, interface de controle, gráficos de métricas, sistema de usuários e comunicação com ESP32.

Integração Hardware-Software: fases de teste conjunto, validação de controle do motor e visualização de métricas em tempo real.





# Bibliografia

- Nairck. 3D Printable Jet Engine - V3 Turbofan Driver. Disponível em: <https://www.thingiverse.com/thing:4743929>

