# Visão Geral do Projeto

Sistema de controle e monitoramento de motor elétrico via aplicativo web, desenvolvido para a disciplina Sistemas Embarcados.

## 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

# Estrutura do Projeto

## Fase 1: Desenvolvimento Paralelo (Hardware + Software)

* **Metodologias:**
  + Hardware: Cascata (Waterfall)
  + Software: SCRUM

## Fase 2: Integração Hardware-Software

## Divisão de Equipes

### Equipe Hardware

* **Membros:** Talita e Charles
* **Responsabilidades:**
  + Controle do motor elétrico
  + Implementação de sensores de rotação
  + Modificações físicas no motor (redução de atrito)
  + Sistema de alimentação por bateria
  + Configuração do módulo WiFi

### Equipe Software

* **Membros:** Paulo e Rafael
* **Responsabilidades:**
  + Desenvolvimento do aplicativo web
  + Interface de controle de velocidade
  + Sistema de visualização de métricas
  + Sistema de fila de usuários
  + Geração de QR Code

# Arquitetura do Sistema de Controle

## Como Funciona o Controle de Rotação

### Fluxo de Controle

[App Web] → [WiFi] → [Microcontrolador] → [BTS7960] → [Motor GA25-370] → [Sensor] → [Microcontrolador] → [App Web]

### Função do BTS7960 (Driver H-Bridge)

O BTS7960 atua como "amplificador de potência" entre Microcontrolador e motor:

* Microcontrolador: Fornece sinais de controle (3.3V, ~40mA)
* BTS7960: Amplifica para potência necessária (6V, até 450mA para o motor)
* Proteções integradas: Sobrecorrente, sobretensão, superaquecimento
* Controle bidirecional: Permite rotação horária e anti-horária

### Controle PWM (Modulação por Largura de Pulso)

* 0% PWM: Motor parado
* 50% PWM: Motor à meia velocidade
* 100% PWM: Motor na velocidade máxima (169 RPM)
* Frequência: Até 25kHz (padrão Arduino ~1kHz)

### Alimentação

* Bateria 12V → Regulador 6V → Motor
* Bateria 12V → Regulador 5V → BTS7960
* Bateria 12V → Regulador 3.3V → Microcontrolador

# Hardware

## Componentes Definidos

* Motor: GA25-370 (169 RPM @ 6V, corrente ~0.45A com carga)
* Driver de Motor: BTS7960 H-Bridge (43A máximo, PWM até 25kHz)
* Microcontrolador: Microcontrolador-WROOM-32 DevKit (WiFi integrado, 3.3V)
* Sensor de rotação: Sensor óptico ou magnético
* Alimentação: Bateria 12V com reguladores (6V motor, 5V BTS7960, 3.3V Microcontrolador)
* Componentes auxiliares: Reguladores de tensão, capacitores, resistores

## Especificações Técnicas

* Controle de velocidade: PWM 0-100% via BTS7960
* Direções: Horária e anti-horária (H-Bridge completo)
* Monitoramento: RPM real via Sensor, corrente via sensor integrado
* Comunicação: WiFi 802.11 b/g/n integrado no Microcontrolador
* Proteções: Sobrecorrente, sobretensão, superaquecimento (BTS7960)
* Autonomia: Dependente da bateria escolhida

## Requisitos do Microcontrolador

* **Para o BTS7960:**
  + Tensão de controle: 3.3V ~ 5V
  + PWM: Até 25kHz
  + Pinos digitais: Mínimo 4 (RPWM, LPWM, R\_EN, L\_EN)
  + Corrente de saída: Baixa (apenas sinal de controle)
* **Para o Motor GA25-370:**
  + Alimentação: 6V DC (separada do microcontrolador)
  + Controle: Via BTS7960 (não direto)
* **Para o Projeto:**
  + WiFi: Obrigatório para aplicativo web
  + PWM: Para controle de velocidade
  + ADC: Para leitura de sensores (Sensor, corrente)

## KY-003 Hall Effect Sensor

* Funcionamento: Digital (HIGH/LOW)
* Alimentação: 3.3V-5V
* Saída: Sinal digital limpo
* Instalação: Fixar próximo ao motor + ímã no eixo
* **Uso:**
  + Ímã pequeno colado no eixo do motor (ou acoplado)
  + Sensor fixo próximo ao eixo
  + Cada volta completa = 1 pulso no microcontrolador
  + RPM = (pulsos/minuto)
* **Para controle do motor:**
  + Envia dados digitais para o BTS7960 que os traduz em pulsos e envia para o motor

# Software

## Stack

* Frontend: React.js
* Backend: Node.js com Express
* Comunicação: WebSockets para tempo real
* Banco de Dados: SQLite (local)
* Estilização: CSS3 com animações

## Funcionalidades Principais

1. **Interface de Controle**
   * Velocímetro visual (similar ao de carro)
   * Controle deslizante de velocidade
   * Botões de emergência/parada
2. **Sistema de Métricas**
   * RPM atual
   * Consumo de energia estimado
   * Tempo de operação
   * Gráficos históricos
   * Eficiência energética
3. **Sistema de Usuários**
   * Registro simples (apenas nome)
   * Fila de espera
   * Tempo limite por sessão
   * Lista de nomes proibidos (regex)
4. **Conectividade**
   * Geração de QR Code automático para acessar o site
   * SSID e senha padrão configuráveis
   * Interface responsiva para mobile

## Funcionalidades Detalhadas

### Sistema de Controle de Acesso

* Acesso único: Apenas 1 usuário controlando por vez
* Fila de espera: Sistema de turnos automático
* Tempo limite: 5-10 minutos por sessão (configurável)
* QR Code: Acesso rápido via smartphone

### Visualização de Dados

* Velocímetro Digital: Interface visual atrativa
* **Métricas em Tempo Real:**
  + RPM atual
  + Potência consumida
  + Eficiência
  + Temperatura (se sensor disponível)
* Histórico: Gráficos das últimas sessões

### Sistema de Validação

* Nomes Proibidos: Lista configurável
* Regex: Validação de padrões
* Sanitização: Prevenção de caracteres especiais

## Métricas Calculadas

Com os dados de RPM do Sensor, calcularemos:

* Velocidade angular (rad/s, RPM)
* Potência mecânica (P = τ × ω)
* Eficiência energética (Saída/Entrada × 100%)
* Consumo de corrente via sensor BTS7960
* Energia acumulada (kWh)
* Tempo de operação por sessão