**Documentação de Desenvolvimento e Teste**

**Sistema de Monitoramento Remoto para Dispositivo Médico**

**Índice**

1. [Introdução](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#1-introdu%C3%A7%C3%A3o)
2. [Ambiente de Desenvolvimento](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#2-ambiente-de-desenvolvimento)
3. [Estrutura do Projeto](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#3-estrutura-do-projeto)
4. [Desenvolvimento](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#4-desenvolvimento)
   * 4.1 [Firmware do Dispositivo Médico](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#41-firmware-do-dispositivo-m%C3%A9dico)
   * 4.2 [Comunicação Segura](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#42-comunica%C3%A7%C3%A3o-segura)
   * 4.3 [Interface de Usuário](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#43-interface-de-usu%C3%A1rio)
   * 4.4 [Servidor Remoto](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#44-servidor-remoto)
5. [Testes](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#5-testes)
   * 5.1 [Testes Unitários](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#51-testes-unit%C3%A1rios)
   * 5.2 [Testes de Integração](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#52-testes-de-integra%C3%A7%C3%A3o)
   * 5.3 [Testes de Sistema](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#53-testes-de-sistema)
6. [Documentação Adicional](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#6-documenta%C3%A7%C3%A3o-adicional)
7. [Conclusão](https://chat.openai.com/c/b2609ee9-8541-4866-825a-4aac2814abed#7-conclus%C3%A3o)

**1. Introdução**

O Sistema de Monitoramento Remoto para Dispositivo Médico é um projeto inovador da EmbeddedTech, desenvolvido para fornecer uma solução avançada e confiável para o monitoramento remoto de pacientes. Este documento apresenta detalhes sobre o desenvolvimento do sistema, incluindo a arquitetura, as tecnologias utilizadas e os procedimentos de teste.

**2. Ambiente de Desenvolvimento**

* Linguagens de programação:
  + Firmware: C/C++
  + Comunicação Segura: Python
  + Interface de Usuário: HTML5, CSS3, JavaScript (React)
  + Servidor Remoto: Node.js, Express, MongoDB
* Ferramentas:
  + IDE para Firmware: Visual Studio Code
  + Ferramentas de Desenvolvimento Web: Visual Studio Code, Adobe XD
  + Ferramentas de Comunicação Segura: OpenSSL, PyCrypto
  + Ferramentas de Servidor Remoto: npm, MongoDB Compass

**3. Estrutura do Projeto**

**Sistema-de-Monitoramento-Remoto/**

**|-- Firmware/**

**| |-- src/**

**| | |-- main.c**

**| |-- include/**

**| | |-- sensor.h**

**|-- ComunicacaoSegura/**

**| |-- scripts/**

**| | |-- encrypt.py**

**|-- InterfaceUsuario/**

**| |-- css/**

**| | |-- styles.css**

**| |-- js/**

**| | |-- main.js**

**| |-- html/**

**| | |-- index.html**

**|-- ServidorRemoto/**

**| |-- src/**

**| | |-- server.js**

**| |-- database/**

**| | |-- models/**

**|-- Documentacao/**

**|-- Testes/**

**4. Desenvolvimento**

4.1 Firmware do Dispositivo Médico

* O firmware é responsável pela aquisição de dados dos sinais vitais, utilizando sensores avançados e garantindo uma transmissão eficiente ao servidor remoto.

4.2 Comunicação Segura

* Implementação de protocolos seguros utilizando algoritmos de criptografia avançados para a transmissão de dados entre o dispositivo médico e o servidor remoto.

4.3 Interface de Usuário

* Desenvolvimento de uma interface de usuário moderna e responsiva, utilizando tecnologias web avançadas como React para proporcionar uma experiência intuitiva aos profissionais de saúde.

4.4 Servidor Remoto

* Configuração do servidor remoto utilizando Node.js e Express para receber, armazenar e processar os dados do dispositivo médico. A integração com o banco de dados MongoDB assegura a eficiência na manipulação de grandes volumes de dados.

**5. Testes**

5.1 Testes Unitários

* Realização de testes unitários utilizando frameworks como Jest (JavaScript) e Unity (C/C++) para garantir a funcionalidade correta de cada componente individual.

5.2 Testes de Integração

* Verificação da integração efetiva entre os diferentes módulos do sistema, simulando cenários realistas de comunicação entre o dispositivo médico e o servidor remoto.

5.3 Testes de Sistema

* Execução de testes abrangentes para validar o sistema como um todo, incluindo a interação entre o dispositivo médico, a comunicação segura, a interface de usuário e o servidor remoto.

**7. Conclusão**

Esta documentação abrangente proporciona uma visão detalhada do desenvolvimento e dos testes do Sistema de Monitoramento Remoto para Dispositivo Médico da EmbeddedTech. Quaisquer atualizações ou modificações devem ser registradas aqui para manter a coesão e a compreensão do projeto.