



MODELO DE PLANO DE TESTES

NOME DA EQUIPE: ARKHAM

PARTICIPANTES: Arthur Ramos, Leonardo Castro e Lucas Gabriel

Este modelo pode ser adaptado conforme necessário para atender aos requisitos específicos do projeto.

1. Introdução

Objetivo: Este documento define o plano de teste e os casos de teste para o Projeto Maloca das iCoisas, com o intuito de verificar a funcionalidade, desempenho, segurança e confiabilidade dos dispositivos e sistemas implementados.

Escopo: Os testes cobrem os principais componentes e funcionalidades do sistema, incluindo a integração de sensores, atuação de dispositivos e a comunicação entre as plataformas (Arduino, Raspberry Pi, ESP32) e o servidor.

2. Estratégia de Teste

Metodologia: A metodologia utilizada será baseada em testes manuais e automáticos, com foco em testes funcionais, de integração, de desempenho e de segurança.

Ambiente de Teste:

- Dispositivos: ESP32
- Ferramentas: Arduino IDE, Wokwi, Firebase, MQTT

Responsáveis pelo teste: Arkham

3. Casos de Teste

Caso de Teste 1: Leitura de Sensores de Temperatura e Umidade

- **ID:** CT-001
- **Descrição:** Verificar se o sensor de temperatura e umidade coleta e transmite corretamente os dados para o sistema.
- **Pré-condição:** O sensor de temperatura está conectado ao ESP32 e configurado corretamente.



- Passos de Teste:
 1. Ligar o dispositivo ESP32.
 2. Coletar leituras de temperatura e umidade em intervalos de 10 segundos.
 3. Verificar a precisão das leituras com base em uma fonte de referência.
- Resultado Esperado: As leituras de temperatura devem ser precisas dentro da margem de erro especificada pelo fabricante do sensor.
- Resultado Real: [Espaço para preenchimento após o teste]
- Status: [Passou/Falhou]

Caso de Teste 2: Leitura de Sensores de Giroscópio e Acelerômetro

- ID: CT-002
- Descrição: Validar se o dispositivo ESP32 conecta ao Wi-Fi, lê dados dos sensores de giroscópio e acelerômetro, e envia essas informações para o servidor de forma confiável.
- Pré-condição: Configuração do Wi-Fi salva no dispositivo ESP32 e sensores de giroscópio e acelerômetro conectados ao ESP32.
- Passos de Teste:
 1. Ligar o ESP32 e verificar a conexão Wi-Fi.
 2. Testar leitura dos sensores de giroscópio e acelerômetro.
 3. Enviar os dados dos sensores para o servidor.
- Resultado Esperado: O ESP32 deve se conectar automaticamente ao Wi-Fi, ler corretamente os dados dos sensores de giroscópio e acelerômetro, e enviar os dados para o servidor sem perda de pacotes.
- Resultado Real: [Espaço para preenchimento após o teste]
- Status: [Passou/Falhou]

Caso de Teste 3: Integração do Firebase com o ESP32

- ID: CT-003
- Descrição: Validar se o dispositivo ESP32 se conecta ao Wi-Fi, lê dados dos sensores, e envia essas informações para o Firebase de forma confiável.
- Pré-condição: Configuração do Wi-Fi salva no dispositivo ESP32, sensores de giroscópio e acelerômetro conectados ao ESP32, e integração com o Firebase configurada.
- Passos de Teste:
 1. Ligar o ESP32 e verificar a conexão Wi-Fi.
 2. Testar leitura dos sensores.



3. Enviar os dados dos sensores para o Firebase.

- Resultado Esperado: O ESP32 deve se conectar automaticamente ao Wi-Fi, ler corretamente os dados dos sensores, enviar os dados para o Firebase sem perda de pacotes e garantir que a recepção no Firebase seja precisa.
- Resultado Real: [Espaço para preenchimento após o teste]
- Status: [Passou/Falhou]

Caso de Teste 4: Integração com o Dart

- ID: CT-004

- Descrição: Validar se o dispositivo ESP32 se conecta ao Wi-Fi, lê dados dos sensores, envia essas informações para o Firebase e se a aplicação Dart consegue recuperar e exibir esses dados corretamente.
- Pré-condição: Configuração do Wi-Fi salva no dispositivo ESP32. Sensores de giroscópio e acelerômetro conectados ao ESP32. Integração com o Firebase configurada corretamente. Aplicação Dart configurada para acessar o Firebase e exibir os dados..

- Passos de Teste:

1. Ligar o ESP32 e verificar a conexão Wi-Fi.
2. Testar a leitura dos sensores no ESP32.
3. Enviar os dados dos sensores para o Firebase.
4. Verificar se os dados aparecem corretamente no Firebase sem perda de pacotes.
5. Abrir a aplicação Dart e validar se os dados são recuperados corretamente do Firebase.
6. Comparar os valores exibidos na aplicação Dart com os dados enviados pelo ESP32.

- Resultado Esperado:

- O ESP32 deve se conectar automaticamente ao Wi-Fi.

- O dispositivo deve ler corretamente os dados dos sensores.
- Os dados devem ser enviados para o Firebase sem perdas ou inconsistências.
- O Firebase deve armazenar os dados com precisão.
- A aplicação Dart deve conseguir recuperar e exibir os dados corretamente, sem atrasos ou erros.

- Resultado Real: [Espaço para preenchimento após o teste]
- Status: [Passou/Falhou]

4. Critérios de Aprovação



- Funcionalidade: Todos os casos de teste funcionais devem ser aprovados.
- Desempenho: O tempo de resposta dos dispositivos não deve exceder o limite especificado.
- Segurança: Nenhuma vulnerabilidade crítica deve ser encontrada.
- Resiliência: O sistema deve retomar a comunicação após falhas de rede sem perda de dados.

5. Conclusão

Resumo dos Resultados: [Espaço para um resumo geral dos resultados após a execução dos testes]

Recomendações: [Espaço para recomendações adicionais ou ajustes no sistema]