



## PLANO DE TESTES

**NOME DA EQUIPE:** 2A1R

**PARTICIPANTES:** Álefe Alves, Andersson Silva, Ranier Sales

### 1. Introdução

**Objetivo:** Este documento define o plano de teste e os casos de teste para o Projeto do Maloca das iCoisas nomeado de genSUS, 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 (ESP32 e Flutter) 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, VScode, Wokwi, FireBase.

**Responsáveis pelo teste:** Equipe 2A1R

### 3. Casos de Teste

#### #### Caso de Teste 1: Leitura do Sensor RFID

- **ID: CT-001**
- **Descrição:** Verificar se o sensor RFID está apresentando a leitura das tags corretamente.
- **Pré-condição:** O sensor RFID deve estar corretamente conectado ao ESP32 e configurado corretamente.
- **Passos de Teste:**
  1. Ligar o dispositivo ESP32.
  2. Coleta de leituras do cartão RFID.



3. Verificar se as leituras ativam corretamente as funcionalidades previstas.

- Resultado Esperado: As leituras das tags devem ser reconhecidas sem apresentar problema caso a tag esteja previamente cadastrada no sistema, e ativando as funcionalidades previstas no sistema.
- Resultado Real: O teste apresentou resultados positivos quanto a leitura de tags.
- Status: Aprovado.

#### #### Caso de Teste 2: Verificação do Sensor LDR

- ID: CT-002
- Descrição: Validar se o sensor LDR está respondendo corretamente aos diferentes intensidades de luz(Fraco ou Forte).
- Pré-condição: O sensor LDR deve estar corretamente conectado ao ESP32 e configurado corretamente conforme o código.
- Passos de Teste:
  1. Ligar o dispositivo ESP32.
  2. Verificar se o sensor LDR responde corretamente as diferentes intensidades de luz.
  3. Verificar se além de responder corretamente, ele ativa os LED's no circuito.
- Resultado Esperado: Com luz, o sensor LDR deve fazer com que o LED vermelho permaneça ligado, já sem luz o LED verde é que deve ser ativado.
- Resultado Real: O sensor LDR funciona corretamente simulando a leitura de uma digital. Ele estimula corretamente o sistema enviando sinais para ativar ou não os LED's.
- Status: Aprovado.

#### #### Caso de Teste 3: Verificação do LCD

- ID: CT-003
- Descrição: Validar se o display LCD está exibindo corretamente as informações enviadas pelo ESP32, de acordo com o código e as interações com o sensor LDR/Leitor de Digital.
- Pré-condição: O display LCD deve estar corretamente conectado ao ESP32 e o código de exibição configurado corretamente.
- Passos de Teste:
  1. Ligar o dispositivo ESP32.
  2. Verificar a inicialização do LCD, exibindo a mensagem correta definida no código.



3. Verificar se o LCD responde corretamente ao potenciômetro que regula o brilho.
  4. Verificar se o LCD responde de forma rápida e sem falhas ao código enviado pelo ESP32.
- Resultado Esperado: Os botões devem enviar os estímulos necessários para ativar as determinadas opções.
  - Resultado Real: O LCD funciona corretamente apresentando os resultados esperados a cada interação com os sensores presentes.
  - Status: Aprovado.

#### ### Caso de Teste 4: Verificação dos Push Buttons

- ID: CT-004
- Descrição: Validar se os três push buttons estão funcionando corretamente e ativando as opções no código, conforme pressionados.
- Pré-condição: Os três push buttons devem estar corretamente conectados ao ESP32 e o código deve estar configurado para reconhecer cada botão e ativar a opção correspondente.
- Passos de Teste:
  1. Ligar o dispositivo ESP32
  2. Pressionar cada push button e verificar se a opção corresponde a cada botão.
  3. Verificar se a resposta ao pressionar os botões é imediata e sem falhas.
- Resultado Esperado: O sistema deve responder corretamente a cada botão pressionado, ativando a opção correspondente no código e exibindo a resposta corretamente no LCD, LEDs ou outra interface.
- Resultado Real: Cada botão responde corretamente caso pressionado, selecionando uma opção específica do sistema.
- Status: Aprovado.

#### ### Caso de Teste 5: Verificação da Interface Mobile com Flutter

- ID: CT-005
- Descrição: Validar se a interface mobile desenvolvida com Flutter está funcionando corretamente em dispositivos móveis.
- Pré-condição: O servidor Flutter deve estar corretamente configurado e executando no ambiente local ou em produção.
- Passos de Teste:
  1. Ligar o servidor Flutter



2. Acessar a interface no dispositivo móvel e verificar se a interface é carregada corretamente, sem erros ou falhas de exibição.
  3. Verificação do tempo de resposta da interface ao interagir.
- Resultado Esperado: A interface Flutter deve ser acessível corretamente em dispositivos móveis, ajustando-se ao tamanho da tela e exibindo todos os elementos de forma legível e interativa.
  - Resultado Real: A interface Flutter apresentou o comportamento esperado, podendo realizar cadastro de novos usuários e agendar consultas futuras.
  - Status: Aprovado.

### ### Caso de Teste 6: Verificação da Integração com o Firebase

- ID: CT-006
- Descrição: Validar a integração entre a aplicação (Flutter e ESP32) e o Firebase para garantir que os dados sejam corretamente enviados, armazenados e recuperados do banco de dados em tempo real.
- Pré-condição: A configuração do Firebase deve estar realizada corretamente (com as credenciais adequadas e o banco de dados configurado). O código da aplicação (Flutter e ESP32) deve estar configurado para interagir com o Firebase.
- Passos de Teste:
  1. Verificar se a biblioteca Firebase foi corretamente instalada e configurada no ambiente da aplicação (firebase-admin no Flutter e FirebaseESP32 no ESP32).
  2. Inicializar a conexão com o Firebase a partir da aplicação (via código de inicialização no Flutter ou ESP32).
  3. Verificar se a aplicação consegue se conectar corretamente ao Firebase, sem erros de autenticação ou configuração.
  4. Confirmar que os dados estão sendo armazenados com os valores corretos e no formato adequado.
- Resultado Esperado: A aplicação deve conseguir se conectar com o Firebase corretamente, enviar dados e armazená-los no banco de dados.
- Resultado Real: A integração entre as aplicações ESP32 e Flutter ocorreram de forma satisfatória apresentando resultados esperados conforme diferentes submissões de requisições.
- Status: Aprovado.

### ### Caso de Teste 7: Verificação do Sensor de Leitura Digital



- **ID: CT-006**
- **Descrição:** Validar a leitura correta da digital pelo sensor, garantindo que a aplicação (Flutter ou ESP32) interaja corretamente com o hardware e armazene a informação no banco de dados, como o Firebase, sem falhas de leitura ou armazenamento.
- **Pré-condição:** O sensor de leitura digital deve estar corretamente instalado e configurado, e o código da aplicação (Flutter ou ESP32) deve estar preparado para capturar e processar os dados da digital. O Firebase também deve estar configurado corretamente para armazenar os dados.
- **Passos de Teste:**
  1. Verificar se a biblioteca do sensor de leitura digital foi corretamente instalada e configurada no ambiente da aplicação (por exemplo, "Adafruit Fingerprint Sensor Library" para ESP32).
  2. Inicializar a conexão com o sensor e verificar se a leitura da digital é iniciada sem erros.
  3. Validar a captura da digital por meio do sensor, verificando se a impressão digital é lida de maneira precisa (evitar erros de leitura ou falhas no sensor).
  4. Confirmar que, após a leitura, os dados da impressão digital são enviados para o banco de dados, e que a aplicação consegue armazená-los corretamente no Firebase.
  5. Verificar se a aplicação consegue recuperar a digital do Firebase corretamente quando necessário.
  6. Testar a aplicação com diferentes impressões digitais para garantir que o sensor funcione de forma robusta.
- **Resultado Esperado:** O sensor de leitura digital deve capturar e armazenar as digitais corretamente, sem erros de leitura, e os dados devem ser enviados e recuperados do Firebase de maneira eficiente.
- **Resultado Real:** O sensor de leitura digital apresentou um funcionamento esperado lendo corretamente as digitais passadas. Não foi integrado a funcionalidade de envio de dados ao Firebase
- **Status:** Parcialmente aprovado.

### ### 8. Critérios de Aprovação



- Funcionalidade: Todos os casos de teste funcionais devem ser aprovados em diversas tentativas de testes.
- Desempenho: O tempo de resposta dos dispositivos não deve exceder o limite especificado.
- Segurança: Nenhuma vulnerabilidade crítica deve ser encontrada no sistema, principalmente na conexão com o firebase.
- Resiliência: O sistema deve retomar a comunicação após falhas de rede sem perda de dados.

### ### 9. Conclusão

- Resumo dos Resultados: A primeira tentativa de implementação do projeto apresentou algumas dificuldades em relação a IDE arduino. Com isso não foi possível implementar os casos de testes previstos para cada componente da prototipagem no tempo previsto. Após finalizar a implementação do código para a prototipagem foi possível realizar teste referente a cada componente presente na prototipagem. Todos os componentes apresentaram comportamentos esperados quando testados de forma individual. Quanto ao sensor LDR foi decidido a aprimoração da prototipagem com relação a este sensor, trocando-o por um sensor de Leitura de Digital, não necessitando mais a simulação por meio do LDR. Quanto à interface mobile, foi implementada uma interface amigável, simples e intuitiva que funciona corretamente simulando o cadastro e consulta de usuários. A integração com Firebase foi finalizada permitindo o armazenamento de dados entre os dados fornecidos pela interface Flutter e atendendo as requisições feitas pela aplicação do ESP32.
- Recomendações: Antes das verificações deve-se sempre verificar se as ligações da prototipagem estão corretamente ligadas. Verifique também se a integração com o firebase ocorreu.