

**Documento de Requisitos do Sistema
Gerenciamento Remoto de Temperatura**

versão4.5

Autores:

Jhonatan Rafael de Souza
Marcelo Benigno Benigno Feltran
Pablo Eduardo Gimenez Nunes
Rafael Augusto Camargo Pires
Sérgio Hiroshi Nonomura

[illegible]

Conteúdo

1. INTRODUÇÃO	4
1.1 CONVENÇÕES, TERMOS E ABREVIACÕES	4
1.1.1 Identificação dos requisitos.....	4
1.1.2 Prioridades dos requisitos.....	4
2. DESCRIÇÃO GERAL DO SISTEMA.....	5
2.1 ABRANGÊNCIA E SISTEMAS RELACIONADOS.....	5
3. REQUISITOS FUNCIONAIS (CASOS DE USO)	5
3.1 RF001 - CONFIGURAR REDE DE ACESSO WIFI DO MICROCONTROLADOR ESP32.....	5
3.2 RF002 - CRIAR O DATABASE NO FIREBASE	7
3.3 RF003 - ABRIR E FECHAR AUTOMATICAMENTE AS JANELAS	9
3.4 RF004 - LIGAR E DESLIGAR AUTOMATICAMENTE OS VENTILADORES	11
3.5 RF005 - LIGAR E DESLIGAR AUTOMATICAMENTE OS AQUECEDORES	13
3.6 RF006 - MUDANÇA DO CONTROLE AUTOMÁTICO PARA MANUAL E VICE-VERSA.....	15
4. REQUISITOS NÃO-FUNCIONAIS	17
[NF001] Usabilidade.....	17
[NF002] Desempenho	17
[NF003] Hardware e Software	17

1. Introdução

1.1 Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

1.1.1 Identificação dos requisitos

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos, seguidos do identificador do requisito, de acordo com a especificação a seguir:

[nome da subseção. identificador do requisito]

Por exemplo, o requisito funcional [Recuperação de dados.RF016] deve estar descrito em uma subseção chamada “Recuperação de dados”, em um bloco identificado pelo número [RF016]. Já o requisito não-funcional [Confiabilidade.NF008] deve estar descrito na seção de requisitos não-funcionais de Confiabilidade, em um bloco identificado por [NF008].

Os requisitos devem ser identificados com um identificador único. A numeração inicia com o identificador [RF001] ou [NF001] e prossegue sendo incrementada à medida que forem surgindo novos requisitos.

1.1.2 Prioridades dos requisitos

Para estabelecer a prioridade dos requisitos, nas seções 4 e 5, foram adotadas as denominações “essencial”, “importante” e “desejável”.

1. **Essencial** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
2. **Importante** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
3. **Desejável** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

2. Descrição geral do sistema

2.1 Abrangência e sistemas relacionados

O gerenciamento Remoto de Temperatura permitirá via internet obter informações da temperatura de um ambiente residencial. Permitirá controlar a temperatura do ambiente residencial e possibilitar mesmo à distância poder controlar dispositivos que possam estabelecer a temperatura desejada.

3. Requisitos funcionais (casos de uso)

3.1 RF001 - Configurar rede de acesso wifi do microcontrolador ESP32

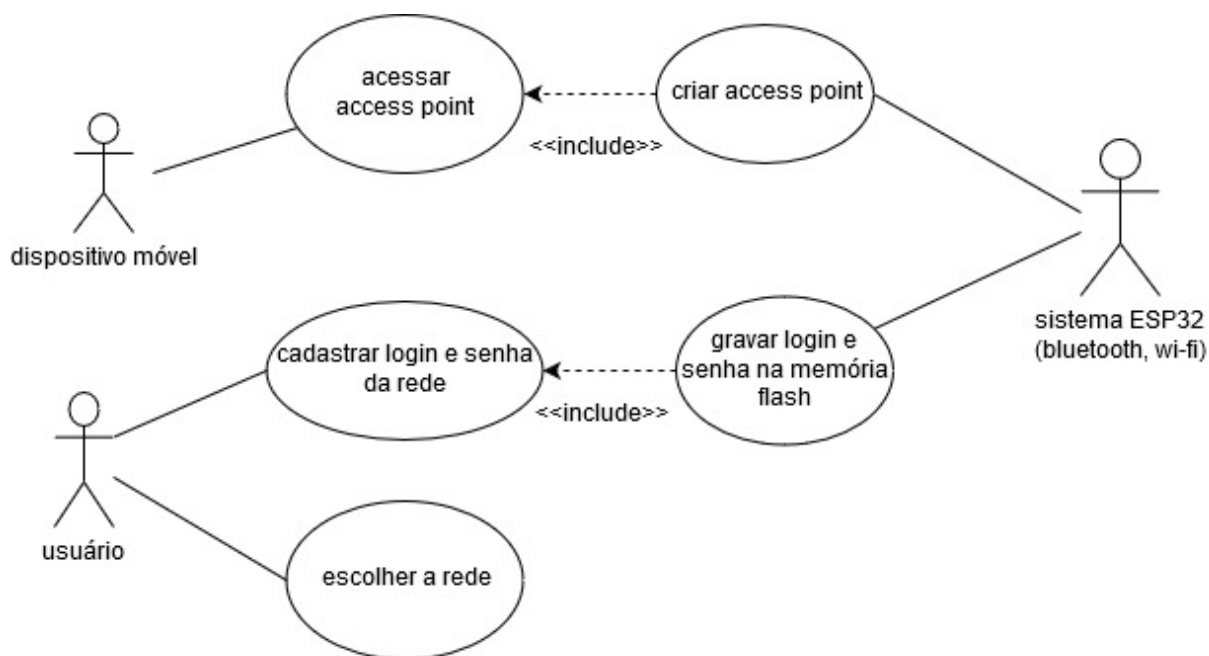


figura -diagrama de caso de uso - RF001- Configurar rede de acesso wifi do microcontrolador ESP32

Descrição:

Configuração da rede de acesso wifi do microcontrolador ESP32.

Eventos:

- criação de uma Access Point (AP);
- seleção de rede wifi a ser utilizada pelo ESP32;
- gravação da rede wifi selecionada na Flash Memory.

Atores:

Usuário
Dispositivo móvel
Sistema ESP32

Pré-Condições:

- o ESP32 estar ligado
- dispositivo móvel ligado e com navegador instalado
- rede wifi conectado a internet
- nome da rede (ssid) e senha conhecidos

Pós-Condições:

1. Conclusões com sucesso:
ESP32 com SSID e senha gravada na flash memory
2. Conclusões sem sucesso:
ESP32 mostrando falha de comunicação

Fluxo básico:

1. Criar um AP-Acess Point .
 - 1.1. ESP32 é resetado;
 - 1.2. ESP32 cria um AP cujo o nome da rede é ESP_AP (A1)
 - 1.3. ESP32 envia IP padrão 192.168.4.1
2. Acessar a rede ESP_AP e IP padrão 192.168.4.1
 - 2.1. Usuário acessa a rede ESP_AP e conecta ao IP padrão 192.168.4.1
 - 2.2. Tela de configuração WifiManager é disponibilizada
3. Selecionar a rede wifi desejada
 - 3.1. Usuário seleciona o botão Configura Wifi
 - 3.2. Usuário seleciona a rede wifi desejada
4. Cadastrar login e senha da rede selecionada
 - 4.1. Usuário informa login e senha da rede selecionada
 - 4.2. Usuário tecla botão de salvar
5. Gravar login e senha na Memória Flash
 - 5.1. ESP32 grava na memória flash o login e senha da rede
6. Fim do caso de uso

Fluxos alternativos:

- A1. O ESP32 não consegue criar um AP-Access Point
1. Envia mensagem "Falha ao conectar"
 2. ESP32 reinicializa

3.2 RF002 - Criar o database no Firebase

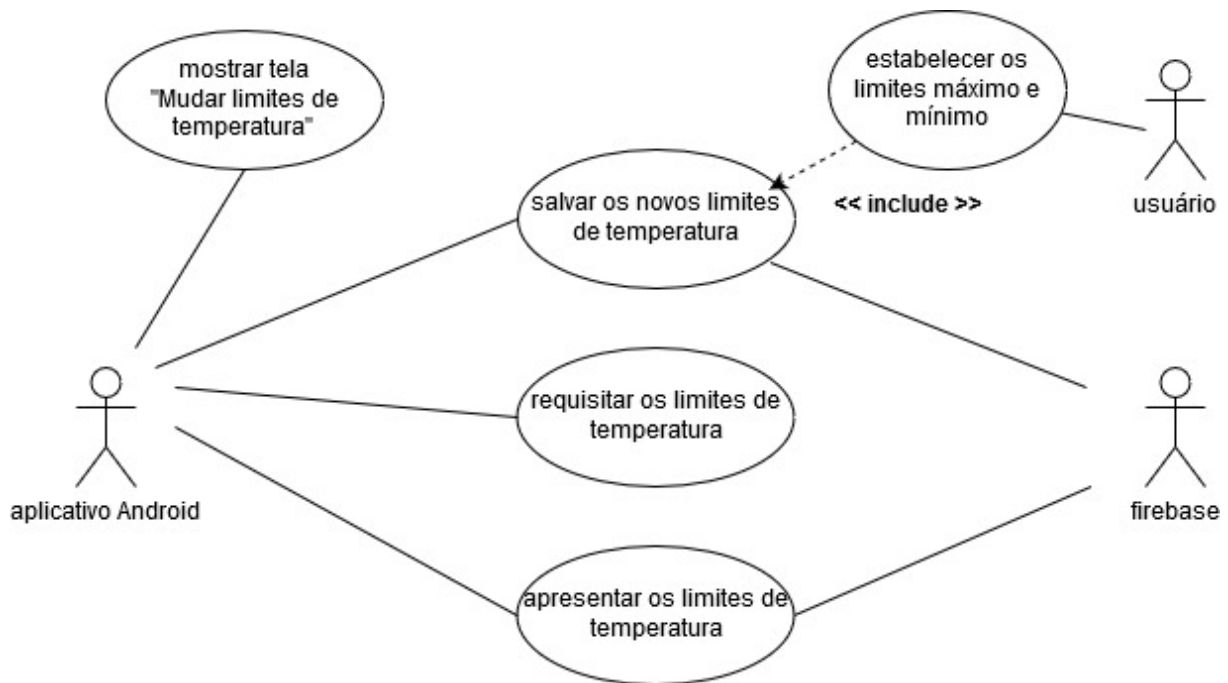


figura -diagrama de caso de uso - RF002- Criar o database no Firebase

Descrição

Permitir o usuário acessar o Firebase, criar uma database que será populada e consumida pelo ESP32.

Eventos:

- Acesso e autenticação do ESP32 ao Firebase.
- Criação do database JSON
- Definição da lista de pares JSON
- Leitura do MacAddress do ESP32

Atores:

ESP32
Firebase
Aplicativo Android

Pré-Condições:

O ESP32 deve estar conectado à internet.
O host e chave de autenticação do firebase já estar cadastrada no aplicativo.

Pós-Condições:

1. Conclusões com sucesso:
O banco de dados será criado no Firebase e terá definido a lista de pares JSON para o MacAddress do dispositivo ESP32.
2. Conclusões sem sucesso:

Não será criado o database no Firebase.

Fluxo básico:

1. Mostrar a tela "Mudar limites de temperatura"
 - 1.1. O usuário acessa a tela "Mudar limites de temperatura" do aplicativo Android
2. Definir limites mínimo e máximo
 - 2.1. O usuário define os novos valores para limites mínimo e máximo
3. Salvar limites de temperatura
 - 3.1. O usuário clica no botão "Salvar" (A1)
 - 3.2. O aplicativo envia os novos valores definidos para o Firebase
 - 3.3. O aplicativo mostra a mensagem "Novos limites definidos"
 - 3.4. O aplicativo volta para a tela inicial
4. O caso de uso é encerrado

Fluxos alternativos:

- A1. A base de dados não existe
1. O ESP32 envia as chaves primárias da base de dados
 2. O ESP32 preenche as chaves primárias com as temperaturas definidas
 3. O aplicativo mostra a mensagem "Novos limites definidos"
 4. O aplicativo volta para a tela inicial
 5. O caso de uso é encerrado

3.3 RF003 - Abrir e fechar automaticamente as janelas

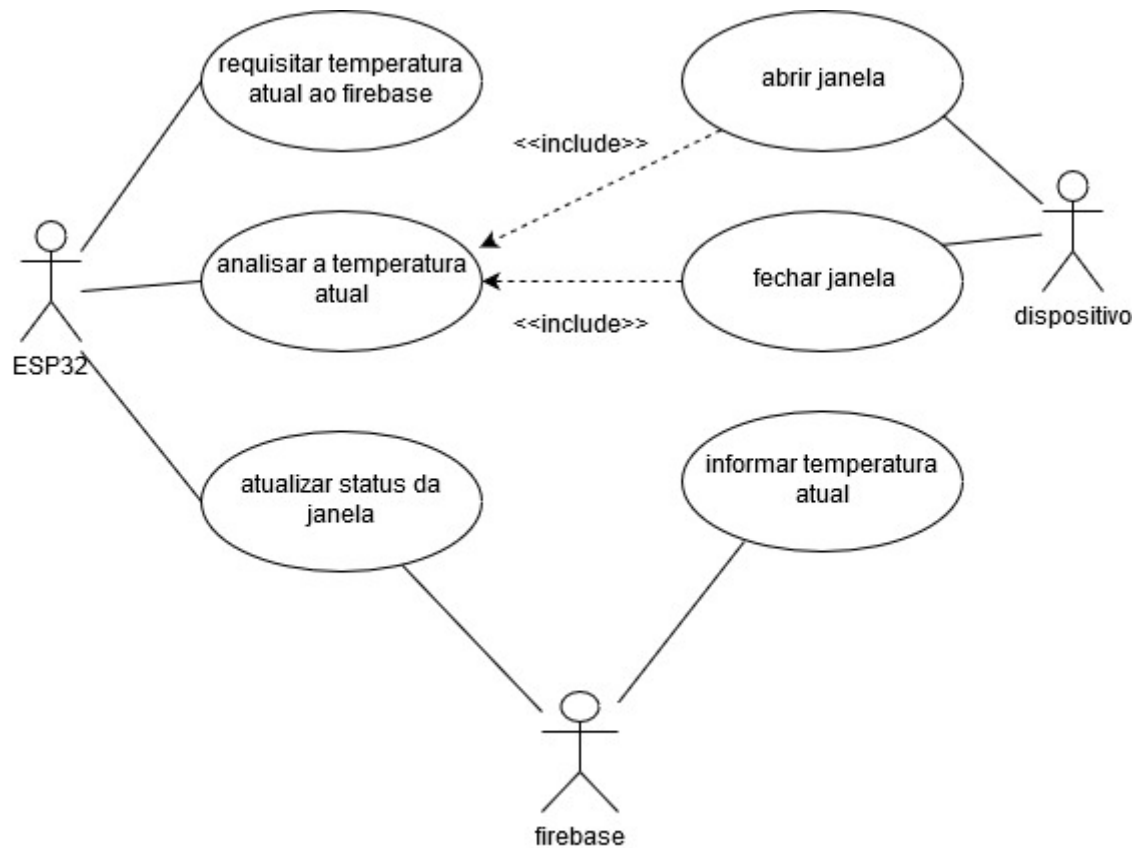


figura - diagrama de caso de uso - RF003- Abrir e Fechar automaticamente as janelas

Descrição:

Esse caso de uso demonstra o comportamento do sistema com relação à atuação sobre as janelas, caso a temperatura fique fora dos limites mínimo e máximo definidos. Caso a temperatura desejada não seja atingida, aciona-se os ventiladores.

Eventos:

Detecção de temperatura fora do intervalo definido.
Janela é fechada.
Janela é aberta.

Atores:

Sistema EPS32
Dispositivos (fecham/abrem as janelas) por meio de relês.
Firebase

Pré-Condições:

Deve existir uma configuração de temperatura mínima e máxima.
A chave "automaticostatus" do firebase é true (automático).

Pós-Condições:

1. Conclusões com sucesso:

As janelas serão fechadas ou abertas até que a temperatura do ambiente esteja dentro ou fora do intervalo de temperaturas mínimo e máximo.

2. Conclusões sem sucesso:

Caso RF4 - Ligar e desligar automaticamente os ventiladores é acionado.

Fluxo básico:

1. Requisitar a temperatura atual
 - 1.1. O sistema EPS32 requisita a temperatura atual ao firebase.
2. Analisar se temperatura atual está dentro do limite
 - 2.1. O sistema EPS32 verifica se a temperatura está no limite definido (A1)(A2)
3. Comandar relê do janela
 - 3.1. O sistema ESP32 comanda o relê do janela
4. Atualizar o firebase
 - 4.1. O sistema ESP32 atualiza o status no firebase
5. O caso de uso é encerrado

Fluxos alternativos:

- A1. A temperatura está acima do limite máximo. Acionar a abertura da janela
1. O relê realiza a abertura da janela (B1)
- B1. Mudar o status da janela no firebase
1. O sistema ESP32 muda o status da janela para aberta no firebase.
 2. O caso de uso é encerrado
- A2. A temperatura está abaixo do limite mínimo ou está chovendo. Acionar fechamento da janela
1. O relê realiza o fechamento da janela (B2).
- B2. Mudar o status da janela no firebase
2. O sistema ESP32 muda o status da janela para fechado no firebase.
 3. O caso de uso é encerrado

3.4 RF004 - Ligar e desligar automaticamente os ventiladores

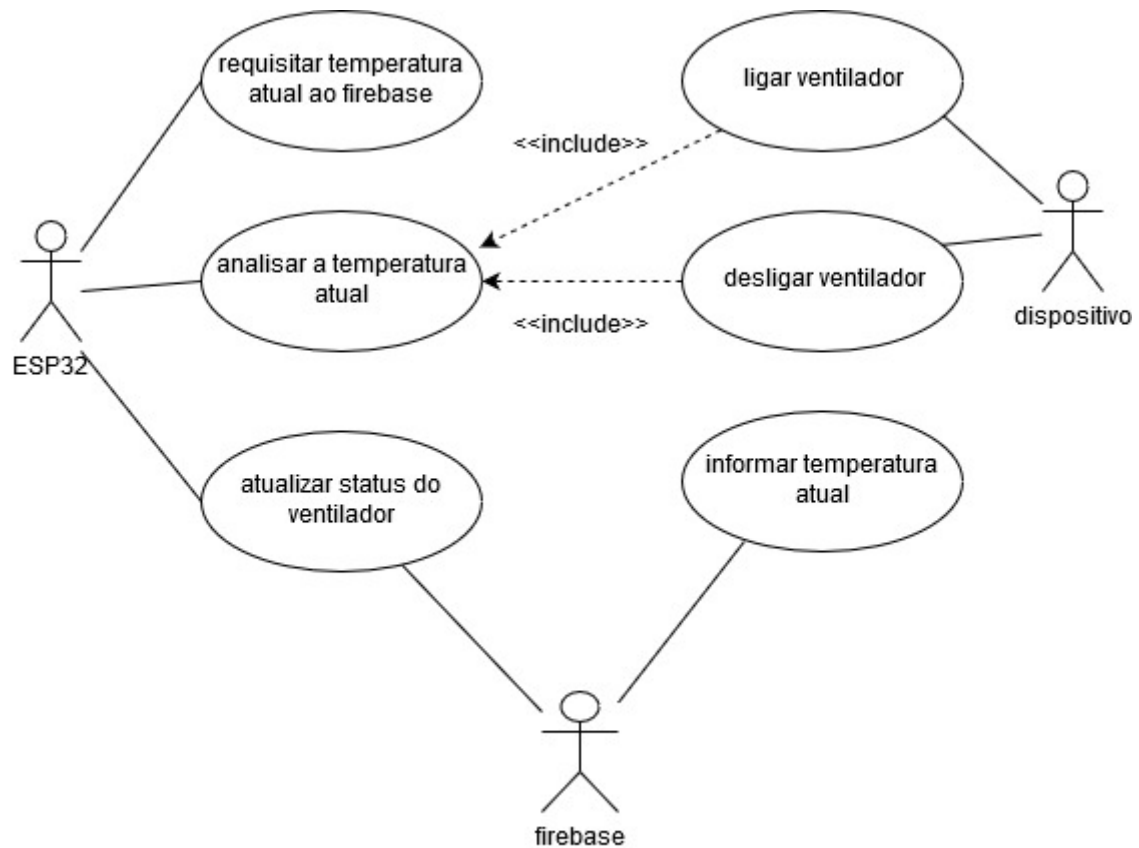


figura - diagrama de caso de uso - RF004- Ligar e Desligar automaticamente os ventiladores

Descrição:

Esse caso de uso demonstra o comportamento do sistema com relação à atuação do ventilador, caso a janela já esteja aberta e a temperatura continue fora dos limites mínimo e máximo definidos.

Eventos:

- Detecção de temperatura fora do intervalo definido
- Ventilador é ligado
- Ventilador é desligado

Atores:

- Sistema ESP32
- Dispositivo liga e desliga o ventilador por meio de relê.
- Firebase

Pré-Condições:

- Deve existir uma configuração de temperatura mínima e máxima.
- A chave "ventiladorautomatico" do firebase é true (automático).

Pós-Condições:

1. Conclusões com sucesso:

O ventilador será ligado/desligado caso a temperatura atual seja maior/menor que a máxima, respectivamente.

2. Conclusões sem sucesso:

O caso de uso é encerrado.

Fluxo básico:

1. Requisitar a temperatura atual
 - 1.1. O sistema EPS32 requisita a temperatura atual ao firebase.
2. Analisar se temperatura atual está dentro do limite
 - 2.1. O sistema EPS32 verifica se a temperatura está no limite definido (A1)(A2)
3. Comandar relê do ventilador
 - 3.1. O sistema EPS32 comanda o relê do ventilador
4. Atualizar o firebase
 - 4.1. O sistema ESP32 atualiza status no firebase
5. O caso de uso é encerrado

Fluxos alternativos:

- A1. A temperatura está acima do limite máximo. Ligar o ventilador.
1. O relê liga o ventilador (B1).
- B1. Mudar o status do ventilador no firebase
1. O sistema ESP32 muda o status do ventilador para ligado no firebase.
 2. O caso de uso é encerrado
- A2. A temperatura está abaixo do limite mínimo. Desligar o ventilador.
1. O relê realiza o fechamento da janela (B2).
- B2. Mudar o status da janela no firebase
1. O sistema ESP32 muda o status do ventilador para desligado no firebase.
 2. O caso de uso é encerrado

3.5 RF005 - Ligar e desligar automaticamente os aquecedores

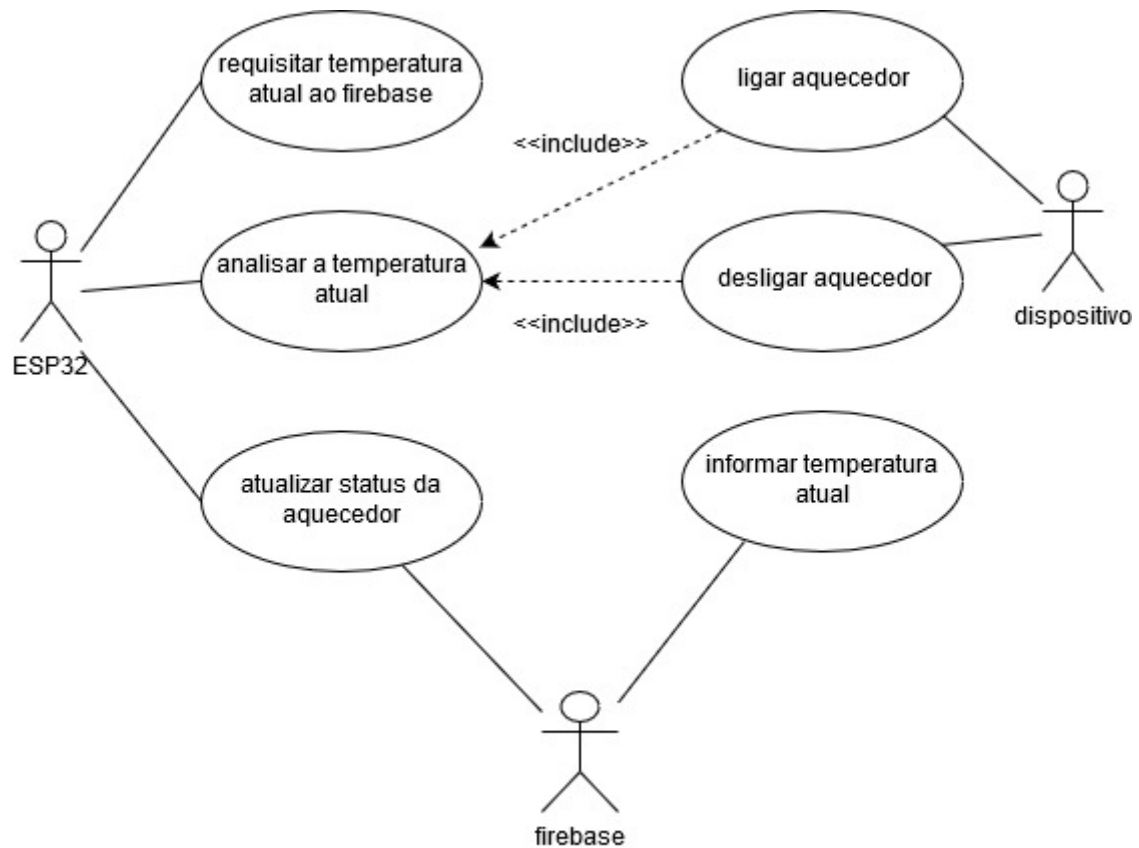


figura - diagrama de caso de uso - RF005- Ligar e desligar automaticamente os aquecedores

Descrição:

Esse caso de uso demonstra o comportamento do sistema com relação à atuação do aquecedor quando a temperatura continua fora dos limites mínimo e máximo definidos.

Eventos:

- Detecção de temperatura fora do intervalo definido
- Aquecedor é ligado
- Aquecedor é desligado

Atores:

- Sistema ESP32
- Dispositivo (ligam e desligam aquecedores)
- Firebase

Pré-Condições:

- Deve existir uma configuração de temperatura mínima e máxima.
- A chave "aquecedorautomatico" do firebase é true (automático).
- A janela está fechada e ventilador desligado, se a temperatura está abaixo da mínima desejada.

A janela está aberta e ventilador ligado, se a temperatura está acima da máxima desejada.

Pós-Condições:

1. Conclusões com sucesso:
O aquecedor será ligado/desligado caso a temperatura atual seja menor/maior que a mínima.
2. Conclusões sem sucesso:
O caso de uso é encerrado.

Fluxo básico:

1. Requisitar a temperatura atual
 - 1.1. O sistema EPS32 requisita a temperatura atual ao firebase.
2. Analisar se temperatura atual está dentro do limite
 - 2.1. O sistema EPS32 verifica se a temperatura está no limite definido (A1)(A2)
3. Comandar relê do aquecedor
 - 3.1. O sistema EPS32 comanda o relê do aquecedor .
4. Atualizar o firebase
 - 4.1. O sistema ESP32 atualiza status no firebase
5. O caso de uso é encerrado

Fluxos alternativos:

- A1. A temperatura está acima do limite máximo. Desligar o aquecedor.
1. O relê desliga o aquecedor (B1).
- B1. Mudar o status do aquecedor no firebase
1. O sistema ESP32 muda o status do aquecedor para desligado no firebase.
 2. O caso de uso é encerrado
- A2. A temperatura está abaixo do limite mínimo. Ligar o aquecedor.
1. O relê liga o aquecedor (B2).
- B2. Mudar o status da janela no firebase
1. O sistema ESP32 muda o status do aquecedor para ligado no firebase.
 2. O caso de uso é encerrado

3.6 RF006 - mudança do controle automático para manual e vice-versa

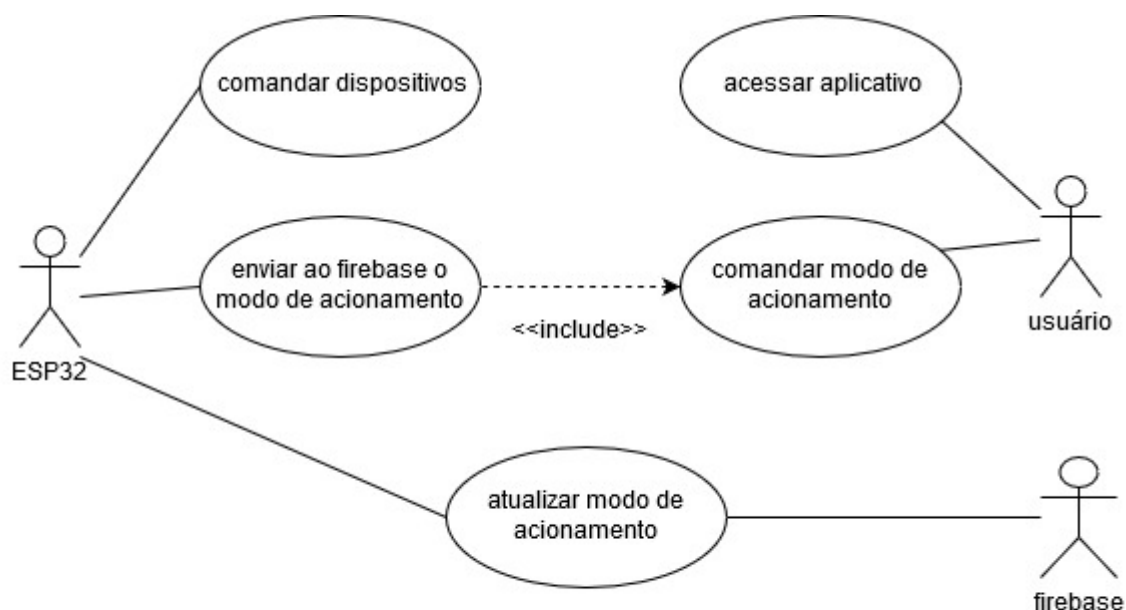


figura -diagrama de caso de uso - RF006 - mudança do controle automático para manual e vice-versa

Descrição:

Esse caso de uso demonstra o usuário mudar o status de automático para manual.

Eventos:

Mudança do controle do modo automático para manual.

Mudança de status dos dispositivos acionadores no firebase.

Atores:

Usuário

Aplicativo Android

ESP32

Firebase

Pré-Condições:

O sistema deve estar conectado à internet [RF001].

O usuário estar autenticado no firebase.

O sistema ESP32 estar ligado.

Pós-Condições:

1. Conclusões com sucesso:

Os dispositivos são configurados conforme atuação do usuário do modo automático para manual e vice-versa

2. Conclusões sem sucesso:
O caso de uso retorna ao início

Fluxo básico:

1. Acessar o aplicativo Android
 - 1.1. O usuário acessa o aplicativo
2. Configurar o modo de acionamento dos dispositivos
 - 2.1. Usuário altera o modo de atuação dos dispositivos (A1)(A2).
3. Atualizar novo modo de acionamento
 - 3.1. O sistema ESP32 envia ao firebase novo modo de acionamento dos dispositivos (B1)(B2)
 - 3.2. Firebase atualiza o novo modo de acionamento dos dispositivos no database. (C1)(C2)
4. Comandar os dispositivos conforme novo modo de acionamento
 - 4.1. O sistema ESP32 comandas os dispositivos conforme novo modo de acionamento (D1)(D2).
5. O caso de uso é encerrado

Fluxos alternativos:

- A1. O modo de atuação dos dispositivos passa de automático para manual
 1. O usuário muda o modo de atuação dos dispositivos de automático para manual.
- B1. Enviar novo status de acionamento ao firebase
 1. O sistema ESP32 envia o novo status (false) dos dispositivos no firebase.
- C1. Atualizar novo status de acionamento no firebase
 1. O firebase atualiza o novo status (false) dos dispositivos no database.
- D1. Comandar dispositivos manualmente
 1. O sistema ESP32 aguarda comando do usuário.
- A2. O modo de atuação dos dispositivos passa de manual para automático
 1. O usuário muda o modo de atuação dos dispositivos de manual para automático.
- B2. Enviar novo status de acionamento ao firebase
 1. O sistema ESP32 envia o novo status (true) dos dispositivos no firebase.
- C2. Atualizar novo status de acionamento no firebase
 1. O firebase atualiza o novo status (true) dos dispositivos no database.
- D2. Comandar dispositivos automaticamente
 1. O sistema ESP32 reinicia o processo.

4. Requisitos não-funcionais

[NF001] Usabilidade

A interface com o usuário é de vital importância para o sucesso do sistema.

O sistema terá uma interface direta e amigável ao usuário primário sem se tornar cansativa aos usuários mais experientes.

Prioridade: ☒ Essencial ☐ Importante ☐ Desejável

[NF002] Desempenho

Embora não seja um requisito essencial ao sistema, deve ser considerada por corresponder a um fator de qualidade de software.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável

[NF003] Hardware e Software

Visando criar um produto com maior extensibilidade, reusabilidade e flexibilidade, deve-se adotar como linguagem principal de desenvolvimento Java (plataforma Android) seguindo cuidadosamente as técnicas de orientação a objetos no desenvolvimento no aplicativo móvel.

O uso da linguagem Java não especifica qual será o sistema operacional e a máquina em que o programa irá executar. No entanto, esse aplicativo Android deverá se comunicar com o sistema de banco de dados Firebase.

O ESP32 deverá ser programado em linguagem "C++" própria do Arduino.

Prioridade: ☐ Essencial ☒ Importante ☐ Desejável