

# **Documento de Requisitos do Sistema**

## ***Adestrador Eletrônico para cães - AdestraKit***

**Versão 1.0**

**Autores:**

Aroldo Nicácio da Silva  
Elaine Breda M. Schwaner  
Erivelton Ribeiro Pinto  
Raiff Sales da Fonseca

### Histórico de Alterações

[illegible]

## Conteúdo

<b>1. Introdução</b>	<b>5</b>
1.1 Convenções, termos e abreviações	5
1.1.1 Identificação dos requisitos	5
1.1.2 Prioridades dos requisitos	5
<b>2. Descrição geral do sistema</b>	<b>6</b>
2.1 Abrangência e sistemas relacionados	6
<b>3. Requisitos funcionais (casos de uso)</b>	<b>6</b>
3.1 Aplicativo AdestraKit	6
[RF001] Cadastrar Usuário / Login	6
[RF002] Cadastrar microcontrolador	6
[RF003] Configurar microcontrolador	7
[RF004] Excluir microcontrolador	7
[RF005] Configurar distância	7
[RF006] Visualizar relatório	8
[RF007] Controlar o microcontrolador	8
3.2 Microcontrolador ESP32	8
[RF008] Monitorar Distância	8
[RF009] Emitir sinal luminoso	9
[RF010] Emitir sinal sonoro	9
3.3 Firebase	9
[RF011] Receber conexões	9
<b>4. Requisitos não-funcionais</b>	<b>9</b>
[NF001] Usabilidade	10
[NF002] Acesso a Internet	10
[NF003] Desempenho	10
[NF004] Confiabilidade	11
<b>5. Descrição dos Casos de Usos</b>	<b>11</b>
5.1 Cadastrar Usuário / Login	11
5.2 Cadastrar microcontrolador	12
5.3 Configurar microcontrolador	13
5.4 Excluir microcontrolador	13
5.5 Configurar distância	14
5.6 Visualizar relatório	15
5.7 Controlar o microcontrolador	16
5.8 Monitorar Distância	17

5.9 Emitir sinal luminoso	18
5.10 Emitir sinal sonoro	19
5.11 Receber conexões	20
<b>6. Diagramas de Casos de Uso</b>	<b>22</b>
UC01 - Casos de Uso	22
<b>7. Diagramas de Classe</b>	<b>23</b>
7.1 Diagrama de Classe	23
<b>8. Diagrama Comportamental</b>	<b>24</b>
8.1 Diagrama de atividades	24
<b>9. Diagramas de Implantação</b>	<b>25</b>
9.1 Diagrama de componentes	25
9.2 Diagrama de implementação	25
<b>10. Acompanhamento do cronograma</b>	<b>26</b>
<b>11. Dificuldades</b>	<b>27</b>
<b>12. Atividades futuras</b>	<b>27</b>

## 1. Introdução

Projeto desenvolvido com objetivo de criar um sistema de adestramento para cães, sendo possível ao proprietário do animal colocar o aparelho em locais de sua preferência e a partir deste momento definir distâncias que o animal possa chegar, sendo que ao atingir este limite de proximidade o sistema irá gerar um ruído ultrassônico capaz de afastar o animal da área definida pelo dono.

### 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 sistema terá sua estrutura dividida em três partes que colaboram e trocam informações entre si. Para tanto está sendo criado um aplicativo para que o usuário possa ter informações e histórico da ativação dos sensores nos locais que tenha deixado antecipadamente, inicialmente o aplicativo será disponibilizado na versão Android e na sequência será disponibilizado também na versão para Iphone, um microcontrolador ESP32 programado em C com sensor ultrassônico de distância, um buzzer para emitir um sinal que consiga afastar o animal de estimação da área a ser preservada pelo dispositivo. E para controlar estas informações e funcionar como repositório de dados, teremos o Firebase fazendo esta integração.

O microcontrolador será colocado em algum lugar escolhido pelo proprietário, sendo que então este proprietário vai definir a distância em que não quer aproximação do animal e com isso caso haja o avanço do animal nesta área, será enviado os dados para o repositório do Firebase, onde estes dados serão acessados pelo aplicativo e poderão auxiliar o proprietário do animal no controle que se deseja.

## 3. Requisitos funcionais (casos de uso)

### 3.1 Aplicativo AdestraKit

#### [RF001] Cadastrar Usuário / Login

**Descrição do caso de uso:** Este caso de uso permite que o usuário possa realizar seu cadastro pessoal e com isso conseguir acessar e controlar os controladores que desejar.

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

**Entradas e pré-condições:** smartphone do usuário com acesso à internet para efetuar o cadastro ou login nos bancos de dados da aplicação.

**Saídas e pós-condição:** usuário devidamente cadastrado e com acesso às funcionalidades do aplicativo.

#### [RF002] Cadastrar microcontrolador

**Descrição do caso de uso:** Este caso de uso permite que o usuário cadastre e gerencie o seu microcontrolador, podendo definir distâncias seguras de forma individualizada.

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

**Entradas e pré-condições:** não tem.

**Saídas e pós-condição:** um componente é cadastrado no sistema

[RF003] Configurar microcontrolador
-------------------------------------

**Descrição do caso de uso:** Este caso de uso permite que o usuário conecte ao microcontrolador por meio de bluetooth e faça a configuração da rede Wifi que o microcontrolador usará para o envio dos dados para o repositório.

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

**Entradas e pré-condições:** bluetooth ativo no microcontrolador e no smartphone.

**Saídas e pós-condição:** realizado a configuração da rede wifi como desejado pelo usuário.

[RF004] Excluir microcontrolador
----------------------------------

**Descrição do caso de uso:** Este caso de uso permite que o usuário retire de seu cadastro um microcontrolador que não esteja utilizando.

**Prioridade:**     ☐ Essencial            ☒ Importante            ☐ Desejável

**Entradas e pré-condições:** recebe como entrada o microcontrolador que se deseja excluir.

**Saídas e pós-condição:** o usuário consegue excluir o microcontrolador que solicitou.

[RF005] Configurar distância
------------------------------

**Descrição do caso de uso:** Este caso de uso permite que o usuário defina a distância na qual o sistema de adestramento será acionado.

**Prioridade:**     ☐ Essencial            ☒ Importante            ☐ Desejável

**Entradas e pré-condições:** recebe como entrada o valor em centímetros.

**Saídas e pós-condição:** o controlador é acionado quando a distância é ultrapassada.

#### [RF006] Visualizar relatório

**Descrição do caso de uso:** Este caso de uso permite que o usuário visualize relatórios individuais a respeito dos acionamentos de cada microcontrolador.

**Prioridade:**      ☐ Essencial                      ☒ Importante                      ☐ Desejável

**Entradas e pré-condições:** recebe como entrada o microcontrolador que se deseja gerar o relatório.

**Saídas e pós-condição:** o aplicativo fornece ao usuário um relatório com os dados solicitados.

#### [RF007] Controlar o microcontrolador

**Descrição do caso de uso:** Este caso de uso permite que o usuário altere o status de cada microcontrolador, deixando-o ligado ou desligado.

**Prioridade:**      ☐ Essencial                      ☒ Importante                      ☐ Desejável

**Entradas e pré-condições:** recebe como entrada o microcontrolador e o status que deseja que o microcontrolador assuma.

**Saídas e pós-condição:** o microcontrolador altera seu status, de acordo com o que foi pedido.

## 3.2 Microcontrolador ESP32

#### [RF008] Monitorar Distância

**Descrição do caso de uso:** Este caso de uso o microcontrolador permanece com o sensor ultrassônico ligado em seu campo de ação, monitorando qualquer entrada neste seu campo de controle e caso haja violação de distância definida enviará os dados para o Firebase.

**Prioridade:**      ☒ Essencial                      ☐ Importante                      ☐ Desejável

**Entradas e pré-condições:** deve manter seu sensor operante aguardando por interferências, conectado à internet.



**Saídas e pós-condição:** envio de dados ao banco de dados.

[RF009] Emitir sinal luminoso
-------------------------------

**Descrição do caso de uso:** Neste caso de uso o microcontrolador após a ultrapassagem do limite, aciona um sinal luminoso de alerta.

**Prioridade:**     ☒ Essencial                      ☐ Importante                      ☐ Desejável

**Entradas e pré-condições:** sensor ultrasônico operando e limite ultrapassado.

**Saídas e pós-condição:** ativação de led.

[RF010] Emitir sinal sonoro
-----------------------------

**Descrição do caso de uso:** Neste caso de uso o microcontrolador após 3 segundos da emissão do sinal luminoso, aciona um sinal sonoro.

**Prioridade:**     ☒ Essencial                      ☐ Importante                      ☐ Desejável

**Entradas e pré-condições:** sensor ultrasônico operando, limite ultrapassado e led .

**Saídas e pós-condição:** ativação de Buzzer.

### 3.3    Firebase

[RF011] Receber conexões
--------------------------

**Descrição do caso de uso:** Este caso de uso permite que o Firebase permaneça ativo e aguardando as conexões para o envio e recebimento de dados tanto dos aplicativos quanto dos microcontroladores, atuando como repositório de dados.

**Prioridade:**     ☒ Essencial                      ☐ Importante                      ☐ Desejável

**Entradas e pré-condições:** plataforma disponível na internet e atendimento às regras de segurança dos componentes.

**Saídas e pós-condição:** o usuário visualiza o componente desejado

<b>4.    Requisitos não-funcionais</b>
--

## 4.1 Aplicativo

### [NF001] Usabilidade

A interface com o usuário é de vital importância para o sucesso do sistema, propiciando ao usuário a gestão dos microcontroladores, bem como acesso aos relatórios desejados, esta parte tem que ser intuitiva tendo em vista não ter treinamento disponível aos usuários.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

### [NF002] Acesso a Internet

Inicialmente para a instalação no smartphone do usuário o acesso a internet é imprescindível para a obtenção do aplicativo por download, o qual deve estar disponível nas lojas para que o software seja instalado em dispositivos móveis.

Com o aplicativo devidamente instalado, necessita de acesso integral à internet, para conseguir realizar o login, para acessar os dados, visualizar relatórios e configurar distância.

Na outra ponta do sistema, teremos o microcontrolador que também deve ter acesso a uma rede wifi local com acesso a internet para operar na sua integralidade, controlando a distância e realizando o envio dos dados para o repositório de forma a serem acessados e usados pelos usuários, bem como ter seus parâmetros ajustados da maneira que o usuário desejar.

E como integrador de todo o sistema teremos a plataforma do Firebase, o qual só pode ser acessado via internet, portanto deve estar disponível para o perfeito funcionamento do sistema.

**Prioridade:**    ☒ Essencial            ☐ Importante            ☐ Desejável

### [NF003] Desempenho

O sistema tem como dependência os dados coletados pelos microcontroladores, os quais serão encaminhados para o repositório do Firebase, e como a aplicação dos microcontroladores será no monitoramento de animais de estimação, o microcontrolador necessita de execução constante, como poder detectar qualquer movimentação dentro do seu campo de ação, tomar as medidas definidas pelos usuários do sistema, e levando em conta que o controle será de animais de estimação, portanto necessita que opere em sempre ativo sem demora nas respostas e que consiga executar as medidas de educação do animal.

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

[NF004] Confiabilidade
------------------------

O repositório do Firebase deverá permanecer disponível e manter os dados seguros, a fim de serem acessados apenas pelos seus respectivos usuários, e que possam ser recuperadas as informações assim que o usuário desejar.

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

<b>5.      Descrição dos Casos de Usos</b>
--

### **5.1    Cadastrar Usuário / Login**

**Nome do Caso de Uso:** UC01 - Cadastrar Usuário / Login.

**Descrição:**

Este caso de uso inicia com o usuário na tela de Login do aplicativo.

**Eventos:**

- O usuário efetuará o Login no aplicativo.

**Atores:**

- Usuário.  
- Firebase.

**Pré-Condições:**

- Aparelho com acesso à internet e aplicativo instalado em seu aparelho.

**Pós-Condições:**

**1. Conclusões com sucesso:**

- Usuário deve preencher os campos informados e ter um cadastro válido salvo no Firebase.

**2. Conclusões sem sucesso:**

- O usuário não preenche os campos informados.  
- Os dados do usuário não estão cadastrados

**Fluxo básico:**

1. Usuário preenche o campo login.
2. Usuário preenche o campo senha.
3. Usuário clica no botão entrar.

**Fluxos alternativos:**

- A1. Caso usuário não tenha cadastro.
- A1.1. Será solicitado um email ao usuário para proceder o cadastro.
- A1.2. Usuário informa email válido.
- A1.3. Efetivado o cadastro do usuário.

## 5.2 Cadastrar microcontrolador

**Nome do Caso de Uso:** UC02 - Cadastrar microcontrolador.

**Descrição:**

Este caso de uso inicia com o usuário na tela de Cadastro de microcontroladores.

**Eventos:**

- Usuário informará os dados do microcontrolador.

**Atores:**

- Usuário.
- Firebase.

**Pré-Condições:**

- Aparelho com acesso à internet e aplicativo instalado em seu aparelho.
- Login realizado.

**Pós-Condições:****1. Conclusões com sucesso:**

- Usuário deve preencher os campos solicitados.

**2. Conclusões sem sucesso:**

- O usuário não preenche os campos solicitados.

**Fluxo básico:**

1. Usuário preenche o campo nome do microcontrolador.
2. Usuário preenche o campo local de instalação do microcontrolador.
3. Usuário preenche o campo ativo / inativo.
4. Usuário clica no botão Salvar.

**Fluxos alternativos:**

- A1. Caso usuário não faça o cadastro poderá voltar clicando na seta de retorno.
- A1.2. Usuário será direcionado para a Tela Inicial.

### 5.3 Configurar microcontrolador

**Nome do Caso de Uso:** UC03 - Configurar microcontrolador

**Descrição:**

Este caso de uso destina-se a configuração inicial da rede wifi do microcontrolador, ao inicializar o microcontrolador o usuário fará o acesso a uma página onde deverá informar os dados a serem utilizados para a conexão local do microcontrolador na internet.

**Eventos:**

- Usuário informa os dados da rede.

**Atores:**

- Microcontrolador.
- Usuário.

**Pré-Condições:**

- Microcontrolador reinicializado, sem dados de conexão wifi,

**Pós-Condições:**

**1. Conclusões com sucesso:**

- microcontrolador conecta na rede desejada e fica online.

**2. Conclusões sem sucesso:**

- microcontrolador não consegue conectar na rede desejada

**Fluxo básico:**

1. Usuário conecta na página de configurações disponibilizada.
2. Usuário digita o nome da rede que deseja utilizar.
3. Usuário digita a senha da rede que deseja utilizar.
4. Usuário clica em Gravar dados;
5. Microcontrolador recebe as informações e se conectar na rede informada.

### 5.4 Excluir microcontrolador

**Nome do Caso de Uso:** UC04 - Excluir microcontrolador

**Descrição:**

Este caso de uso permite que o usuário retire de seu cadastro algum microcontrolador que não esteja mais utilizando.

**Eventos:**

- Usuário seleciona o microcontrolador que deseja excluir.

**Atores:**

- Usuário.
- Firebase.

**Pré-Condições:**

- microcontrolador cadastrado e presente na listagem do usuário.

**Pós-Condições:****1. Conclusões com sucesso:**

- exclusão do microcontrolador com sucesso.

**2. Conclusões sem sucesso:**

- microcontrolador continua sob o controle do usuário.

**Fluxo básico:**

1. Usuário seleciona o microcontrolador.
2. Usuário clica em opções do microcontrolador.
3. Usuário clica em excluir microcontrolador
4. Usuário clica em confirmar.

**Fluxos alternativos:**

A1: em “Usuário clica em confirmar”, o Usuário pode clicar em cancelar

A1.1. Usuário clica em cancelar

A1.2. Sistema apresenta a tela anterior e não efetua alteração do microcontrolador.

## 5.5 Configurar distância

**Nome do Caso de Uso:** UC5 - Configurar distância

**Descrição:**

Este caso de uso o Usuário tem a possibilidade de ajustar a distância que deve ser enviada ao microcontrolador para que mantenha as regras de controle por ele executadas.

**Eventos:**

- Usuário informa a distância limite.

**Atores:**

- Usuário.
- Firebase.
- Microcontrolador.

**Pré-Condições:**

- todos os atores com acesso à internet, e devidamente vinculados.

**Pós-Condições:****1. Conclusões com sucesso:**

- distância ajustada e microcontrolador utilizando este valor para o controle.

**2. Conclusões sem sucesso:**

- microcontrolador continua com o valor de distância anterior.

**Fluxo básico:**

1. Usuário seleciona o microcontrolador.
2. Usuário clica em opções do microcontrolador.
3. Usuário clica em configurar microcontrolador
4. Usuário clica em confirmar.
5. Aplicativo envia os novos valores para o Firebase.
6. Microcontrolador requisita novos valores para o Firebase e atualiza sua referência.

**Fluxos alternativos:**

A1: em “Usuário clica em confirmar”, o Usuário pode clicar em cancelar

A1.1. Usuário clica em cancelar

A1.2. Sistema apresenta a tela anterior e não efetua alteração de distância no microcontrolador.

## 5.6 Visualizar relatório

**Nome do Caso de Uso:** UC06 - Visualizar relatório

**Descrição:**

Este caso de uso o usuário obtém acesso aos dados do repositório do Firebase.

**Eventos:**

- Usuário solicita o relatório.
- Firebase retorna os dados.

**Atores:**

- Usuário.
- Firebase.

**Pré-Condições:**

- dados disponíveis no Firebase.

**Pós-Condições:****1. Conclusões com sucesso:**

- relatório gerado conforme solicitado.

**2. Conclusões sem sucesso:**

- o sistema exibe o relatório com os dados em branco.

**Fluxo básico:**

1. Usuário seleciona o microcontrolador.
2. Usuário clica em Gerar relatório.
3. Aplicativo recupera os dados do Firebase.
4. Firebase retorna os dados do microcontrolador.
6. Usuário visualiza os dados.

**Fluxos alternativos:**

A1: em "Aplicativo recupera os dados do Firebase", caso não haja dados disponíveis.

A1.1. Aplicativo exibe um relatório com os dados em branco.

## 5.7 Controlar o microcontrolador

**Nome do Caso de Uso:** UC07 - Controlar o microcontrolador

**Descrição:**

Este caso de uso o Usuário tem a possibilidade de alterar o status do microcontrolador, mantendo-o como inativo ou ativo.

**Eventos:**

- Usuário informa o novo status do microcontrolador.

**Atores:**

- Usuário.
- Firebase.
- Microcontrolador.

**Pré-Condições:**

- todos os atores com acesso à internet, e devidamente vinculados.

**Pós-Condições:****1. Conclusões com sucesso:**

- status ajustado e microcontrolador obedecendo ao que foi solicitado.

**2. Conclusões sem sucesso:**

- microcontrolador continua no status anterior.

**Fluxo básico:**

1. Usuário seleciona o microcontrolador.
2. Usuário clica em opções do microcontrolador.
3. Usuário clica em configurar o status do microcontrolador.
4. Usuário seleciona o novo status do microcontrolador.
5. Usuário clica em confirmar.
6. Aplicativo envia novo status para o Firebase.



7. Microcontrolador requisita novo status para o Firebase e atualiza sua referência.

**Fluxos alternativos:**

A1: em “Usuário clica em confirmar”, o Usuário pode clicar em cancelar

A1.1. Usuário clica em cancelar

A1.2. Sistema apresenta a tela anterior e não efetua alteração de distância no microcontrolador.

## **5.8 Monitorar Distância**

**Nome do Caso de Uso:** UC08 - Monitorar Distância

**Descrição:**

Este caso de uso o microcontrolador realiza a varredura de acordo com os parâmetros que constam no Firebase, e havendo movimentações neste período envia os dados para o controle do Firebase.

**Eventos:**

- microcontrolador realiza varredura e envia dados.

**Atores:**

- Microcontrolador.
- Firebase.
- Cão.

**Pré-Condições:**

- microcontrolador com acesso a internet.

**Pós-Condições:**

**1. Conclusões com sucesso:**

- dados enviados ao Firebase.

**2. Conclusões sem sucesso:**

- Firebase não recebe os dados

**Fluxo básico:**

1. Microcontrolador realiza varredura, com os parâmetros salvos.
2. Microcontrolador identifica presença física em sua área de varredura.
3. Microcontrolador aciona o sinal luminoso, como advertência.
4. Microcontrolador grava o horário inicial que acionou o sinal luminoso
5. Microcontrolador identifica avanço da área não permitida.
6. Microcontrolador aciona o sinal sonoro.
7. Microcontrolador grava o horário inicial que acionou o sinal sonoro.
8. Microcontrolador identifica a saída da área não permitida.
9. Microcontrolador desliga o sinal sonoro.
10. Microcontrolador desliga o sinal luminoso.

11. Microcontrolador grava o horário final que desligou o sinal sonoro e o luminoso.
12. Microcontrolador envia os dados ao Firebase.

**Fluxos alternativos:**

A1: em “Microcontrolador aciona o sinal luminoso, como advertência”, o cão pode sair da área de controle do microcontrolador.

A1.1. Cão sai da área de controle.

A1.2. Microcontrolador vai para o item “Microcontrolador desliga o sinal luminoso” e continua o processo.

## 5.9 Emitir sinal luminoso

**Nome do Caso de Uso:** UC09 - Emitir sinal luminoso

**Descrição:**

Este caso de uso o microcontrolador realiza a varredura de acordo com os parâmetros que constam no Firebase, e havendo movimentações neste período envia os dados para o controle do Firebase e aciona o led de advertência.

**Eventos:**

- microcontrolador realiza varredura e aciona o led, após envia dados ao Firebase.

**Atores:**

- Microcontrolador.
- LED.
- Firebase.

**Pré-Condições:**

- microcontrolador com acesso a internet.
- alimentação de energia para acionamento do led.

**Pós-Condições:****1. Conclusões com sucesso:**

- acionamento do led.
- dados enviados ao Firebase.

**2. Conclusões sem sucesso:**

- led não é acionado.
- Firebase não recebe os dados

**Fluxo básico:**

1. Microcontrolador realiza varredura, com os parâmetros salvos.
2. Microcontrolador identifica presença física em sua área de varredura.
3. Microcontrolador aciona o sinal luminoso, como advertência.
4. Microcontrolador grava o horário inicial que acionou o sinal luminoso.
5. Microcontrolador identifica avanço da área não permitida.
6. Microcontrolador envia os dados ao Firebase.

**Fluxos alternativos:**

A1:Não há.

**5.10 Emitir sinal sonoro**

**Nome do Caso de Uso:** UC10 - Emitir sinal sonoro

**Descrição:**

Este caso de uso o microcontrolador realiza a varredura de acordo com os parâmetros que constam no Firebase, e havendo movimentações neste período envia os dados para o controle do Firebase e aciona um sinal sonora para afastar o animal de estimação do local determinado.

**Eventos:**

- microcontrolador realiza varredura e aciona o buzzer com sinal sonoro em uma frequência audível somente para cães, após envia dados ao Firebase.

**Atores:**

- Microcontrolador.
- Buzzer.
- Firebase.

**Pré-Condições:**

- microcontrolador com acesso a internet.

**Pós-Condições:****1. Conclusões com sucesso:**

- acionamento do buzzer na frequência determinada.
- dados enviados ao Firebase.

**2. Conclusões sem sucesso:**

- buzzer não é acionado.
- Firebase não recebe os dados.

**Fluxo básico:**

1. Microcontrolador realiza varredura, com os parâmetros salvos.
2. Microcontrolador identifica presença física em sua área de varredura.
3. Microcontrolador identifica avanço da área não permitida.
4. Microcontrolador aciona o sinal sonoro.
5. Microcontrolador grava o horário inicial que acionou o sinal sonoro.
6. Microcontrolador identifica a saída da área não permitida.
7. Microcontrolador desliga o sinal sonoro.
8. Microcontrolador grava o horário final que desligou o sinal sonoro.
12. Microcontrolador envia os dados ao Firebase.

**Fluxos alternativos:**

A1: em “Microcontrolador aciona o sinal sonoro, para que o cão saía da área de controle do microcontrolador.

A1.1. Cão sai da área de controle.

A1.2. Microcontrolador vai para o item “Microcontrolador desliga o sinal sonoro” e continua o processo.

**5.11 Receber conexões**

**Nome do Caso de Uso:** UC11 - Receber conexões

**Descrição:**

Este caso de uso onde o Firebase permanece aguardando as conexões, tanto do aplicativo, quanto do Microcontrolador, respondendo às solicitações de cada um, as quais podem ser de gravação ou de consulta a dados.

**Eventos:**

- Firebase disponível.
- Microcontrolador solicita ou encaminha informações ao Firebase.
- Aplicativo solicita ou encaminha informações ao Firebase.

**Atores:**

- Firebase
- Aplicativo
- Microcontrolador

**Pré-Condições:**

- Firebase disponível na internet.
- Aplicativo com conexão de internet.
- Microcontrolador com conexão ativa de internet.

**Pós-Condições:****1. Conclusões com sucesso:**

- Firebase grava em seu repositório os dados encaminhados.
- Firebase fornece os dados solicitados.
- Microcontrolador recebe do Firebase os dados solicitados.
- Microcontrolador encaminha os dados para o Firebase.
- Aplicativo recebe do Firebase os dados solicitados.
- Aplicativo encaminha os dados para o Firebase.

**2. Conclusões sem sucesso:**

- Firebase não recebe os dados e não consegue gravar em repositório.
- Firebase não responde às solicitações e deixa de fornecer os dados solicitados.
- Microcontrolador não recebe do Firebase os dados solicitados.
- Microcontrolador não encaminha os dados para o Firebase.
- Aplicativo não recebe do Firebase os dados solicitados.
- Aplicativo não encaminha os dados para o Firebase.

**Fluxo básico:**

1. Microcontrolador conecta ao Firebase.
2. Microcontrolador envia dados para gravação no Firebase.
3. Firebase executa a gravação dos dados.
4. Aplicativo conecta ao Firebase.
5. Aplicativo envia dados para gravação no Firebase.
6. Firebase executa a gravação dos dados.

**Fluxos alternativos:**

- A1. em “Microcontrolador conecta ao Firebase”.
- A1.1. Microcontrolador solicita ao Firebase dados.
- A1.2. Firebase retorna os dados ao Microcontrolador.
- A1.3. Microcontrolador atualiza seus parâmetros.
- A2. em “Aplicativo conecta ao Firebase”.
- A2.1. Aplicativo solicita ao Firebase dados.
- A2.2.. Firebase retorna os dados ao Aplicativo .
- A2.3.. Aplicativo atualiza seus dados e exibe em tela ao usuário.

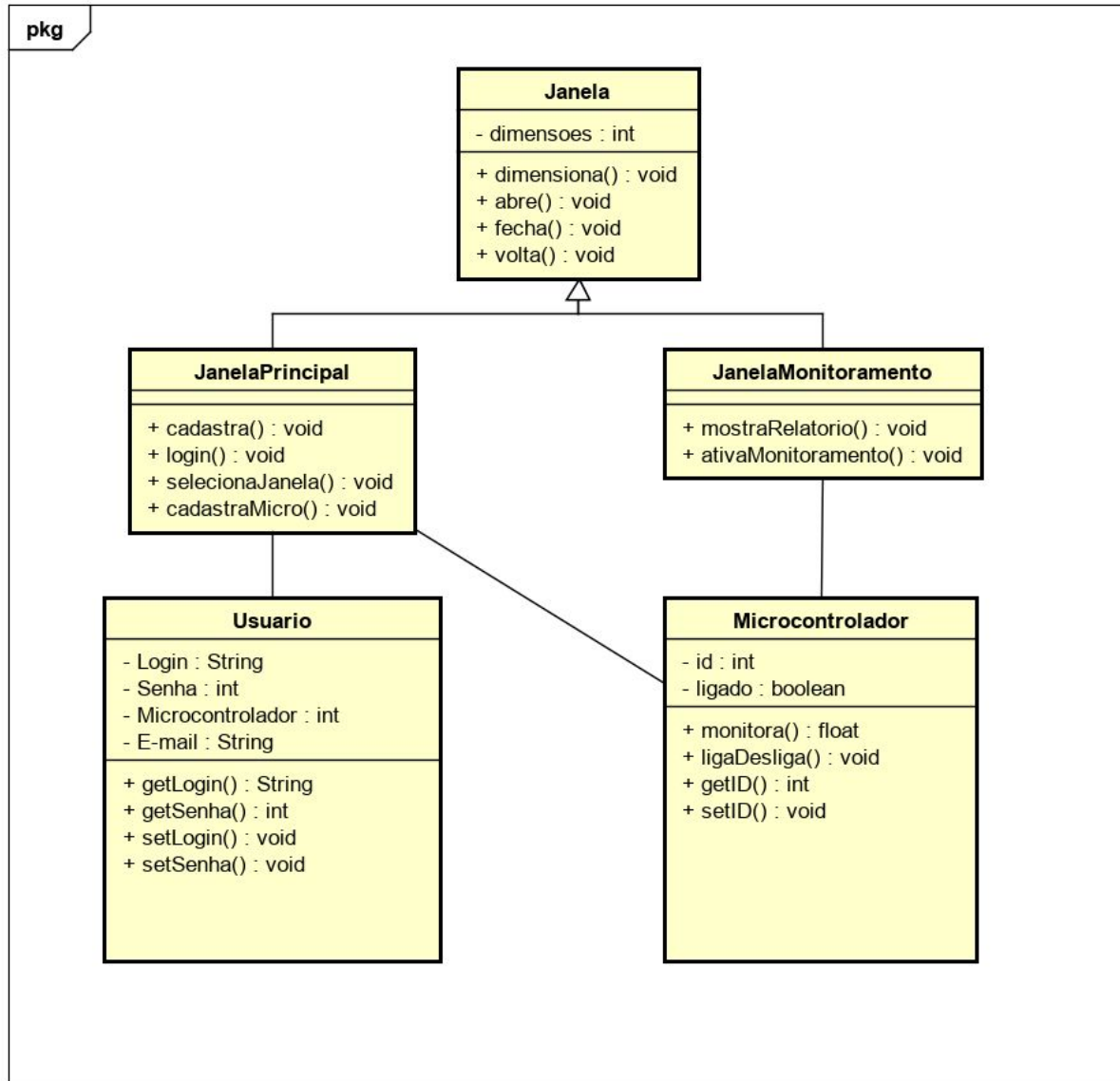
## 6. Diagramas de Casos de Uso

### UC01 - Casos de Uso



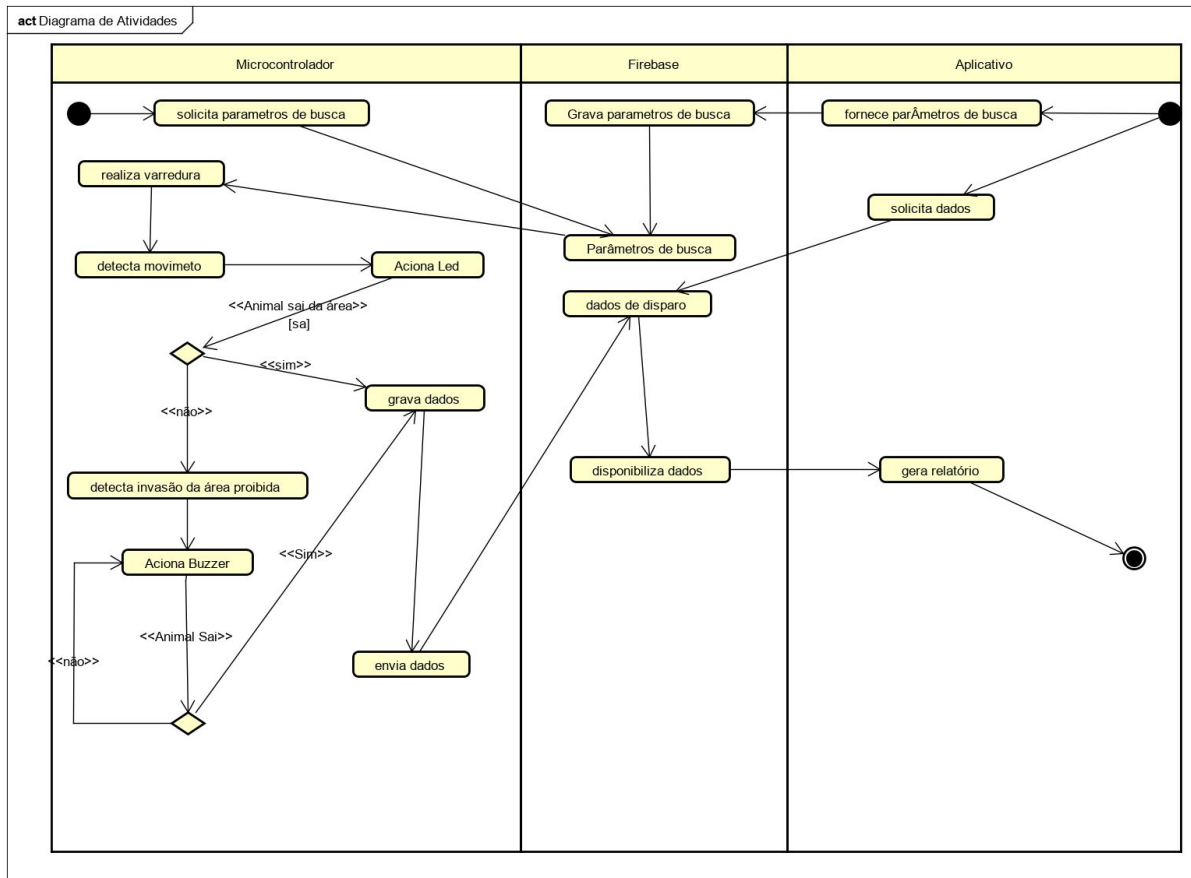
## 7. Diagramas de Classe

### 7.1 Diagrama de Classe



## 8. Diagrama Comportamental

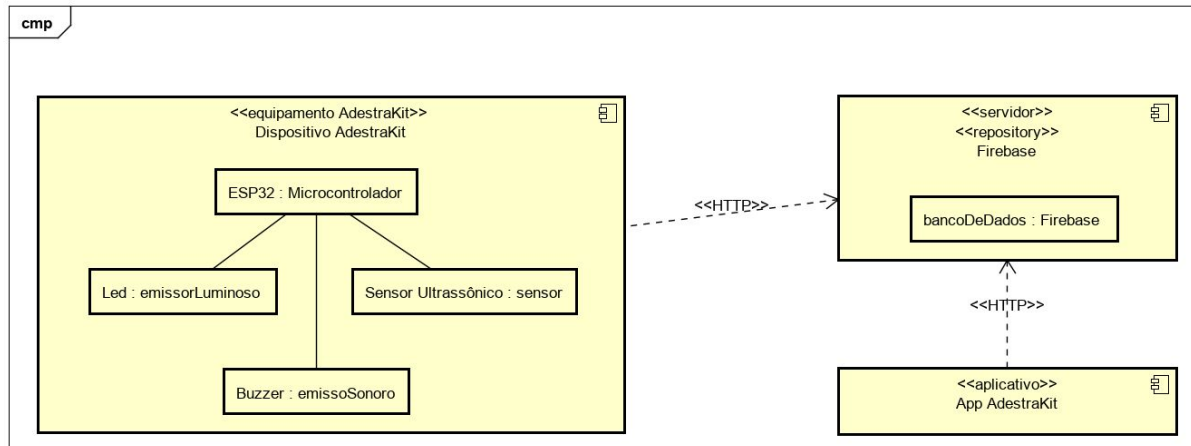
### 8.1 Diagrama de atividades



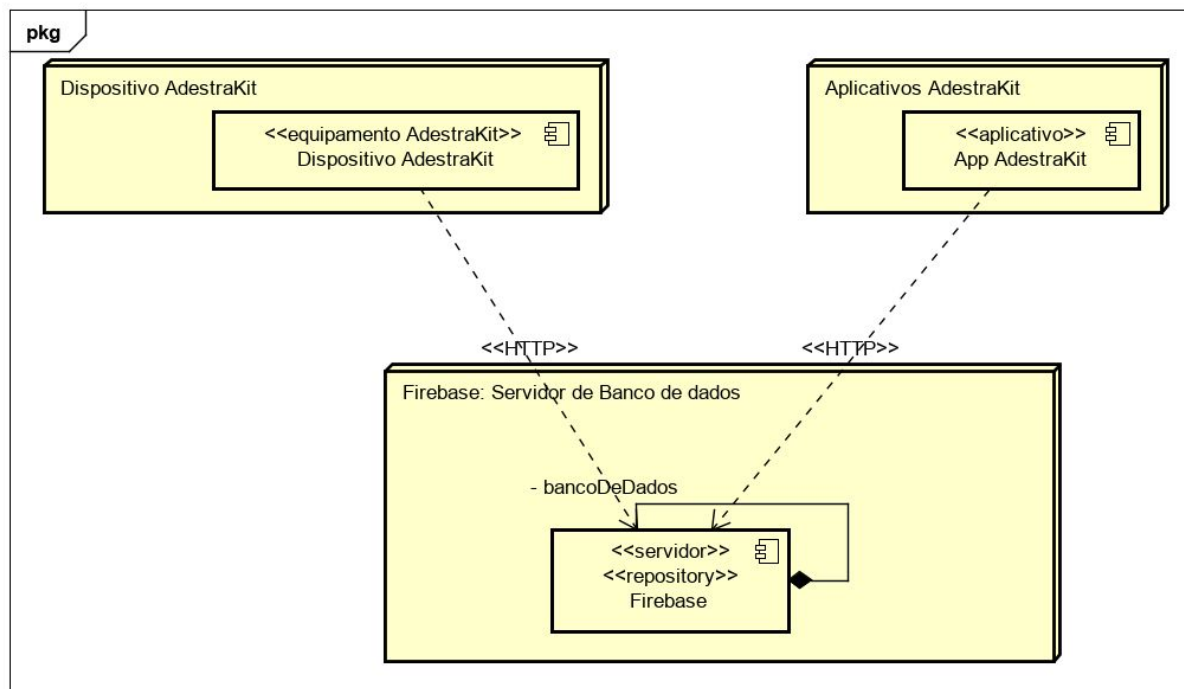


## 9. Diagramas de Implantação

### 9.1 Diagrama de componentes



### 9.2 Diagrama de implementação



## 10. Acompanhamento do cronograma

Nesta semana foi realizado a escrita da documentação e especificação do projeto, além de aquisição de itens necessários aos testes e ao andamento das próximas etapas do projeto, como descrito mais detalhado no quadro abaixo:

Tarefa	Data	Status	Resultado
Elaboração Plano do Projeto	02/10/2019	OK	Drive/PlanodeProjeto
Criação Github e Blog	03/10/2019	OK	Primeira Versão <a href="https://github.com/raiffsf/adestra">https://github.com/raiffsf/adestra</a>
Entrega Moodle	04/10/2019	OK	Moodle
Apresentação Plano	04/10/2019	OK	Em aula
Compra Material Projeto - Esp32, monitor distância ultrasonico, led, fios e protoboard, leitor de cartão RFID e tag RFID	04/10/2019	OK	Material a disposição do grupo
Divisão Próximas Tarefas -	04/10/2019	OK	Em aula
Criação Drive do Projeto e Disponibilização de Modelos	08/10/2019	OK	Drive Compartilhado
Problema com ESP32	08/10/2019	OK	Comprado outro
Aquisição Buzzer e Led	08/10/2019	OK	Material a disposição do grupo
Aquisição ESP32	08/10/2019	OK	Material a disposição do grupo
Criação de e-mail para utilizar no Github:	08/10/2019	OK	Gmail: <a href="mailto:oficinaiotadestra@gmail.com">oficinaiotadestra@gmail.com</a>
Publicação do site/blog no Github	09/10/2019	OK	Novo formato do blog (Primeira Versão) - <a href="https://oficinaiot.github.io/">https://oficinaiot.github.io/</a>
Seção Requisitos no Blog	09/10/2019	20%	Seção para Preenchimento <a href="https://oficinaiot.github.io/">https://oficinaiot.github.io/</a>
Atualização do link do blog	10/10/2019	OK	Enviado novo link pelo Moodle em Comentário na Entrega 1
Planilha Histórico e Próximos Passos	10/10/2019	Andamento	Drive

Requisitos de Sistema na forma de casos de uso	11/10/2019	OK	Entrega 2 - Moodle
Diagrama de casos de uso	11/10/2019	OK	Entrega 2 - Moodle
Diagrama de classes	11/10/2019	OK	Entrega 2 - Moodle
Diagrama comportamental	11/10/2019	OK	Entrega 2 - Moodle

## 11. Dificuldades

A compressão de tempo, no caso uma semana para a documentação e especificação do projeto, tendo em vista que as atividades concorrentes dos participantes do grupo, ficando complicado de encontrar um horário em que todos pudessem participar do debate e definições do projeto. Por tal razão e para que o conhecimento e a opinião de todos sejam aproveitadas no projeto, foi gasto bastante tempo da gestão do projeto por meio de whatsapp, afinal o documento é extenso e necessita da integração e participação de toda a equipe.

## 12. Atividades futuras

Na próxima semana inicia a etapa de desenvolvimento do programa para o microcontrolador ESP32, realizando testes dos sensores e aplicação do buzzer com a frequência desejada, para saber se os resultados no afastamento do cão realmente funciona com eficiência e segurança.

Iniciando ainda a preparação do aplicativo, para que este realize o controle dos microcontroladores, será uma etapa de testes e definição da estrutura base para todo o sistema, com testes no sistema eletrônico, no hardware e do software do sistema móvel.

Revisão e ajustes das especificações de projeto, bem como o início da montagem do Microcontrolador juntamente com o sensores demais dispositivos.