**ENGENHARIA DE SOFTWARE**

6º SEMESTRE - NOTURNO

EDUARDO MACEDO GUEDES

1968709-2

FELIPE CARNEIRO MAGRINELLI

19120039-2

IAGO DA COSTA OLIVEIRA

1982887-2

MAGNO GIOVANNI GABRIEL SORIA TONIOLO

20130836-2

SIDNEY EDSON MELLA NETO

20071624-2

**SMART HERD – Aplicativo**

ATIVIDADE DE ESTUDO PROGRAMADA DO 3º BIMESTRE

MARINGÁ

2021

**SMART HERD**

**Repositório do projeto:** <<https://github.com/eduardooguedes/ESOFT6S-AEP>>

1. **INTRODUÇÃO**

Quando se fala em agricultura e pecuária, é inevitável que o Brasil não entre na conversa. Sendo um dos maiores países do mundo, dentre as atividades econômicas brasileiras de grande relevância, a criação de animais e cultivo de plantas ocupam destaque no cenário mundial. Por mais que o foco das exportações esteja relacionado à criação de gados, destaca-se também a criação de suínos, aves, equinos, ovinos e bufalinos.

Apesar de ser uma atividade antiga e consolidada, a pecuária apresenta desafios para os criadores de animais, principalmente suínos, onde a taxa de mortalidade nos rebanhos pode chegar a 18%. Um desses dados é chamado de mortalidade de porcas (MP), o qual impacta fortemente nos sistemas de produção desses animais e tem grande importância para definir o sucesso financeiro do rebanho. Dos 18% de mortalidade na criação de suínos, cerca de metade (9%) está relacionado à morte de porcas e, dessas mortes, 40% são causadas por eutanásia pois o animal não se encontrava em condições de recuperação ou bem-estar. Dessa maneira, percebe-se uma necessidade enorme de melhoria no controle dos ambientes de criação e vigilância animal.

Pensando nisso, o projeto Smart Herd (Rebanho Inteligente) chega com uma proposta de prevenção e bem-estar de suínos, levando em consideração as principais doenças que assolam esse meio. Dentre essas doenças, encontra-se a peste suína, coccidiose, rinite atrófica, pneumonia, desinteria, entre outras. O interessante é que todas elas podem ser prevenidas ou curadas caso tratadas em seus estágios iniciais. Além disso, os sintomas são visíveis, manifestando-se através de diarreias, feridas na pele do animal, perda de peso, espirros e má alimentação.

Diante desse cenário, por que a taxa de mortalidade continua alta na criação de suínos? A resposta para essa pergunta está ligada ao meio em que esses animais se desenvolvem. Pelo fato de a maioria das doenças citadas serem contagiosas, um único porco infectado pode transmitir para grande parte da matriz em que está envolvido, causando grandes estragos no rebanho. Além disso, ambientes mal higienizados e falta de vigilância por parte do criador contribuem para a propagação das mesmas. Para suprir essa necessidade, o aplicativo Smart Herd traz uma solução de monitoramento dos barracões de criação, desde a temperatura do animal até o comportamento dos mesmos. O intuito é, através de leitores de RFID (etiquetas lidas por radiofrequência), controle de câmeras térmicas no ambiente e microfones espalhados pelo criadouro, o responsável pelo barracão garanta o bem-estar dos animais e se atente a possíveis doenças em seus estágios iniciais.

**2. OBJETIVO**

Por meio de um aplicativo, permitir que o responsável por barracões de porcos monitore os ambientes de criação individualmente e seja notificado de possíveis animais enfermos, facilitando o tratamento precoce de doenças e proporcionando um ambiente de bem-estar animal. Além disso, diminuir a taxa de mortalidade nas matrizes de criação de porcos e aumentar a lucratividade dessa atividade.

**3. DESENVOLVIMENTO**

**3.1. DESCRIÇÃO E FUNCIONALIDADES DO SOFTWARE**

O aplicativo, denominado Smart Herd, permitirá que o usuário, desde que possua a devida infraestrutura em seus barracões e lotes de criação de suínos, tenha acesso às câmeras térmicas instaladas em pontos estratégicos do local, permitindo que, através de leitores de RFID espalhados pelo barracão e etiquetas eletrônicas RFID fixadas nas orelhas dos animais, identifique os porcos por código e visualize a temperatura dos mesmos através de integrações dos leitores com as câmeras térmicas. Caso animais com temperatura abaixo ou acima do normal sejam identificados, o aplicativo emitirá um alerta para o usuário identificando o porco pela sua etiqueta RFID, lote em que se encontra, data e horário.

Outro recurso oferecido pela aplicação, juntamente com conceitos e algoritmo de inteligência artificial, será a identificação, por meio dos microfones alocados no ambiente, de frequências sonoras incomuns dos animais, sinalizando possíveis tosses, espirros, comportamentos de raiva ou outros sintomas de doenças. O usuário será notificado pelo aplicativo através de um alerta, onde poderá reproduzir a gravação do áudio e vídeo captado pelos microfones e câmeras e identificado pela IA, com dados de data e hora do ocorrido.

**3.2. ESPECIFICAÇÃO DE REQUISITOS E CASOS DE USO DO SISTEMA**

**3.2.1. Requisitos Não-Funcionais**

RNF1 - A aplicação será feita em React Native e deverá funcionar em Android.

RNF2 - A base de dados será feita em SQL

RNF3 - A autenticação dos usuários será feita utilizando a autenticação do Firebase

RNF4- A aplicação deve ter acesso à internet para ser utilizada.

RNF5 - A aplicação deverá ter integração com as câmeras instaladas nos barracões.

RNF6 - A aplicação deverá ter integração com os sensores RFID instalados nos barracões.

RNF7 - A aplicação deverá ter integração com os microfones instalados nos barracões.

RNF8 - As câmeras devem ter integração e compartilhamento de informações com os sensores RFID.

RNF9- A câmera deve ser térmica.

RNF10 - Os sensores RFID deverão ser instalados em pontos estratégicos do lote.

RNF11 - A aplicação não deve demorar mais que 5 segundos para carregar as informações.

RNF12 - A aplicação deve estar disponível 24 horas por dia.

RNF13 - O sistema será implementado C#

RNF14 - A aplicação terá uma API desenvolvida em PHP.

RNF15 - A aplicação terá um IA que irá monitorar as frequências sonoras recebidas através dos microfones instalados no barracão.

**3.2.2. Requisitos Funcionais**

RF1 - O sistema deverá gerar alertas.

RF2 - O sistema deverá notificar o usuário sobre novos alertas gerados.

RF3 - O sistema deverá disponibilizar um relatório do lote.

RF4 - O sistema deverá permitir acesso às câmeras térmicas em tempo real.

RF5 - O sistema deverá permitir a reprodução dos áudios do microfone de alertas que ocorram no lote.

RF6 - O sistema deverá permitir a reprodução de vídeo de alertas que ocorram no lote.

RF7 - O usuário poderá visualizar os detalhes dos alertas.

RF8 - O usuário, ao clicar no alerta, será redirecionado aos detalhes do alerta.

RF9 - O usuário deverá selecionar um barracão.

RF10 - O usuário deverá selecionar um lote.

**3.2.3. Casos de Uso**

**3.2.3.1. UC1**

|  |
| --- |
| **Especificação de Caso de Uso:**  **Gerar alerta de saúde** |
| Descrição: Gerar alertas ao usuário caso seja identificado alguma anormalidade de saúde no lote. |
| Pré-Condições:   1. A anormalidade de saúde no lote deve ter sido identificada pelo sistema. |
| Atores:   1. Sistema. 2. Usuário. |
| Fluxo Básico: |
| P1 – O sistema detecta uma anormalidade de saúde (espirro, tosse, temperatura do porco).  P2 – O sistema gera um alerta referente a anomalia identificada, contendo as informações do lote, com a data e a hora.  P3 – O sistema irá arquivar o áudio do microfone e o vídeo da câmera do momento do ocorrido. [FA1]  P4 – O sistema atualiza a listagem de notificações dos lotes.  P5 – O sistema gera uma notificação ao usuário. |
| Fluxos Alternativos: |
| **FA1 – Caso o alerta seja de anormalidade na temperatura do porco.**  FA1.1 – O sistema arquiva apenas o vídeo da câmera do momento do ocorrido. |
| Fluxos de Exceção: |
|  |
| Pós Condições:   1. O usuário deverá receber uma notificação em seu dispositivo móvel. |
| Requisitos Não Funcionais:  **RNF1** – O sistema deve garantir que as informações inseridas pelo usuário, sejam armazenadas de forma segura no banco de dados.  **RNF2** – O sistema deve responder em no máximo 5 segundos.  **RNF3** – A aplicação deve ser acessada pelo dispositivo móvel. |
| Tabelas: |

**3.2.3.2. UC2**

|  |
| --- |
| **Especificação de Caso de Uso:** *UC4*  Gerar alerta de comportamento anormal |
| Descrição: Gerar alertas ao usuário caso seja identificado alguma anormalidade comportamental no lote. |
| Pré-Condições:   1. A anormalidade comportamental no lote deve ter sido identificada pelo sistema. |
| Atores:   1. Sistema. 2. Usuário. |
| Fluxo Básico: |
| P1 – O sistema detecta uma anormalidade comportamental (baixo consumo, briga, canibalismo).  P2 – O sistema gera um alerta referente a anomalia identificada, contendo as informações do lote, com a data e a hora.  P3 – O sistema irá arquivar o áudio do microfone e o vídeo da câmera do momento do ocorrido.  P4 – O sistema atualiza a listagem de notificações dos lotes.  P5 – O sistema gera uma notificação ao usuário. |
| Fluxos Alternativos: |
|  |
| Fluxos de Exceção: |
|  |
| Pós Condições:   1. O usuário deverá receber uma notificação em seu dispositivo móvel. |
| Requisitos Não Funcionais:  **RNF1** – O sistema deve garantir que as informações inseridas pelo usuário, sejam armazenadas de forma segura no banco de dados.  **RNF2** – O sistema deve responder em no máximo 5 segundos.  **RNF3** – A aplicação deve ser acessada pelo dispositivo móvel. |
| Tabelas: |

**3.2.3.3. UC3**

|  |
| --- |
| **Especificação de Caso de Uso:** *UC4*  Gerar alerta de temperatura fora da faixa |
| Descrição: Gerar alertas ao usuário caso seja identificado alguma anormalidade na temperatura do lote. |
| Pré-Condições:   1. A anormalidade de temperatura no lote deve ter sido identificada pelo sistema. |
| Atores:   1. Sistema. 2. Usuário. |
| Fluxo Básico: |
| P1 – O sistema detecta uma anormalidade de temperatura no lote (temperatura acima ou abaixo da faixa).  P2 – O sistema gera um alerta referente a anomalia identificada, contendo as informações do lote, com a data e a hora.  P3 – O sistema irá arquivar o áudio do microfone e o vídeo da câmera do momento do ocorrido.  P4 – O sistema atualiza a listagem de notificações dos lotes.  P5 – O sistema gera uma notificação ao usuário. |
| Fluxos Alternativos: |
| FA1 – **Caso o alerta seja de anormalidade na temperatura do porco.**  FA1.1 – O sistema arquiva apenas o vídeo da câmera do momento do ocorrido. |
| Fluxos de Exceção: |
|  |
| Pós Condições:   1. O usuário deverá receber uma notificação em seu dispositivo móvel. |
| Requisitos Não Funcionais:  **RNF1** – O sistema deve garantir que as informações inseridas pelo usuário, sejam armazenadas de forma segura no banco de dados.  **RNF2** – O sistema deve responder em no máximo 5 segundos.  **RNF3** – A aplicação deve ser acessada pelo dispositivo móvel. |
| Tabelas: |

**3.2.3.4. UC4**

|  |
| --- |
| **Especificação de Caso de Uso:** *UC4*  Gerar relatório do lote |
| Descrição: Gerar relatório do lote com base no histórico do lote |
| Pré-Condições:   1. O usuário deve ter selecionado um lote. |
| Atores:   1. Sistema. 2. Usuário. |
| Fluxo Básico: |
| P1 – O usuário clica no botão “Gerar relatório”  P2 – O sistema utiliza a API para trazer o histórico de informações referente ao lote. [FE1]  P3 – O sistema gera o relatório utilizando as informações trazidas na API.  P4 – O sistema gera um documento PDF que será salvo no dispositivo do usuário. [FE2] |
| Fluxos Alternativos: |
|  |
| Fluxos de Exceção: |
| **FE1 – O sistema não consegue trazer as informações do lote.**  FE1.1 – O sistema apresenta uma mensagem de erro ao usuário e cancela a geração do relatório.  **FE2 – O dispositivo móvel do usuário não possui memória suficiente para salvar o arquivo.**  FE2.1 – O sistema apresenta uma mensagem de erro ao usuário e cancela a geração do relatório. |
| Pós Condições: |
| Requisitos Não Funcionais:  **RNF1** – O sistema deve garantir que as informações inseridas pelo usuário, sejam armazenadas de forma segura no banco de dados.  **RNF2** – O sistema deve responder em no máximo 5 segundos.  **RNF3** – A aplicação deve ser acessada pelo dispositivo móvel. |
| Tabelas: |

**3.2.3.5. UC5**

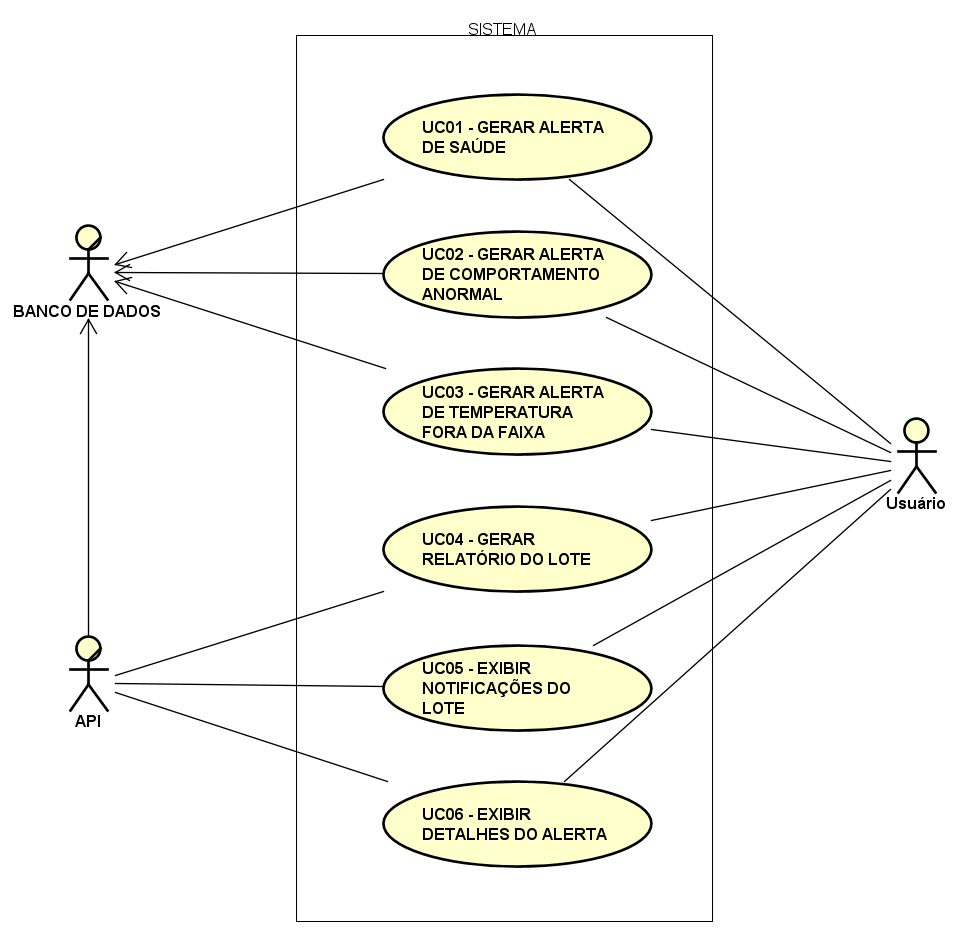
|  |
| --- |
| **Especificação de Caso de Uso:** *UC4*  Exibir notificações do lote |
| Descrição: Exibir a lista de todos os alertas do lote selecionado |
| Pré-Condições: |
| Atores:   1. Sistema. 2. Usuário. |
| Fluxo Básico: |
| P1 – O usuário, na tela inicial, seleciona o lote desejado. [FA1]  P2 - O sistema identifica o lote selecionado pelo usuário.  P3 – O sistema busca, através da API, os alertas vinculados ao lote selecionado. [FA2] [FE1]  P4 – O sistema dispõe todos os alertas para o usuário. |
| Fluxos Alternativos: |
| **FA1 – O sistema não identifica nenhum lote cadastrado para aquele barracão.**  FA1.1 – O sistema exibe a mensagem “Não existe lotes cadastrados para este barracão”.  **FA2 – O sistema não identifica nenhum alerta gerado para o lote selecionado.**  FA2.1 – O sistema exibe a mensagem “Não existe alertas para este lote”. |
| Fluxos de Exceção: |
| **FE1 – O sistema não consegue trazer as informações do lote.**  FE1.1 – O sistema apresenta uma mensagem de erro ao usuário e cancela a geração do relatório. |
| Pós Condições: |
| Requisitos Não Funcionais:  **RNF1** – O sistema deve garantir que as informações inseridas pelo usuário, sejam armazenadas de forma segura no banco de dados.  **RNF2** – O sistema deve responder em no máximo 5 segundos.  **RNF3** – A aplicação deve ser acessada pelo dispositivo móvel. |
| Tabelas: |

**3.2.3.6. UC6**

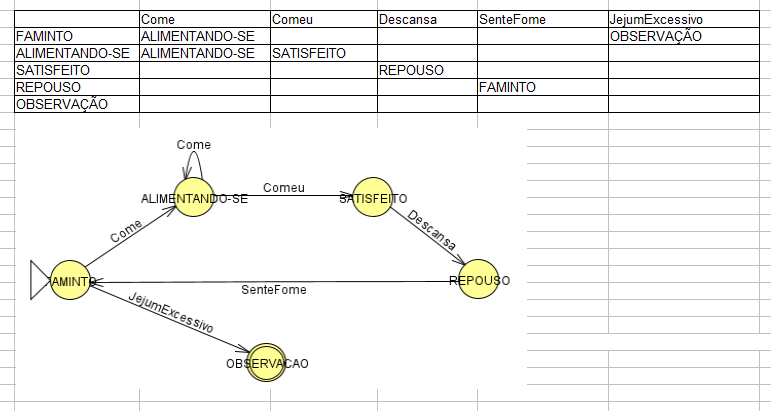
|  |
| --- |
| **Especificação de Caso de Uso:** *UC4*  Exibir detalhes do alerta |
| Descrição: Exibir os detalhes do alerta selecionado |
| Pré-Condições:   1. O sistema deve identificar pelo menos um alerta. |
| Atores:   1. Sistema. 2. Usuário. |
| Fluxo Básico: |
| P1 – O usuário seleciona o alerta desejado.  P2 – O sistema busca, através da API, os detalhes do alerta selecionado. [FA1] [FA2] [FA3] [FE1]  P3 – O sistema exibe as informações do alerta ao usuário. |
| Fluxos Alternativos: |
| **FA1 – O sistema identifica um alerta de saúde (espirro, tosse).**  FA1.1 – O sistema busca as informações referente ao alerta de saúde (data e hora, identificação do animal, barracão, lote e o áudio do ocorrido). [P3]  **FA2 – O sistema identifica um alerta de comportamento anormal (baixo consumo, canibalismo, briga).**  FA2.1 – O sistema busca as informações referente ao alerta de comportamento anormal (data e hora, identificação do animal, barracão, lote e o vídeo do ocorrido). [P3]  **FA3 – O sistema identifica um alerta de temperatura fora da faixa.**  FA3.1 – O sistema busca as informações referente ao alerta de temperatura fora da faixa (data e hora, identificação do animal, barracão, lote, e as temperaturas). [P3]. |
| Fluxos de Exceção: |
| **FE1 – O sistema não consegue trazer as informações do lote.**  FE1.1 – O sistema apresenta uma mensagem de erro ao usuário e cancela a geração do relatório. |
| Pós Condições: |
| Requisitos Não Funcionais:  **RNF1** – O sistema deve garantir que as informações inseridas pelo usuário, sejam armazenadas de forma segura no banco de dados.  **RNF2** – O sistema deve responder em no máximo 5 segundos.  **RNF3** – A aplicação deve ser acessada pelo dispositivo móvel. |
| Tabelas: |

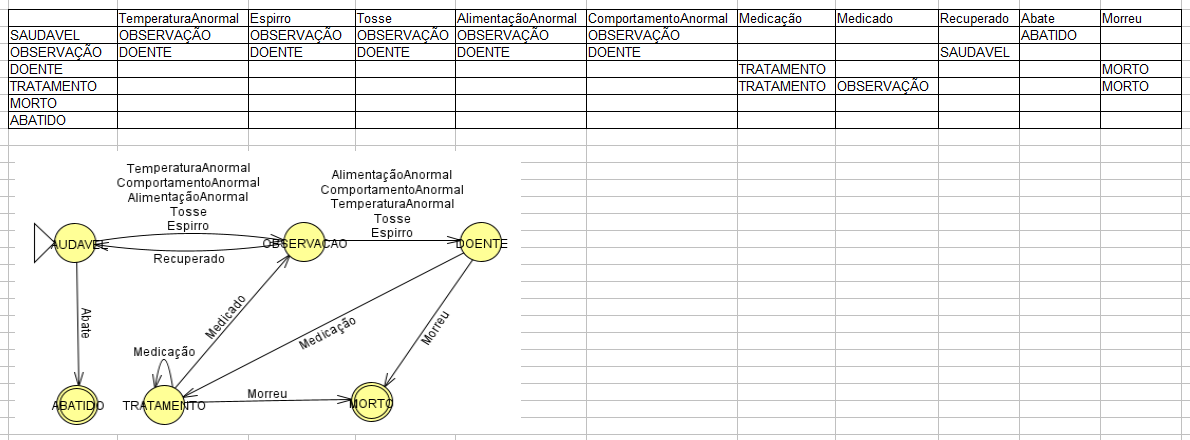
**3.3. DIAGRAMAS UML**

**3.3.1. Diagrama de Caso de Uso**

****

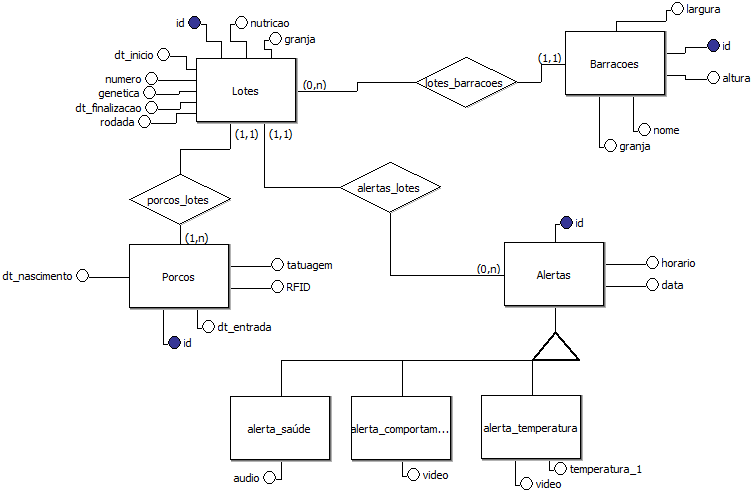
**3.3.2. Diagramas de Estado**

**3.3.2.1. Estado de alimentação**

**3.3.2.2. Estado de saúde**

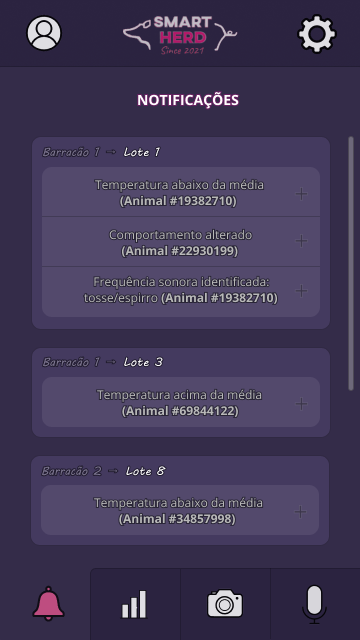
**3.4. BANCO DE DADOS**

**3.4.1. Diagrama Entidade-Relacionamento**

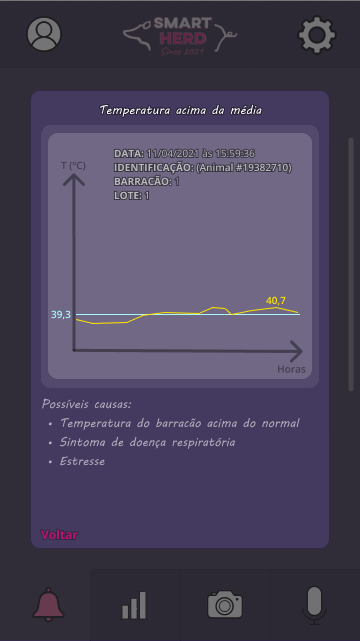
****

**3.5. PRÉ-VISUALIZAÇÃO DAS TELAS (MOCKUP)**

**3.5.1. Notificações/Alertas**

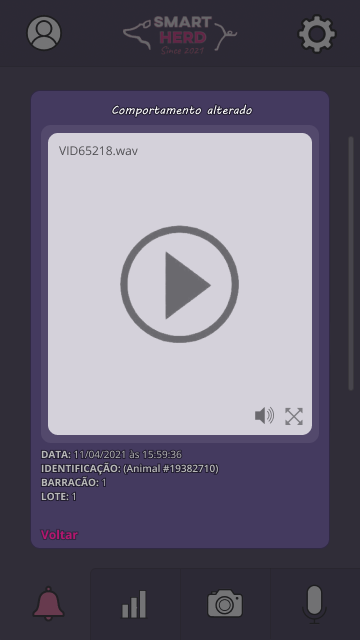
****

Nessa tela serão mostradas as notificações por barracão e lote. Assim que o usuário clicar em alguma delas, será redirecionado para a respectiva tela de notificação relacionada à ocorrência.

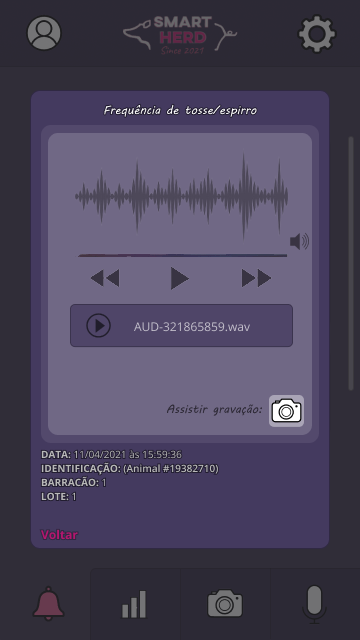
**3.5.2. Alerta de temperatura**

O usuário será redirecionado para essa tela ao clicar em uma notificação de temperatura acima ou abaixo da média. Aqui serão mostrados detalhes da notificação, assim como ID de identificação do animal, temperatura aferida, possíveis sintomas e média de temperatura do lote.

**3.5.3. Alerta de comportamento**



O usuário será redirecionado para essa tela ao clicar em uma notificação de comportamento alterado identificado pelas câmeras e microfone. Aqui o usuário poderá ver o vídeo do animal identificado na hora do comportamento agressivo, assim como ouvir o áudio captado pelos microfones.

**3.5.4. Notificação de frequência sonora suspeita (tosse ou espirro)**

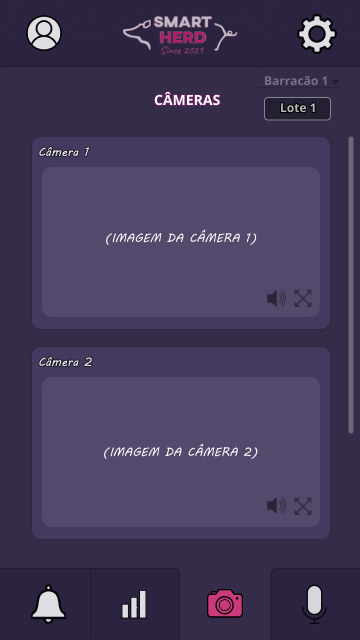
O usuário será redirecionado para essa tela ao clicar em uma notificação de frequência sonora de tosse ou espirro captada pelos microfones. Aqui o usuário poderá ouvir o áudio captado, assim como visualizar as gravações da câmera no momento de captura para averiguar se realmente o barulho foi anormal.

**3.5.5. Relatório**



Nessa tela serão mostrados dados do lote selecionado, assim como oferecerá a opção de gerar um arquivo em formato PDF com o relatório do mesmo. Dentre esses dados haverá o número de animais ativos, animais que estão em cuidados médicos, peso médio do lote, entre outros.

**3.5.6. Câmera**



Nessa tela o usuário poderá ver em tempo real as imagens transmitidas pela câmera do ambiente monitorado, assim como averiguar as temperaturas dos animais através da integração da câmera térmica com os leitores de RFID.

**3.5.7. Gravações**



Nessa tela serão mostradas as gravações de todos os áudios suspeitos ou de comportamento alterado captados pelos microfones do barracão. Ao clicar em uma das gravações, o usuário será redirecionado para uma tela similar à de notificação de áudios suspeitos, onde poderá ver gravação das câmeras e ouvir o áudio.

**3.6. CONCEITOS DE INTELIGÊNCIA ARTIFICIAL UTILIZADOS**

Nossa motivação para utilização de inteligência artificial em nosso projeto se deve à proposta de haver incluso diversas análises com base nos dados que serão recolhidos pelo ambiente, alguns mais complexos que outros.

Através da utilização de microfones, a IA irá fazer um reconhecimento de quando houver algum barulho diferente como um espirro ou uma tosse do animal, assim, proporcionando uma relação de sintomas para a execução de um tratamento precoce.

Essa análise será feita através da comparação de frequências que serão armazenadas e retroalimentadas para o reconhecimento de futuras frequências.

Com isso o dono dos animais poderá ter uma relação de quais animais apresentam indícios de estarem com algum tipo de comorbidade. A IA com base nesses dados também mostrará probabilidades de um dos animais ficar doente. Cada animal será identificado por um RFID, o que permitirá que sejam feitas análises individuais.

Baseado nos conceitos citados acima e na hierarquia de classes para os componentes do projeto, criou-se uma ontologia que pode ser encontrada no repositório abaixo, na pasta “3º Bim 2021 - ESOFT6S/InteligenciaArtificial”:

**Repositório:** <<https://github.com/eduardooguedes/ESOFT6S-AEP>>

**3.7. GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE**

As técnicas aplicáveis de gerenciamento de configuração do projeto, assim como a política estabelecida para solicitações de mudança e responsabilidades de cada membro da equipe encontram-se no repositório abaixo, na pasta “[3º Bim 2021 - ESOFT6S](https://github.com/eduardooguedes/ESOFT6S-AEP/tree/master/3%C2%BA%20Bim%202021%20-%20ESOFT6S)/GerenciamentoDeConfiguracao”:

**Repositório:** <<https://github.com/eduardooguedes/ESOFT6S-AEP>>

**3.8. MÉTRICAS DE SOFTWARE**

Para nos orientarmos em relação ao desenvolvimento do software proposto, utilizaremos a Análise de Pontos de Função. Essa é uma métrica internacional que tem como objetivo principal descobrir qual o tamanho de um software a partir dos seus requisitos funcionais. Isso sem levar em conta a linguagem de programação utilizada na construção do sistema, que só será analisada após a contagem ser finalizada.

Métrica de software implica em tudo que pode ser medido e quantificado em um sistema, documentação ou processo. Em nossa solução Smart Herd, utilizaremos métricas de tamanho do software, buscando obter uma análise próxima da realidade, de quantos pontos precisaremos.

As métricas podem ser classificadas em métricas de controle ou de previsão. Métricas de controle estão associadas a processos de software, como: esforço médio e tempo necessário para implementar uma funcionalidade ou corrigir um bug. Métricas de previsão se associam ao sistema em si, como linhas de código e tamanho das classes.

Já para a análise de pontos de função, existem três tipos de contagem: projeto de desenvolvimento, projeto de melhoria e aplicação. Utilizaremos a contagem de projeto de desenvolvimento por se tratar de um software totalmente novo, e para isso, analisaremos todas as funcionalidades do sistema, avaliando suas complexidades e esforços, para então chegarmos à estimativa final da contagem.

O primeiro passo será determinar o escopo da aplicação, isso a partir do ponto de vista do usuário. O ponto de vista do usuário é importante nesse ponto pois é ele que de fato utilizará a aplicação, portanto é o melhor em dizer quais funcionalidades são necessárias.

A partir disso, contaremos as funções do tipo dados, também chamadas de arquivos lógicos. Esses arquivos podem ser mantidos pela aplicação (Arquivos Lógicos Internos - ALI) ou serem lidos de outra (Arquivos de Interface Externa – AIE). Definiremos para cada arquivo lógico encontrado qual sua complexidade e qual sua contribuição.

Então, contaremos as funções do tipo transação, também chamadas de processos elementares. Representam a menor unidade de uma função disponível ao usuário e são classificadas em Entradas Externas, Saídas Externas e Consultas Externas. Assim que contadas, também definiremos as complexidades e contribuição de cada uma.

O quarto passo é determinar a contagem dos pontos de função não ajustados, que representa a relação de um arquivo lógico com um processo elementar, somando os dois na hora de calcular. Seria seguido do fator de reajuste, mas não o calcularemos (o fator de reajuste é opcional desde 2002 por sua contagem poder resultar em uma falha na contagem final). E finalizaremos realizando o cálculo dos pontos de função ajustados.

Para cada tipo de contagem existe uma lógica para realizar a contagem final. Como escolhemos o tipo “Projeto de Desenvolvimento”, utilizaremos sua lógica que é: DFP = (UFP + CFP) x VAF, sendo:

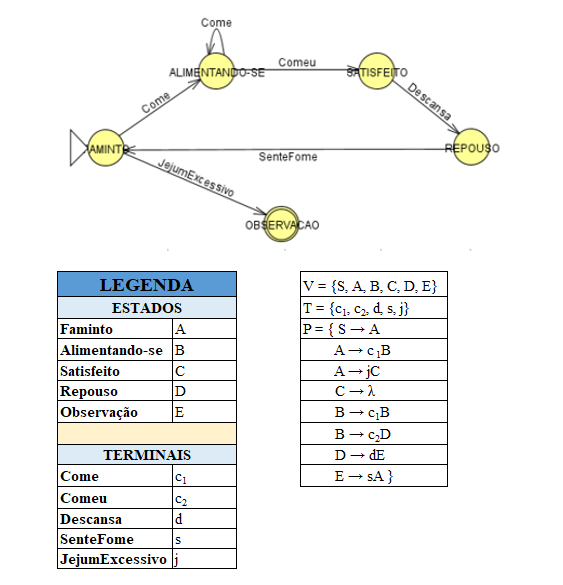
* DFP: número de pontos de função do projeto de desenvolvimento
* UFP: número de pontos de função não ajustado das funções disponíveis aos usuários após instalação
* CFP: número de pontos de função não ajustados das funções de conversão
* VAF: valor do fator de ajuste.

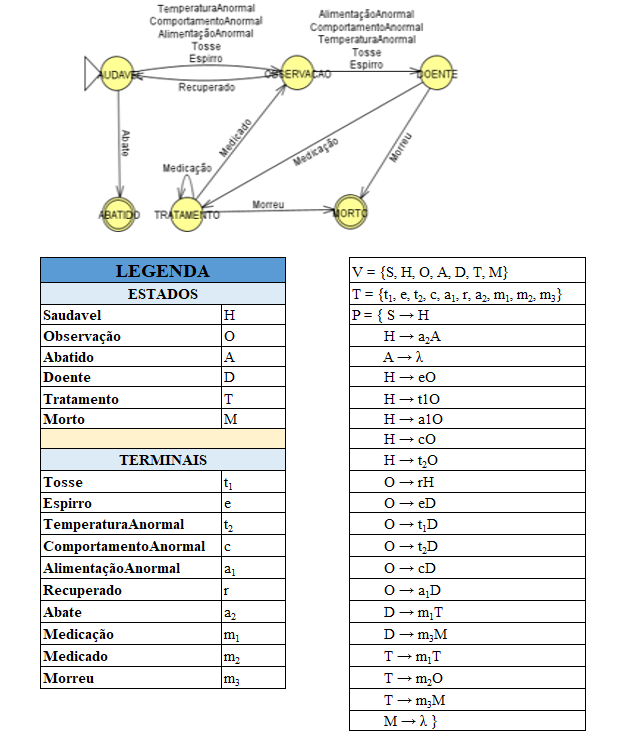
Após realizarmos todo o cálculo e chegarmos à quantidade de pontos de função necessários para o desenvolvimento do novo projeto, definiremos quantas horas valem cada ponto de função para nossa equipe. Isso a partir do conhecimento da equipe em relação a linguagem de programação que será utilizada, conhecimento em banco de dados, entre outras tecnologias que serão analisadas, o que nos ajudará a chegar em uma medida muito próxima da realidade de quanto tempo levaremos para finalizar a aplicação.

**3.9. TEORIA DA COMPUTAÇÃO**

Baseando-se nos dois diagramas de estado (Pág. 13), representados por autômatos finitos determinísticos, para estado de alimentação e estado de saúde dos animais do lote, desenvolveu-se as seguintes gramáticas regulares:

**3.9.1. Estado de alimentação**



**3.9.2. Estado de saúde**

**4. CONCLUSÃO**

Diante do cenário atual da pecuária mundial e com o avanço da tecnologia, o aplicativo Smart Herd propõe de maneira objetiva e funcional melhorias na criação de suínos, sendo de grande utilidade para fazendeiros que se preocupam com a qualidade de seus produtos e buscam aumentar a lucratividade à curto e médio prazo. Além disso, o projeto busca atender um viés ambiental, onde os animais poderão viver com uma melhor qualidade de vida e em ambientes agradáveis para o desenvolvimento saudável dos mesmos. Dessa maneira conclui-se que a solução proposta contribui positivamente para a atividade pecuária não só economicamente, mas também de maneira sustentável, economizando recursos que seriam utilizados para um número maior de animais consequentes do número de mortes que poderão ser evitadas.

**5. REFERÊNCIAS**

GUIMARÃES, T.P., SILVA, M.A.P. e LEÃO, K.M. Índices zootécnicos de uma granja produtora de leitões. PUBVET, Londrina, V. 4, N. 41, Ed. 146, Art. 983, 2010. Disponível em:

<https://www.pubvet.com.br/uploads/3f15922e60b45157169359813f598ad2.pdf/>. Acesso em: 28 ago. 2021.

DECHICHI, C. Criação orgânica de suínos no Brasil e no mundo. UNB (Universidade de Brasília), Brasília, 2013. Disponível em:

<https://bdm.unb.br/bitstream/10483/5967/1/2013\_CassiaDechichi.pdf>. Acesso em 28 ago. 2021

KIRSCHNER, L. Pig diseases you should know. 2016. Disponível em:

<https://www.wattagnet.com/articles/26329-pig-diseases-you-should-know>. Acesso em 04 set. 2021.

CHEMITECH. Doenças comuns aos suínos. CHEMITECH, São Paulo, 2020. Disponível em:

< https://chemitec.com.br/blog/doencas-comuns-aos-suinos/>. Acesso em 04 set. 2021.

ASSESSORIA. Um desafio para suinocultura. O PRESENTE RURAL, 2020. Disponível em:

<https://opresenterural.com.br/um-desafio-para-suinocultura/>. Acesso em 05 set. 2021.

MORÉS, N. Mortes de matrizes em granjas de suínos. EMBRAPA. Disponível em:

<http://www.sossuinos.com.br/Tecnicos/info160.htm>. Acesso em: 05 set. 2021.