

---

Curso de Ciência da Computação  
Universidade Estadual de Mato Grosso do Sul

---

**Trabalho de Conclusão de Curso**

**DESENVOLVIMENTO DE SOFTWARE EM PWA PARA  
AUXÍLIO NA CRIAÇÃO DE BOVINOS**

ARTHUR JOSÉ MEDEIROS

Dr. Evandro Cesar Bracht (Orientador)

Dourados - MS  
2022



# **DESENVOLVIMENTO DE SOFTWARE EM PWA PARA AUXÍLIO NA CRIAÇÃO DE BOVINOS**

**ARTHUR JOSÉ MEDEIROS**

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Arthur José Medeiros e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Dourados, 17 de Outubro de 2022

Prof. Dr. Evandro Cesar Bracht (Orientador)

M438d Medeiros, Arthur José

Desenvolvimento de software em PWA para auxílio na criação de bovinos/ Arthur José Medeiros, MS: UEMS, 2022.

40p.

Monografia (Graduação) – Ciência da Computação – Universidade Estadual de Mato Grosso do Sul, 2022.

Orientador: Prof. Dr. Evandro Cesar Bracht

1. Desenvolvimento de software 2. Desenvolvimento web 3. Processamento de dados 4. Offline first I. Bracht, Evandro Cesar. II. Título

CDD 23.ed. - 006.76

---

Curso de Ciência da Computação  
Universidade Estadual de Mato Grosso do Sul

---

# **Desenvolvimento De Software Em PWA Para Auxílio Na Criação De Bovinos**

Arthur José Medeiros

Novembro de 2022

**Banca Examinadora:**

Prof. Dr. Evandro Cesar Bracht(Orientador)

Área de Computação –UEMS

Prof. Dr. Cleber Valgas Gomes Mira

Área de Computação –UEMS

Prof. Dr. Diogo Fernando Trevisan

Área de Computação –UEMS



## **Agradecimentos**

Agradeço a minha família, pelos valores e lições que me passaram durante toda minha vida.

Agradeço ao meu orientador pela oportunidade e pelo auxílio durante esse trabalho.

Agradeço a todos os meus professores que me passaram seus conhecimentos e sabedoria durante meus anos de estudo, algo que não tem valor.







## **RESUMO**

Nas últimas décadas ocorreram grandes saltos em tecnologias disponíveis em todas as áreas, inclusive para os produtores rurais, os quais se beneficiaram muito com estes avanços. A partir da década de 80, com a chegada dos computadores pessoais, foi possível informatizar muitos processos manuais e um destes processos foi o de criação de bovinos por meio de programas desenvolvidos para esta finalidade. No entanto tais programas estão desatualizados e muitos estão sem suporte. O objetivo deste projeto é criar um aplicativo moderno, utilizando a metodologia de Progressive Web App (voltada para desenvolvimento web), capaz de funcionar tanto em dispositivos móveis quanto desktops para ajudar os produtores no processamento dos dados de seus rebanhos, mantendo essas informações de fácil acesso mesmo que não possuam acesso à internet.

Palavras-chave: Progressive Web App; Bovinos; Offline-first



# SUMÁRIO

<b>RESUMO</b>	<b>9</b>
<b>1. INTRODUÇÃO</b>	<b>13</b>
1.2 Problemática	13
1.3 Objetivo Geral	13
1.3.1 Objetivos Específicos	13
1.4. Justificativa	14
1.5 Aplicativos Existentes	14
<b>2. REFERENCIAL TEÓRICO</b>	<b>17</b>
2.1 PWA	17
2.1.1 Arquitetura de um PWA	18
2.2 Quasar Framework	19
2.3 Google Firebase	19
2.4 Hardware	20
<b>3. BANCO DE DADOS</b>	<b>21</b>
<b>4. INTERFACES</b>	<b>25</b>
<b>5. CONCLUSÃO</b>	<b>37</b>
5.1 Trabalho Futuros	37
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>39</b>



# 1. INTRODUÇÃO

Este estudo consiste na criação de um software de apoio aos produtores pecuaristas, com foco particular nos que utilizam brincos de identificação por radiofrequência (RFID) (Voulodimos et al, 2010) para identificar os seus rebanhos. O software irá auxiliar o produtor no controle dos seus rebanhos, mantendo dados essenciais como identificação, vacinação, e peso.

## 1.2 Problemática

Muitas fazendas ainda usam sistemas arcaicos para controle de seus rebanhos, desde possuírem apenas cópias físicas da documentação de seus animais até tabelas EXCEL salvas localmente e software desatualizado, como o CONGADO, criado pela Alma Informática, que não possui mais suporte técnico desde 2008 e foi criado utilizando bancos de dados em Paradox. Com os avanços tecnológicos dos dias modernos, devemos nos perguntar como podemos melhorar as práticas agropecuárias que foram fundamentais para o ser humano no seu primórdio e continuam a ser até os dias atuais?

## 1.3 Objetivo Geral

Desenvolver software para manejo de bovinos *offline first*<sup>1</sup>, com uma interface de fácil navegação para que possa ser utilizado de forma intuitiva por qualquer usuário, e que contenha os dados necessários para manejo dos rebanhos como: parentesco, peso, vacinação e medicamentos, sexo, idade, raça, identificação individual e o número do SISBOV (Sistema Brasileiro de Identificação Individual de Bovinos e Búfalos) .

### 1.3.1 Objetivos Específicos

1. Conhecer a metodologia *Progressive Web Application* (PWA) para criação de um web app.
2. Analisar as informações de maior importância para a criação de gado.  
Modelagem e Criação do banco de dados.

---

<sup>1</sup> Offline-First é um paradigma de programação em que o software deve funcionar tão bem offline quanto online. Para implementar isso, é preciso armazenar dados no lado do cliente, para que seu aplicativo ainda possa acessá-los quando a Internet sair. (OFFLINE)

3. Desenvolvimento do aplicativo.
4. Verificar a funcionalidade do software em situações reais.

#### **1.4. Justificativa**

A modernização de tecnologias facilitou a vida de todos, inclusive dos trabalhadores rurais. Com essas novas tecnologias é possível realizar coleta de dados e manejo de animais com maior facilidade em campo com apps móveis, porém isso pode trazer novos problemas. Por exemplo, se houver a perda ou danificação do aparelho, isso pode causar atrasos em todo o funcionamento da fazenda até que um novo dispositivo seja adquirido.

Em muitos casos também ocorre de fazendas utilizarem duas máquinas separadas: uma para leitura dos dados e outra para o seu armazenamento, o que pode gerar atrasos ou problemas maiores como a perda total de dados caso ocorra algum defeito nesses.

Por isso, foi proposta a criação deste software em PWA, permitindo que os dados sejam armazenados de forma segura e possam ser acessados de qualquer lugar, por causa do banco de dados na nuvem. O software também poderá ser acessado não apenas de um dispositivo, mas sim de qualquer um que possuir um navegador web compatível, seja ele computador ou dispositivo móvel, podendo eles serem aparelhos mais antigos para acessar o aplicativo pois necessitam apenas de um navegador atualizado.

#### **1.5 Aplicativos Existentes**

Antes do desenvolvimento foi realizada uma pesquisa sobre aplicativos existentes no mercado com o mesmo fim, sendo os mais populares. JetBov (JETBOV), BovExo (BOVEXO) e Fertilí (FERT). Os principais pontos negativos que esses aplicativos possuem em comum é a limitação da quantidade de cabeças que podem ser registradas por usuário além da necessidade de instalação no dispositivo.

Outra similaridade entre os softwares é que possuem um foco na compra e venda dos animais através da própria aplicação, o que muitos usuários podem achar desnecessário. Esses aplicativos possuem um limite de animais que podem ser registrados, em torno de 10000 cabeças.

Com o software proposto, que utiliza o banco de dados em modelo NOSQL é possível ter um número muito maior de registros, pois ele apenas limita os usuários

no tamanho dos documentos não em suas quantidades. Que será explicado mais a fundo no capítulo 4.





## 2. REFERENCIAL TEÓRICO

Para entendermos o funcionamento do software é preciso abordar os conceitos e tecnologias que foram usados em seu desenvolvimento, além de abordar o histórico de seus surgimentos.

### 2.1 PWA

Em 2015, o nome “Progressive Web Apps” foi cunhado pelo desenvolvedor do Chrome Alex Russel e pelo designer Frances Berriman em um artigo (Russel, 2015) que convidava para uma “mudança profunda em nossa compreensão e ferramentas” para “construir melhores experiências em dispositivos e contextos dentro de uma única base de código .”

PWA é uma metodologia utilizada para desenvolver aplicativos que possam ser um meio termo entre páginas web e aplicativos nativos, com a principal função de ser acessível em qualquer aparelho que possua um navegador web padrão como Chrome, Safari ou Edge (Sheppard, 2017).

Os softwares desenvolvidos utilizando a metodologia PWA vão além de páginas web usando *service workers*, manifestos e outros recursos de plataforma da Web em combinação com aprimoramento para oferecer aos usuários uma experiência equivalente a aplicativos nativos (Steiner, 2018), permitindo que usuários instalem o aplicativo em seus dispositivos para uso offline e os desenvolvedores não precisam distribuir por lojas virtuais .

A metodologia de um PWA se baseia em três pilares:

- Capacidade: Os recursos de um PWA estão próximos dos aplicativos nativos, podendo criar aplicativos da Web com notificações por *push*, geolocalização, vídeo, áudio etc.
- Confiável: O PWA é rápido e pode funcionar *offline*.
- Instalável: O PWA pode ser instalado como um aplicativo nativo
  - Para que o aplicativo seja instalável ele precisa se encaixar em três critérios:
  - Origem segura.
  - Registrar um *service worker* com um *fetch handler*.
  - Referenciar um manifesto web.

Alguns site que adotaram a metodologia:

- **FlipKart:** a maior plataforma de comércio eletrônico da Índia, buscando reinventar sua experiência digital, recorre ao PWA em 2016. A empresa viu 3x mais tempo gasto no site; taxa de reengajamento 40% maior; taxa de conversão 70% maior entre aqueles que chegam via “Adicionar à tela inicial”; e uso de dados 3x menor (FLIPKART, 2016) .
- **AliExpress:** aumentou a taxa de conversão para novos usuários em 104% com um novo PWA em 2016 (ALI, 2016).
- **Twitter:** lançou o Twitter Lite para aumentar significativamente o engajamento e reduzir o consumo de dados em 70% em 2017 (TWITTER, 2017).

### 2.1.1 Arquitetura de um PWA

Os principais recursos que diferenciam os PWA são os *service workers* e manifestos. Esses arquivos permitem que o aplicativo não apenas se adapte a qualquer dispositivo compatível, mas também possa ser usado sem acesso à internet.

- **Service Workers**

Os *Service Workers* são um *proxy* virtual entre o navegador e a rede, eles corrigem problemas com os quais os desenvolvedores *front-end* lutam há anos - principalmente como armazenar em *cache* adequadamente os ativos de um site e disponibilizá-los quando o dispositivo do usuário estiver *offline*.

Eles são executados em uma *thread* separado do código JavaScript principal de nossa página e não têm nenhum acesso à estrutura DOM, isso apresenta uma abordagem diferente da programação web tradicional, pois a API (*Application Programming Interface*) não é bloqueante e pode enviar e receber comunicação. Você pode dar a um Service Worker algo para trabalhar e receber o resultado sempre que estiver pronto usando uma abordagem baseada em Promise.

Os PWA podem fazer muito mais do que "apenas" oferecer recursos offline, incluindo manipulação de notificações, execução de cálculos pesados em um thread separado, etc.

- **Manifesto**

Conforme definido na especificação do Manifesto, este fornece informações sobre um aplicativo da Web em um arquivo de texto JSON (JavaScript Object Notation), necessário para que o aplicativo da Web seja baixado e apresentado ao usuário de forma semelhante a um aplicativo nativo. Por exemplo, pode ser instalado na tela inicial de um dispositivo, proporcionando aos usuários um acesso mais rápido e uma experiência mais rica. Os manifestos do PWA incluem seu nome, autor, ícone(s), versão, descrição e lista de todos os recursos necessários, entre outras coisas.

## 2.2 Quasar Framework

Quasar Framework é um framework baseado em Vue.js, open source, usado para construção de aplicativos, com uma única base de código e implantá-lo na Web como SPA, PWA, SSR (De Vylder, 2016), (Quasar).

Ele foi escolhido por possuir funcionalidades que facilitam a implementação de um PWA, como o comando de *build pwa* que garante a criação correta dos arquivos necessários para o funcionamento do aplicativo, contando também com algumas instruções em html que facilita o desenvolvimento do *frontend*.

## 2.3 Google Firebase

O Google Firebase é uma ferramenta de back-end para desenvolver aplicativos escalonáveis e de alto desempenho para Web e dispositivos móveis com baixo custo, tempo e esforço. Foi desenvolvido pela Envolve em 2011 por James Templon e Andrew Lee e, em 2014 foi comprado pelo Google. O firebase possui várias ferramentas para desenvolvedores. As ferramentas do firebase que foram usadas na criação do aplicativo são: Firebase Cloud Firestore, Firebase Auth e Firebase Hosting (FIREBASE).

- **Cloud Firestore**

O Cloud Firestore é um banco de dados flexível e escalável no modelo NoSQL para desenvolvimento em dispositivos móveis e Web. Ele mantém seus dados sincronizados entre aplicativos clientes por meio de listeners em tempo real e oferece suporte *offline* (FIRESTORE). Entretanto a sincronização armazena a atualização mais recente, o que pode gerar problemas caso dois usuários alterem o

mesmo dado *offline*, o banco de dados irá aceitar as atualizações do usuário que entrar *online* por último, levando a possíveis erros.

- **Firestore Auth**

O serviço Firestore Auth fornece recurso de autenticação para aplicativos, possibilitando o login por redes sociais como Gmail, Facebook, Twitter, GitHub, etc. Para esse aplicativo utilizamos apenas a autenticação por email e senha.

## 2.4 Hardware

Entre os principais *hardwares* utilizados no manejo de bovinos devemos citar os brincos de identificação eletrônica, **Figura 1A**, e seus aparelhos de leitura, como o bastão que é utilizado quando os animais estão passando pelos mangueiros, **Figura 1B** e o modelo utilizado em campo e que foi utilizado nos testes deste aplicativo, **Figura 1C**. Estes operam nos padrões internacionais de RFID ISO 11784 e ISO 11785 para tecnologias FDX-B (Full duplex) e HDX (Half duplex) com frequências entre 129-133.2 kHz e 124.2 kHz (TAGS, 2007).



Figura 1- A. Brinco de Identificação eletrônico

B. Bastão de Leitura RFID

C. Aparelho de Leitura RFID Compacto

### 3. BANCO DE DADOS

O Firestore é um banco de dados NoSQL com armazenamento na nuvem que utiliza um sistema de coleções e documentos. Cada coleção contém documentos e esses documentos por sua parte podem conter coleções que contém documentos, ad infinitum. NoSQL é um termo genérico que representa os bancos de dados não relacionais, uma classe definida de banco de dados que fornece um mecanismo para armazenamento e recuperação de dados que são modelados de formas diferentes das relações tabulares usadas nos bancos de dados relacionais (NOSQL).

Na **Figura 2** podemos ver o diagrama do banco de dados e como está estruturado.

- Users:
  - Essa coleção armazena todos os usuários cadastrados no aplicativo e seus UID (User IDs) em documentos.
  - Possui as subcoleções Fazendas e Funcionários
- Funcionário:
  - Essa coleção armazena os funcionários de uma fazenda cadastrados pelo admin através de seu email e o ID da fazenda onde trabalham.
- Fazendas:
  - A coleção Fazendas possui a subcoleção Fazenda.
- Fazenda :
  - O documento armazena os IDs da fazenda e do proprietário.
  - Possui as subcoleções Lote.
- Lote:
  - O documento armazena os IDs da fazenda e do lote.
  - Possui as subcoleções Animal.
- Animal:
  - O documento armazena os IDs do animal, da fazenda, do lote, do pai e mãe, a raça, o sexo, e a data de nascimento.
  - Possui as subcoleções Sanitário e Peso.
- Sanitário:
  - Os documentos armazenam o ID do animal, o medicamento e a quantidade aplicada e o dia da aplicação.

- **Peso:**
  - Os documentos armazenam o ID do animal, o peso, e a data da pesagem.

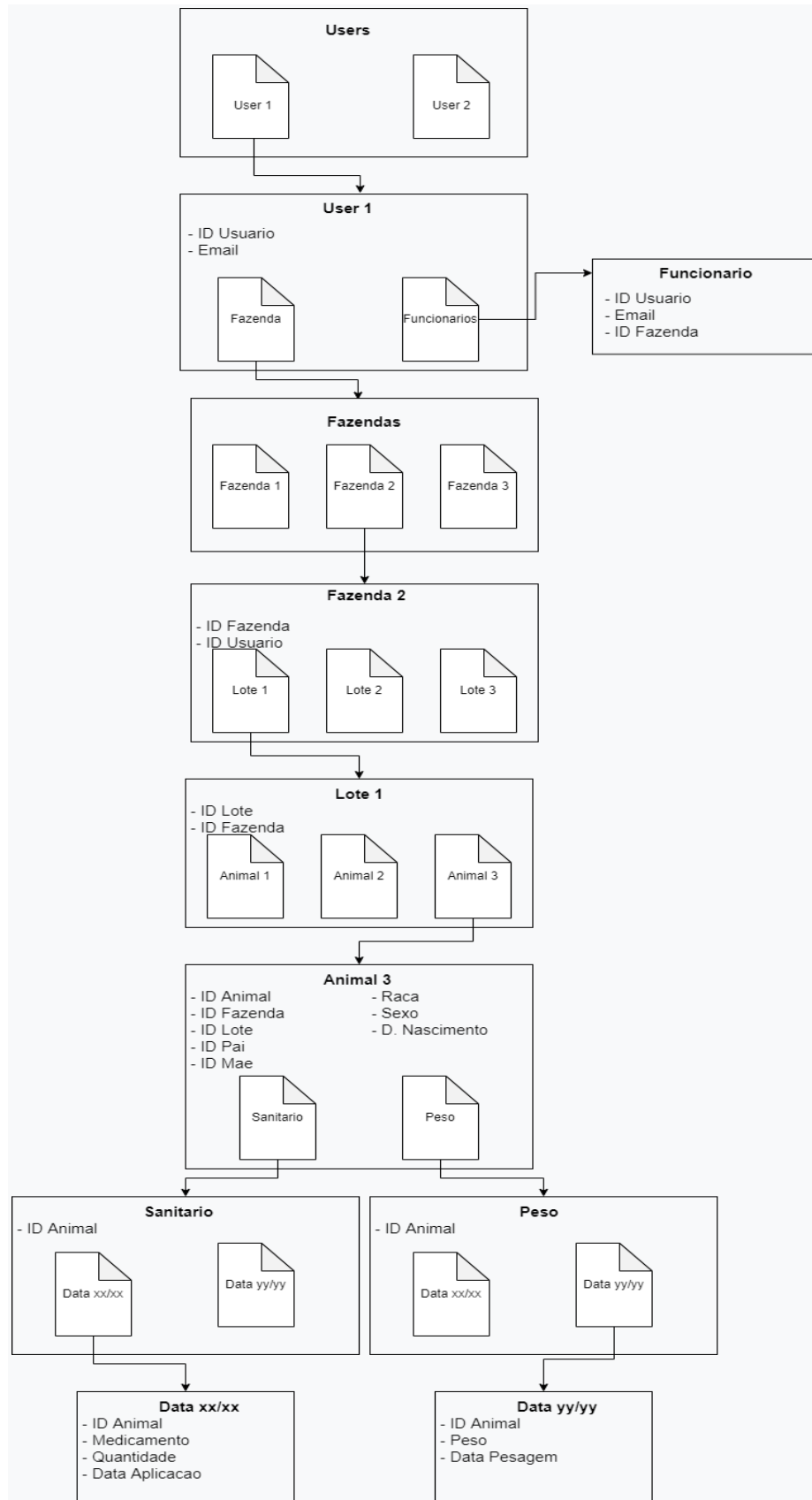


Figura 2 - Esquema do Banco de Dados





## 4. INTERFACES

Nesta parte serão mostrados exemplos do funcionamento da aplicação realizadas em ambiente de desenvolvimento para verificar se o projeto estava executando conforme esperado. Para os testes, foram utilizados um dispositivo móvel com o aplicativo instalado e um computador com o aplicativo usando o navegador Chrome. Para o funcionamento do aplicativo foram criadas 3 contas, duas de administrador e uma de funcionário.

O primeiro acesso ao aplicativo deverá ocorrer através de um navegador, seja ele em dispositivo móvel ou computador. Caso o usuário acesse o web app por um navegador compatível com as funcionalidades de um PWA ele irá se depara com a tela principal, como na **Figura 3**. Essa tela mostra 5 ícones que podem ser clicados pelo usuário e redireciona para as diferentes telas. Aqui também podemos ver que há um banner de rodapé perguntando se deseja baixar o aplicativo para o dispositivo. Caso o usuário escolha **SIM** o download do aplicativo ocorre automaticamente, se escolher **DEPOIS** o banner desaparece até que a página seja recarregada ou se clicar **NUNCA** o rodapé desaparece até que o *cache* do navegador seja limpo.

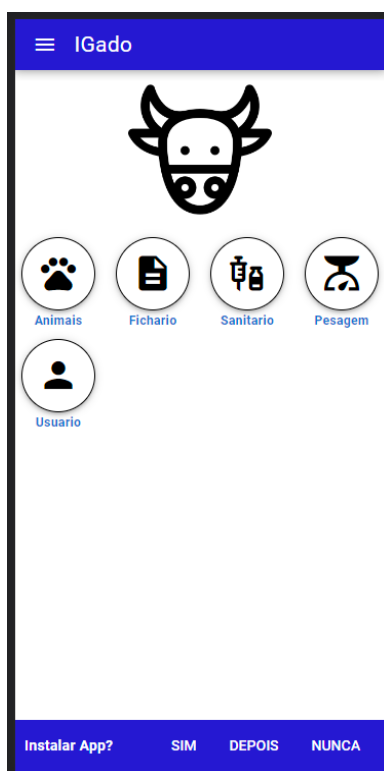


Figura 3- Tela Principal

O menu lateral, ilustrado pela **Figura 4**, pode ser acessado clicando nas três barras no canto esquerdo do cabeçalho. Ele também possibilita rápido acesso às telas do aplicativo sem precisar voltar à tela principal. Caso o usuário não tenha realizado o *login* ele não será capaz de realizar as ações do aplicativo e verá o menu lateral com as duas opções de *LOGIN* e *NOVO USUÁRIO*.

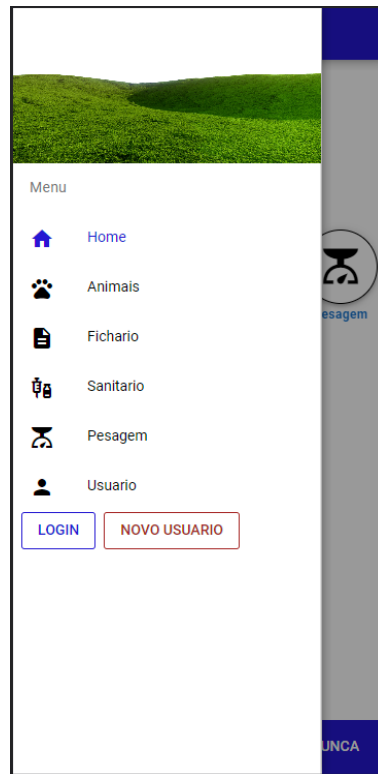
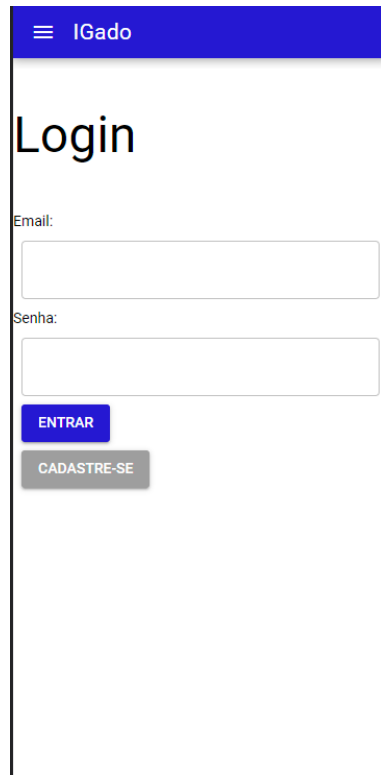


Figura 4- Barra Lateral antes do *login*

Selecionando a opção de *Login* o usuário será direcionado a tela da **Figura 5**, onde deve informar seu *e-mail*, senha. Após preencher os campos, o usuário clica em *Entrar* e os dados são enviados para o Firebase que autentica eles.

Se a solicitação de *login* foi aceita, redireciona para a tela principal e o menu lateral passa a informar qual conta está logada no momento, como ilustra a **Figura 6**. Caso seja negada mostra uma mensagem de erro. Para que a solicitação seja aceita é preciso que o *e-mail* esteja cadastrado e sua senha esteja correta.



The screenshot shows the login interface for IGado. At the top, there is a blue header with a hamburger menu icon and the text "IGado". Below the header, the word "Login" is displayed in a large, bold font. Underneath, there are two input fields: "Email:" and "Senha:". Below the "Senha:" field, there are two buttons: a blue "ENTRAR" button and a grey "CADASTRE-SE" button.

Figura 5- Tela de Login

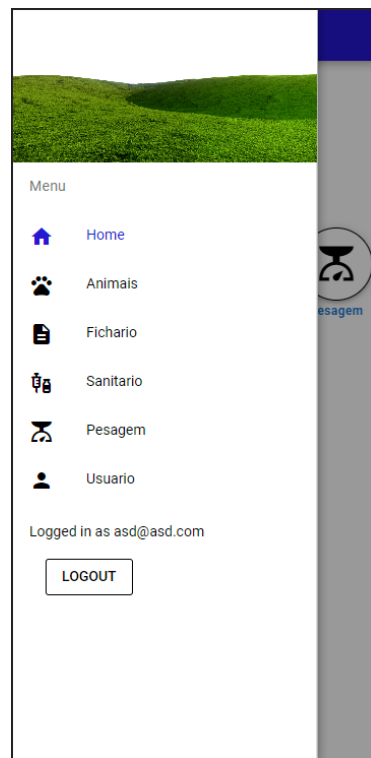


Figura 6- Menu Lateral após login

Caso o usuário não possua uma conta e clicou na opção CADASTRE-SE na tela de *login* ou NOVO USUÁRIO no menu lateral para ser direcionado a página de cadastro, como mostra a **Figura 7**, onde o usuário insere seu nome, seu *e-mail* e uma senha válida.



A imagem mostra a interface de usuário para o cadastro de um novo usuário no aplicativo IGado. No topo, há uma barra azul com o ícone de menu e o nome 'IGado'. O título principal da tela é 'Cadastro de Usuario'. Abaixo, há três campos de entrada de texto rotulados 'Nome:', 'Email:' e 'Senha:'. Um botão azul com o texto 'CADASTRAR' em branco está posicionado abaixo dos campos.

Figura 7- Tela de Cadastro de Usuário

Após a validação do cadastro, o usuário é redirecionado para a tela de cadastro de fazenda, **Figura 8**. Nessa tela o usuário possui três campos: o Nome da Fazenda, Endereço e Estado (UF), sendo apenas o nome obrigatório e os outros opcionais. Ao preencher o campo e clicar no botão, as informações são enviadas para o Firestore onde uma subcoleção com o nome da fazenda é criada, sendo ela necessária para realizar outras operações.

Após o cadastro da fazenda o usuário retorna à tela principal e pode começar a utilizar as funcionalidades do aplicativo.



IGado

Cadastrar Fazenda

Nome da Fazenda\*

Endereco

Estado

ADICIONAR FAZENDA CANCELAR

Figura 8- Tela de Cadastro de Fazenda

Possuindo acesso aos outros links da tela principal o usuário pode clicar no link do Usuário e ser direcionado a **Figura 9**.



IGado

Menu de Usuario

CADASTRAR FUNCIONARIO

CADASTRAR FAZENDA

Figura 9- Tela de Menu de Usuário

Nessa tela o usuário possui duas operações possíveis, a de CADASTRAR FAZENDA, caso o usuário necessite registrar outra propriedade ou caso não tenha realizado essa operação após seu cadastro. A outra operação possível é CADASTRAR FUNCIONÁRIO, ao clicar nesse link será aberto a **Figura 10**, onde o usuário preenche o Nome do funcionário, um *e-mail* e senha para realizar o login e o nome da fazenda onde está empregado. Ao clicar no botão as informações são adicionadas ao sistema de autenticação do firebase e também são adicionadas à coleção Funcionário do banco de dados.



A imagem mostra a tela de cadastro de um funcionário no aplicativo IGado. No topo, há uma barra azul com o ícone de menu e o nome 'IGado'. O título principal da tela é 'Cadastro de Funcionario'. Abaixo do título, há quatro campos de entrada de texto, cada um com um rótulo à esquerda: 'Nome:', 'Email:', 'Senha:' e 'Fazenda:'. Os campos de entrada são retangulares e vazios. Na base do formulário, há um botão azul com o texto 'ADICIONAR FUNCIONARIO' em letras brancas.

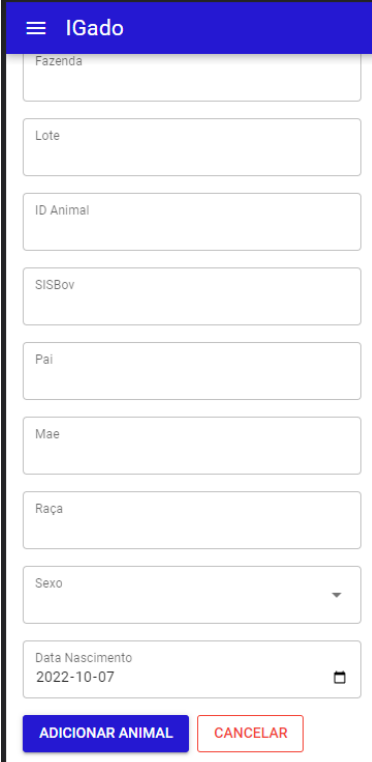
Figura 10- Tela de Cadastro de Funcionário

A diferença entre uma conta de funcionário e de administrador é o nível de acesso a funcionalidades do aplicativo, podendo a conta funcionário realizar operações apenas na fazenda onde está empregado e não possui acesso às telas de Cadastro de Fazenda e de Funcionário.

Retornando a Tela Principal o usuário pode acessar a tela de Fichário, conforme mostra a **Figura 11**. Nela o usuário preenche os campos:

- ID: se refere ao número de identificação presente no brinco utilizado para controle interno. Pode ser preenchido manualmente ou através da leitura de um brinco identificador
- SISBOV: é o número dado ao bovino caso seja destinado à exportação.
- Fazenda e Lote: são os IDs da fazenda e número do lote, respectivamente.
- Raça: qual raça bovina pertence.
- Sexo: indica o sexo biológico.
- Pai e Mãe: são os IDs dos pais para controle de parentesco e genético.
- Data de Nascimento: é o dia, mês e ano em que o animal nasceu.

Antes de enviar os dados para o Firestore, o aplicativo faz duas verificações: se a fazenda está cadastrada pelo usuário e se o animal ainda não foi cadastrado. Caso nenhuma dessas verificações retorne um erro os dados são enviados e armazenados no banco de dados, e o aplicativo retorna uma notificação de sucesso.



The image shows a mobile application interface for adding a new animal record. The form is titled 'IGado' and contains the following fields:

- Fazenda
- Lote
- ID Animal
- SISBov
- Pai
- Mae
- Raça
- Sexo (dropdown menu)
- Data Nascimento: 2022-10-07 (with a calendar icon)

At the bottom of the form, there are two buttons: 'ADICIONAR ANIMAL' (blue) and 'CANCELAR' (red).

Figura 11- Tela de Fichário

Possuindo algum bovino cadastrado o usuário pode buscar suas informações acessando a página Animais, ilustrada pela **Figura 12**. Como podemos ver pela imagem a identificação é inserida na barra de pesquisa acima e o resultado é



mostrado no quadro abaixo. Caso o ID não pertença a nenhum cadastro a tela permanece em branco.

Os dados desta tela são armazenados no cache do navegador para que o usuário possa acessar estas informações mesmo quando estiver offline.



Figura 12- Tela de Animais Cadastrados

Uma questão de suma importância na criação de bovinos é com a saúde deles, por isso o aplicativo possui um registro sanitário, demonstrado pela **Figura 13**. Nessa tela o usuário novamente preenche os campos e escolhe o medicamento a ser aplicado de uma lista, exemplificado pela **Figura 14**.

Após o usuário inserir as informações e clicar em enviar, o aplicativo verifica se existe um cadastro na fazenda com o ID informado pelo usuário. Caso não haja, ele retorna uma mensagem de erro, caso contrário ele cria um documento na coleção Sanitário com as informações inseridas.

IGado

Sanitario

ID Animal  
900001000533338

Fazenda  
Dourados

Lote  
1

Medicamentos  
Ripercol L 150 F

Quantidade Medicamento  
15

Unidade  
ml

Data de Aplicacao  
2022-09-28

ADICIONAR ANIMAL CANCELAR

Busca

Figura 13- Tela de Registro Sanitário

IGado

Sanitario

ID Animal  
900001000533338

Fazenda  
Dourados

Medicamentos  
Ripercol L 150 F

Sincrosin

Ripercol L 150 F

Bovitam

Ciproloc

Unidade  
ml

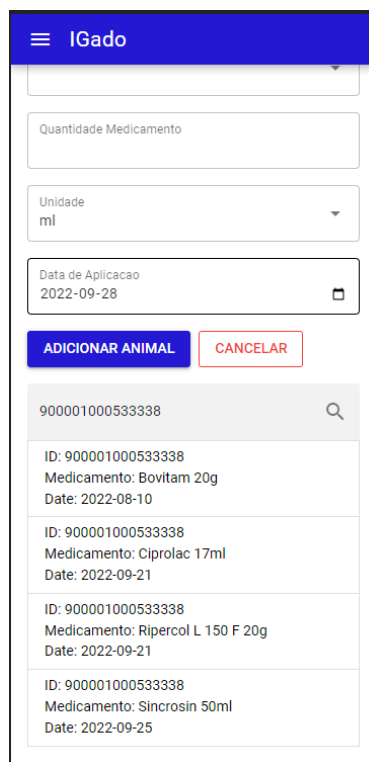
Data de Aplicacao  
2022-09-28

ADICIONAR ANIMAL CANCELAR

Busca

Figura 14- Opções de Medicamento

Realizado o cadastro de medicação o usuário pode buscar por todos os medicamentos aplicado a um id utilizando a barra de busca na parte inferior da página. A **Figura 15** mostra o resultado de uma busca no banco de dados realizada com sucesso.



The screenshot displays the 'IGado' application interface. At the top, there is a blue header with a hamburger menu icon and the text 'IGado'. Below the header, there are several input fields: a search bar (partially visible), 'Quantidade Medicamento', 'Unidade' (set to 'ml'), and 'Data de Aplicacao' (set to '2022-09-28'). Below these fields are two buttons: 'ADICIONAR ANIMAL' (blue) and 'CANCELAR' (red). The main content area shows a search result for the ID '900001000533338'. The results are listed in a table-like format:

ID	Medicamento	Date
900001000533338	Bovitam 20g	2022-08-10
900001000533338	Ciproloc 17ml	2022-09-21
900001000533338	Ripercol L 150 F 20g	2022-09-21
900001000533338	Sincrosin 50ml	2022-09-25

Figura 15- Busca no Registro Sanitário

Outro aspecto importante em fazendas de gado de corte é o controle do peso dos bovinos que pode ser realizado através do aplicativo acessando o link de Pesagem, exposto pela **Figura 16**. Ao preencher os campos da tela e clicar em enviar o aplicativo verifica se o ID está cadastrado, caso esteja cria um documento na coleção Peso com as informações.

Similar a tela de Sanitário, o tela de pesagem possui uma barra de pesquisa em sua parte inferior que pode ser usada para encontrar todas as pesagens anteriores de um animal cadastrado, representado pela **Figura 17**.

IGado

**Pesagem**

ID Animal

Fazenda

Lote

Peso

Data de Aplicacao  
2022-09-30

**ADICIONAR PESO** **CANCELAR**

Busca

Figura 16- Tela de Pesagem

IGado

Lote

Peso

Data de Aplicacao  
2022-10-05

**ADICIONAR PESO** **CANCELAR**

900001000533338

ID: 900001000533338 Peso: 250 Kg Date: 2022-10-05
ID: 900001000533338 Peso: 200 Kg Date: 2022-09-05
ID: 900001000533338 Peso: 100 Kg Date: 2022-08-05
ID: 900001000533338 Peso: 80 Kg Date: 2022-07-05

Figura 17- Busca por Pesagens



## 5. CONCLUSÃO

Com o rápido crescimento e expansão de áreas rurais e as rigorosas exigências dos mercados interno e externo, cada vez mais será preciso métodos flexíveis de controle dos dados de grandes e pequenos rebanhos, de forma rápida e eficiente. Considerando-se as atuais limitações e possíveis problemas que podem ocorrer, foram as razões para a criação deste aplicativo.

Através de estudos foi possível verificar a viabilidade de criar um aplicativo em PWA que suportasse, não apenas, as necessidades diárias de uma fazenda pecuarista, mas também a sua expansão, possibilitando a coleta e verificação dos dados de forma rápida e eficiente em locais fechados ou em campo. Com a ajuda dos bancos de dados na nuvem, os usuários garantem que seus dados estarão seguros, atualizados e de fácil acesso, quando possuírem conexão com a internet. Utilizando essas tecnologias foi possível o desenvolvimento do aplicativo exposto neste trabalho, que poderá facilitar aos pecuaristas estarem informados sobre seu rebanho a qualquer hora e em qualquer lugar.

### 5.1 Trabalho Futuros

Mesmo havendo desenvolvido o aplicativo, como tudo na área de computação, melhoras podem ser feitas. Sendo algumas delas:

- Cadastro e login de usuários através de redes sociais: O Firebase já possui essa funcionalidade e pode ser implementada em alguns passos.
- Melhoras em elementos da interface: Alguns elementos estéticos podem ser melhorados para deixar o aplicativo mais atraente visualmente e fazer com que alguns campos possam ser selecionados através de alguns cliques, sem necessidade de digitar.
- Conectividade com bastões de RFID: As empresas que fabricam os bastões e brincos criptografam os sinais de seus aparelhos, dificultando a conexão com dispositivos sem software de terceiros.
- Funcionalidade para outros tipos de produção: O principal foco do aplicativo era voltado a gado de corte, mas com algumas adições seria possível fazer a integração para fazendas leiteiras e de criação.

- Implementação de Câmera: Integrar a câmera de dispositivos móveis para que caso o produtor deseje salvar uma imagem do animal.
- Conexão com balanças por bluetooth: Algumas balanças possuem conectividade *bluetooth*, possibilitando o envio automático de pesos lidos para o aplicativo.

## REFERÊNCIAS BIBLIOGRÁFICAS

Voulodimos, A. S., Patrikakis, C. Z., Sideridis, A. B., Ntafis, V. A., & Xylouri, E. M. (2010). A complete farm management system based on animal identification using RFID technology. *Computers and electronics in agriculture*, 70(2), 380-388.

PWA. PWA. Disponível em <<https://web.dev/progressive-web-apps/>>. Acesso em 05/04/2022 às 13h10min.

JETBOV. Jetbov. Disponível em <<https://jetbov.com/>>. Acesso em 14/09/22 às 11h30min.

BOVEXO. Bovexo. Disponível em <<https://bovexo.com/>>. Acesso em 14/09/22 às 11h30min.

FERT. Fertili. Disponível em <<https://fertili.com.br/>>. Acesso em 14/09/22 às 11h30min.

OFFLINE. Arctype. Disponível em <[https://arctype.com/blog/pwa-sqlite/#:~:text="Offline%20first"%20is%20an%20application,often%20built%20on%20this%20paradigm](https://arctype.com/blog/pwa-sqlite/#:~:text=)>. Acesso em 23/11/2022 as 14h55min.

Russel, A. Progressive Web Apps: Escaping Tabs Without Losing Our Soul  
Disponível em <<https://medium.com/@slightlylate/progressive-apps-escaping-tabs-without-losing-our-soul-3b93a8561955>>. Acesso em 15/09/22 às 15h40min.

SHEPPARD, Dennis; SHEPPARD, Dennis. *Beginning progressive web app development*. Apress, 2017.

Steiner, T. (2018, April). What is in a web view: An analysis of progressive web app features when the means of web access is not a web browser. In *Companion Proceedings of The Web Conference 2018* (pp. 789-796).



QUASAR. Quasar. Disponível em <<https://quasar.dev/>>. Acesso em 05/04/2022 as 13h15min

FLIPKART. Flipkart. Disponível em <<https://web.dev/flipkart/>>. Acesso em 15/09/22 às 15h40min.

ALI. Aliexpress. Disponível em <<https://web.dev/aliexpress/#results>>. Acesso em 15/09/22 às 15h40min.

TWITTER. Twitter. Disponível em <[https://web.dev/twitter/#lowering\\_data\\_consumption](https://web.dev/twitter/#lowering_data_consumption)>. Acesso em 15/09/22 às 15h40min.

FIREBASE. Firebase. Disponível em <<https://firebase.google.com/>>. Acesso em 15/05/2022 às 10h20min.

FIRESTORE. Firestore. Disponível em <<https://firebase.google.com/docs/firestore>>. Acesso em 15/05/2022 às 10h20min.

NOSQL. Amazon NoSQL. Disponível em <<https://aws.amazon.com/pt/nosql/>>. Acesso em 15/05/2022 às 10h20min.

Biørn-Hansen, A., Majchrzak, T. A., & Grønli, T. M. (2017, April). Progressive web apps: The possible web-native unifier for mobile development. In International Conference on Web Information Systems and Technologies (Vol. 2, pp. 344-351). SciTePress.

Trindade, P. E. (2020). O Progressive Web Apps–PWA: como ferramenta para a produção audiovisual.

TAG, Radio Frequency Identification for Beef Cattle. Disponível em <[https://www.thebeefsite.com/articles/821/radio-frequency-identification-for-beef-cattle#:~:text=The%20present%20RFID%20tags%20used,of%20134.2%20kHz%20\(kilohertz\).](https://www.thebeefsite.com/articles/821/radio-frequency-identification-for-beef-cattle#:~:text=The%20present%20RFID%20tags%20used,of%20134.2%20kHz%20(kilohertz).>)> Acesso em 10/10/2022 às 16h10min

