

PONTFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS  
PRÓ-REITORIA DE GRADUAÇÃO  
ESCOLA DE CIÊNCIAS AGRÁRIAS E BIOLÓGICAS  
CURSO DE ZOOTECNIA

**ZOOTECNISTA COMO DESENVOLVEDOR DE NOVAS  
TECNOLOGIAS: APLICATIVO PARA BALANCEAMENTO  
NUTRICIONAL DE BOVINOS EM CONFINAMENTO**

Náthanno Guilherme

Orientador: Dr. Verner Eichler

Goiânia – Goiás

2021



**NÁTHANNO GUILHERME**



**ZOOTECNISTA COMO DESENVOLVEDOR DE NOVAS  
TECNOLOGIAS: APLICATIVO PARA BALANCEAMENTO  
NUTRICIONAL DE BOVINOS EM CONFINAMENTO**

Trabalho de conclusão de curso apresentado como requisito parcial para obtenção do grau de Zootecnista, junto Escola de Ciências Agrárias e Biológicas, da Pontifícia Universidade Católica de Goiás.

Orientador: Prof. Dr. Verner Eichler

Goiânia – Goiás

2021



**NÁTHANNO GUILHERME**



**ZOOTECNISTA COMO DESENVOLVEDOR DE NOVAS  
TECNOLOGIAS: APLICATIVO PARA BALANCEAMENTO  
NUTRICIONAL DE BOVINOS EM CONFINAMENTO**

Monografia apresentada à banca avaliadora em \_\_\_/\_\_\_/\_\_\_ para conclusão da disciplina de TCC, no curso de Zootecnia, junto a Escola de Ciências Agrárias e Biológicas da Pontifícia Universidade Católica de Goiás, sendo parte integrante para o título de Bacharel em Zootecnia.

Conceito final obtido pelo aluno: \_\_\_\_\_

\_\_\_\_\_  
Prof. Dr. Verner Eichler

(Orientador)

\_\_\_\_\_  
Prof. Dr. Otávio Cordeiro de Almeida

PUC GOIÁS

(Co-orientação)

\_\_\_\_\_  
Prof. Dr. Luiz Carlos Barcelos

PUC GOIÁS

(Co-orientação)

Dedico este

A meu Deus que me traz sentido e  
inspiração para vida.

## **AGRADECIMENTOS**

Acima de tudo agradeço a meu Deus que me dá todos os dias uma nova chance, que no vazio mais profundo da depressão me acolheu e trouxe inspiração e forças para viver, me mostrou o seu amor trazendo pessoas especiais para minha história, me ergueu e abriu meus olhos me fazendo enxergar que diante dele tudo posso conquistar.

A minha família e a todos que contribuíram para que este momento se tornasse possível.

Nada e ninguém no mundo vai me  
fazer desistir. Quero, tudo quero,  
sem medo entregar meus projetos  
[...]

[...] Vou persistir, continuar a  
esperar e crer. E mesmo quando a  
visão se turva e o coração só chora  
Mas na alma, há certeza da vitória.

Composição: Celina Borges

## SUMÁRIO

<b>LISTA DE FIGURAS</b> .....	ix
<b>LISTA DE TABELAS</b> .....	x
<b>LISTA DE ABREVIações</b> .....	xi
<b>RESUMO</b> .....	xii
<b>ABSTRACT</b> .....	xiii
<b>1. INTRODUÇÃO</b> .....	01
<b>2. REVISÃO DE LITERATURA</b> .....	03
<b>2.1 Aplicativo para Android</b> .....	03
<b>2.2 Logotipo mini conf.</b> .....	05
<b>2.3 Main.dart.</b> .....	06
<b>2.4 Home view (tela de início)</b> .....	07
<b>2.5 Home menu</b> .....	09
<b>2.6 Drawer</b> .....	10
<b>2.7 Estimativa de consumo de água</b> .....	11
<b>2.8 Tabelas de composições</b> .....	12
<b>2.9 Aspectos nutricionais em confinamentos</b> .....	14
<b>2.10 Adaptação de consumo de matéria seca</b> .....	16
<b>3. Materiais e métodos</b> .....	17
<b>3.1 Code::Blocks</b> .....	18
<b>3.2 Estrutura do programa</b> .....	19
<b>3.3 Controle dos Fluxos de Entrada e de Saída</b> .....	20
<b>3.4 Criação do mobile</b> .....	21
<b>3.5 Ambiente de trabalho</b> .....	21
<b>3.6 Controle dos fluxos de entrada e de saída dart em flutter ..</b>	24
<b>3.7 Tabelas</b> .....	25
<b>4. Resultados e discussões</b> .....	26
<b>5. CONSIDERAÇÕES FINAIS</b> .....	36
<b>6. REFERÊNCIAS BIBLIOGRAFICAS</b> .....	37
<b>Anexos</b> .....	41

## LISTA DE FIGURAS

Figura 01: Logotipo Mini conf. ....	05
Figura 02: <i>Print</i> tela da <i>Home view</i> (tela de início) .....	08
Figura 03: <i>Print</i> tela <i>Home menu</i> .....	09
Figura 04: <i>Print</i> extremidade superior do <i>home menu</i> .....	10
Figura 05: <i>Print drawer</i> mini conf. ....	11
Figura 06: <i>Print</i> estimativa de consumo de água .....	12
Figura 07: <i>Print</i> tela de tabela de silagens .....	13
Figura 08: Ambiente de trabalho <i>Code::Block</i> .....	18
Figura 09: Fluxograma <i>Code:Block</i> .....	20
Figura 10: <i>Plugins Flutter e Dart</i> no <i>Android Studio</i> .....	22
Figura 11: <i>Plugins Flutter e Dart</i> no <i>Visual Studio Code</i> .....	23
Figura 12: Ambiente de trabalho <i>Visual Studio Code</i> .....	23
Figura 13: Fluxograma sistema Mini Conf .....	24
Figura 14: Plataforma BR-Corte 4 .....	25
Figura 15: <i>Print</i> parte da tela de formulação .....	16
Figura 16: <i>Print</i> tela de resultados .....	24



**LISTA DE TABELAS**

Tabela 01: Tabela de Silagens composição química .....	26
--	----

**LISTA DE ABREVIACOES**

App = Aplicativo .....	03
Ca= Clcio .....	04
CMS = Consumo de matria seca .....	04
CQBAL = Tabela de composio qumica e bromatolgicas de alimentos.	04
EED = Extrato etreo digestvel .....	04
ENND = Extrativo no nitrogenado digestvel .....	04
FBD = Fibra Bruta digestvel .....	04
GPV = Ganho de peso vivo .....	15
IMS = Ingesto de matria seca .....	15
NDT = Nutrientes digestveis totais .....	04
P = Fosforo .....	04
PB= Protena bruta .....	04
PBD = Protena bruta digestvel .....	04
PV = Peso vivo .....	15
STL = <i>Stateless</i> .....	07
STF = <i>Stateful</i> .....	10

## RESUMO

O Brasil situa-se como um dos principais protagonistas na produção e comercialização de carne bovina mundial, resposta de um acentuado processo de desenvolvimento que elevou a produtividade a níveis tecnológicos, conseqüentemente mantendo sua posição e extensão de mercado. O aperfeiçoamento tecnológico se consolidou, ao decorrer das décadas passadas, como um dos principais fatores que garantiram o crescimento do sistema de produção. A dieta dos animais alojados no sistema de produção é um fator crítico, capaz de trazer conseqüências quantitativas como também qualitativas, desenvolver e viabilizar ferramentas capazes de absorver as necessidades nutricionais para uma maior produtividade animal que trazem panoramas economicamente mais viáveis é a realidade do setor de produção de bovinos tipo carne. É nesta busca por desenvolvimento de inteligência e automação que este trabalho se objetiva em aliar as habilidades transversais, as competências zootécnicas, para o desenvolvimento de um aplicativo prático e funcional, capaz de gerar resultados básicos para o balanceamento nutricional, requerimentos energéticos, proteicos, hídricos e minerais.

Palavras-chave: Dieta, inteligência, resultados.

## **ABSTRACT**

Brazil is one of the main protagonists in the production and marketing of beef in the world, a response to a strong development process that has raised productivity to technological levels, consequently maintaining its position and market extension. Technological improvement has consolidated, over the past decades, as one of the main factors that ensured the growth of the production system. The diet of animals housed in the production system is a critical factor, capable of bringing quantitative as well as qualitative consequences, develop and enable tools able to absorb nutritional needs for greater animal productivity that bring economically more viable panoramas is the reality of the beef-type cattle production sector. It is in this search for development of intelligence and automation that this work aims to combine cross-cutting skills, zootechnical skills, for the development of a practical and functional application, capable of generating basic results for nutritional balance, energy, protein, water and minerals requirements.

**Keywords:** Diet, intelligence, results.

## 1. INTRODUÇÃO

Há várias tecnologias sendo desenvolvidas que buscam aprimoramento em suas funcionalidades na procura de otimizar processos e melhorias do manejo dos animais. Hoje, observam-se no mercado variedades de sistemas de auxílio ao técnico e produtor, porém com alguns pontos diferentes que os caracterizam.

A crescente busca pelo aumento da produtividade, aperfeiçoamento da produção e redução de custos, tem possibilitado ao Brasil uma inserção importante no mercado internacional. O uso de tecnologias especializadas e o investimento realizado em todo o processo de produção tem permitido ao país um ganho de mercado (Donatti, 2017).

O Brasil tem se destacando como um dos principais países em fornecimento de carne bovina no mercado externo, principalmente para aqueles onde o preço ainda é um fator importante, além é claro da qualidade do produto fornecido (Pereira et al. 2011).

A Zootecnia de precisão está relacionada com a maximização dos índices de produtividade e a diminuição de perdas, dentro de todo o processo de produção, buscando otimização utilizando-se de tecnologias para requerir as informações ou até mesmo para a aplicação das ações desejadas. Logo, a Zootecnia de precisão estabelece uma grande relação com a eficiência produtiva e a tecnologia.

Segundo Carvalho (2018), o Brasil é detentor do segundo maior rebanho mundial, possuindo 26,64% do rebanho total, já os Estados Unidos, obtém o quarto maior rebanho, porém é o mais eficiente com a maior produção de carne comercial, essa eficiência alcançada se remete as tecnologias e controles zootécnicos de ponta.

Para uma melhor tomada de decisões com maior eficiência, é necessário contar com dados confiáveis buscando o mais próximo possível da realidade, sendo a coleta realizada por estimativas humanas ou por meio de tecnologias que buscam a facilitar e gerar mais rápido e com segurança os dados precisos dentro de um sistema de produção.

De acordo com Tullo, Fontana e Guarino (2013), a automatização deve funcionar em qualquer local e em qualquer condição, porém sempre dependente

da manipulação manual esse passo é fundamental para a correta análise dos dados.

O presente estudo tem por objetivo aliar as habilidades transversais, as competências zootécnicas, para o desenvolvimento de um aplicativo prático e funcional, capaz de gerar resultados básicos para o balanceamento nutricional, requerimentos energéticos, proteicos, hídricos e minerais.

## 2. REVISÃO DE LITERATURA

Para melhor entender o que é o Mini Conf. a construção de uma revisão de literatura, esclarecerá as vias utilizadas e criadas para desempenhar uma ferramenta compacta e capaz de atender as necessidades mínimas em ambientes remotos ou viabilizar o veículo de uma ferramenta em um aparelho celular.

Para o desenvolvimento do projeto Mini Conf. serão utilizadas duas sintaxes de linguagens, uma definindo a organização do esqueleto e esquemas algébricos utilizados para a modulação as análises e resultados, sendo esta usada a linguagem C++.

Na construção do aplicativo, para Android, o emprego da linguagem Dart e o framework flutter trazem funcionalidades que facilitaram o desenvolvimento limpo e simplificado nas aplicações.

### 2.1 Aplicativo para Android

A partir do momento em que o aplicativo (app), é desenvolvido na linguagem Dart orientada a objetos, utilizando o *framework flutter*, o diagrama descrito na figura 18 representa claramente os caminhos que podem ser percorridos até a obtenção dos resultados.

A uma crescente demanda por informações técnicas e atuais na agropecuária e a internet tem sido um excelente veículo. O desenvolvimento de ferramentas digitais como aplicativos móveis, ganham o cenário gradativamente no setor de produção animal demonstrando eficácia na difusão de conhecimento técnico gerado por pesquisadores e levados para o campo (Alves et al, 2019).

É visto que no mercado existem software com mais funcionalidades e aplicações, porem o fato de serem dependentes do uso de computadores ou conexão a redes, os tornam limitados sendo muitos até complexos o seu entendimento.

A grande cartada empregada nesta ferramenta, pode ser notada até mesmo na simplicidade e funcionalidade de operação de suas linhas de código, pois o Mini Conf. traz a praticidade de manuseio, para suprir as necessidades básicas do operador e exigências nutricionais de bovinos confinados. Desenvolvido

para operar em situações remotas, onde o produtor ou técnico não tem acesso a banco de dados online (dados moveis ou redes wireless).

O Mini Conf. opera com seu próprio banco de dados offline trazendo tabelas com exigências nutricionais, composições químicas dos alimentos mais utilizados, e um sistema operacional capaz de balancear dietas utilizando como faixas de segurança NDT (nutrientes digestíveis totais), PB (proteína bruta), Ca(cálcio) e P(fósforo).

Medeiros et al, (2015), traz que o NDT é um dos modos mais empregados de expressão de energia. Ele representa a soma das frações digestíveis dos alimentos de acordo com as análises de Wendee (Sistema Proximal): proteína digestível (PBD), fibra bruta digestível (FBD), extrativo não nitrogenado digestível (ENND) e extrato etéreo digestível (EED), conforme equação:

$$\text{NDT (\%)} = \%PBD + \%FBD + \%ENND + (\%EED \times 2,25).$$

A energia é o componente que o animal utiliza para realizar funções como movimentação, metabolismo, produção crescimento e outras.

A PB tem papel fundamental na nutrição de ruminantes, a proteína encontrada nos alimentos consumidos pelos animais e calculada como Nitrogênio x 6,25. As proteínas são macromoléculas que desempenham inúmeras funções no organismo animal, como a composição de tecidos estruturais, enzimas, hormônios, receptores hormonais e composição de material genético (Clarindo, 2006).

Para a faixa de segurança dos minerais utiliza-se o Ca e o P já que são os minerais mais abundantes no organismo animal, afirmativa que é reforçada por Valadares Filho et al. (2016) onde descrevem que o Ca é o primeiro dos macros minerais mais encontrado no corpo do animal e que está presente em maior parte nos ossos e em segundo está o P também com sua maior parte presente nos ossos, porém ambos também se distribuem nos fluidos e tecidos animal.

A viabilização da aplicação desta ferramenta no campo, extrapola sua aparência simples e compacta, pois, levando em consideração que existem partes no país em que não há cobertura de redes de comunicação desenvolver ferramentas compactas e de fácil entendimento, capazes de operar em aparelhos mais simples, torna a produção mais tecnificada e traz progressos em toda a cadeia de produção do sistema.



## 2.2 Logotipo mini conf.

A primeira impressão é o que leva as pessoas a terem ideia do que nós somos, queremos e pretendemos, considerar este fato nos leva a construirmos características que se tornam nossa identidade. A busca do homem por aumento de produtividade tanto quantitativa quanto qualitativa faz que, a junção de várias técnicas crie diferentes métodos, abordagens e ferramentas, que auxiliam na maximização da produção.

Para Bonfim e Marques (2018), a crescente demanda faz com que haja, uma necessidade de aprimoramento das tecnologias de produção, para que possa ocorrer uma melhoria de desempenho da produtividade. O uso de tecnologia é fundamental na busca de resultados e isto é refletido em todo o país.

A construção do logotipo Mini Conf. (figura 01), conta com a junção de vários elementos que arremetem a um significado maior, a busca pelo equilíbrio através da intervenção do homem através dos estudos das exigências fisiológicas dos animais.



Figura 01: Logotipo Mini conf.

Fonte: *Design* do próprio autor.

Se fragmentada a logo Mini Conf. os elementos nela inserida denotam significados individuais, que compõem a mensagem principal, sendo assim tem-se;

A arroba (figura 01), representativo do objetivo de ganho de peso bovino, rentabilidade econômica e cabeça humana, na gestão dos métodos e planos aplicados na produtividade bovina.

Matos (2020), ressalta que, a importância de um gestor não se prioriza somente nas atividades desempenhadas, mas em todo o funcionamento. Considerando que é referência em todas as atividades desempenhadas, produtos e processos, conduta profissional e ética dentre as demais atribuições.

No logotipo, tem-se a estrutura em “Y” com pêndulos, representando o prolongamento do corpo humano, o equilíbrio e a presença e importância das equações matemáticas para gerir planos e estratégias traçados buscando maior rentabilidade e produção (figura 01).

A precisão dos resultados se encontra entre os fatores de grande importância na computação de cálculos numéricos. Tendo como objetivo obter valores cada vez mais precisos e com menor erro possível (Duarte, 2007).

Englobando todos os elementos já citados, tem-se uma gota de água (figura 01) que traz a simbologia remetida à importância da água em todo o processo de manutenção da vida, e seu fator limitante em todo o processo da cadeia produtiva visto que uma vez ausente ou limitada, a um colapso já que as funções metabólicas não são executadas corretamente levando à estagnação do ganho de peso.

Evidenciando isto, Rodrigues (2018) traz que, no desempenho animal a água é fator limitante, que deve ser disponibilizada em abundância, caso contrário, não se atingira o potencial máximo da produção, mesmo com produtos de qualidade, máquinas, melhoramento genético e outros itens.

### **2.3 Main.dart**

No desenvolvimento das *scripts* para construção e modelação do executável, a construção do arquivo *main.dart* é primordial, pois, este arquivo que

representara o esboço inicial de execução do *framework*, ou seja, este sempre será responsável em carregar os demais arquivos *dart*.

O método *main* da aplicação é responsável por definir quais serão as primeiras instruções do programa a serem executadas, esta é a função responsável por indicar onde o programa deve começar. Toda aplicação em *dart* deve possuir um método *main* para desempenhar seu papel (Silva e Aguiar, 2016).

Desvendar as *scripts* que o compõem, se torna mais simples e compreensível quando fragmentado em partes, sendo assim, encontra-se, nas linhas 1 e 2, arquivos importados e herdados da própria biblioteca *dart*, subsequentemente nas linhas 3 à 12, arquivos de *script* importados, criados pelo autor, da linha 14 à 36, tem-se a função *main* criada, seguida de uma classe STL (*stateless*), responsável em trazer a temática a ser desenvolvida e definindo as rotas automáticas que podem ser percorridas no aplicativo.

Com o *main.dart* estruturado e sabendo que, o mesmo será a responsável em executar todos os outros arquivos subjacentes, parte-se para a construção dos elementos que construirão a aparência e funcionalidade do aplicativo.

## 2.4 Home View (tela de início)

O arquivo *home view*, que configura o que pode ser chamado de tela de início se divide em dois estados, diferente do *main.dart* que não tem o intuito de ser acessado pelo operador, o *home view*, possui um arquivo com as linhas de *scripts* que configuram as funcionalidades e o aspecto da então tela de início, que é apresentada ao operador.

Os *scripts* que compõem esta primeira página, são ordenados através de um STL uma classe de estado fixo, logo que fragmentado vê-se uma ordem predominante em que assim como no *main.dart*, nas primeiras linhas encontram-se *scripts* importados da biblioteca *dart*.

Seguindo da criação de um *widget* denominado *\_body*, composto por todos os elementos que viram a frente do plano de fundo, como o logotipo e botão responsável pelo acesso da próxima página. A arquitetura e disposição destes elementos são influenciados pelo *materialapp*, chamado no *return* do STL da função *main* da *main.dart*, em tem-se a partição que corresponde ao restante do STL contendo os elementos correspondentes ao fundo.

Para melhor entendimento do funcionamento desta configuração de códigos a (figura 02) é o resultado dos elementos descrito.



Figura 02: *Print* tela da *Home view* (tela de início).

Fonte: *Design* do próprio autor.

Dividindo esta imagem em itens consegue-se atribuir ao logotipo um *sizedBox* e um *center*, mantendo-se centralizada e distanciada da margem superior, a mesma característica é atribuída ao *raisedButton*, o distanciando da logo e o centralizando.

Ao fundo encontra-se uma imagem com a função *Scaffold*, função responsável em dar vida ao *background*. Esta imagem recebe um *sizedBox* *height: MediaQuery.of(context).size.height* juntamente com uma *fit: BoxFit.cover*, para que possa se expandir ao máximo em todos os aparelhos, toda esta função recebe um *container* com um *color: colors.black.withOpacity(0.4)*, que diminui o brilho da imagem.

## 2.5 Home menu

Passando do *home view* através de uma rota automática, é imprimido na tela o *home menu*, seguindo no arquivo de código a mesma estrutura lógica, porém possuindo agora uma *STL* e um *STF (stateful)*, um *widget* denominado *\_ted* que comporta todos os itens à frente do *blackgrand* e uma *widget* chamado de *\_doc* configurando todos os itens à frente do *blackgrand* do *drawer*.

Para melhor entendimento, a divisão em dois momentos será necessária, pois, a composição do *Home menu* conta com um *drawer* compacto como *menu*, no primeiro momento serão analisados os elementos da (figura 03).

Nesta tela de início do *home menu* é inserido um *blackgrand* com os mesmos princípios de linhas de código dos demais, três *text* contendo informações para contato e uma mensagem de boas vindas, dois *raisedButton* um levando para a ala de formulação de dietas e outro destinado ao *Info* criadas para auxílio no manuseio do aplicativo.



Figura 03: *Print* tela *Home menu*.

Fonte: *Design* do próprio autor.

Na parte superior encontra-se um *AppBar* que recebe um *text*, no seu canto superior esquerdo é possível notar a inserção do *drawer* (figura 04).



Figura 04: *Print* extremidade superior do *home menu*.

Fonte: *Design* do próprio autor.

## 2.6 Drawer

O *drawer* é uma espécie de *menu hamburger*, muito utilizado para melhor aproveitamento do espaço de uma forma mais compacta. Neste caso o *drawer* (figura 05) é responsável em comportar tabelas de auxílio caso o operador ou técnico não tenha em mãos análises químicas dos alimentos utilizados.

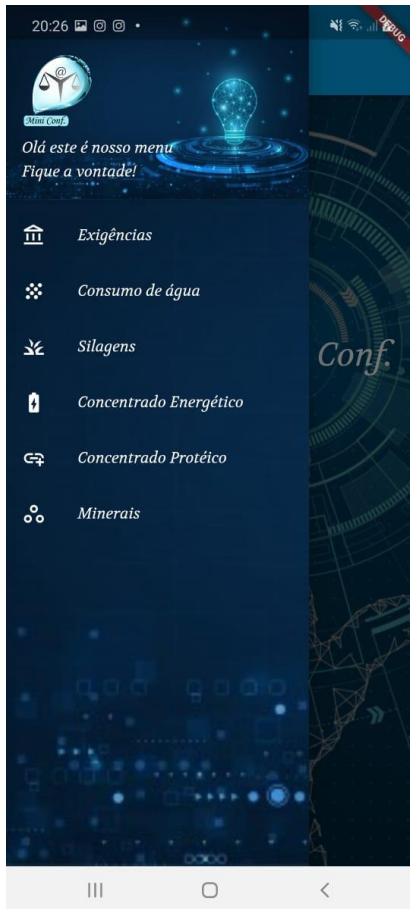


Figura 05: *Print tela Drawer mini conf.*

Fonte: *Design* do próprio autor.

No arquivo de código responsável pelo desenvolvimento deste *drawer*, na parte superior representando o cabeçalho a um *blackgrand* que segue a mesma linha de código que os demais, contudo para a inserção da imagem e textos utiliza-se da função *UserAccountsDrawerHeader*, que está chamada dentro do *widget \_doc*, que também reúne todos os *ListTile* responsável por em caminhar as tabelas de exigências e composições.

## 2.7 Estimativa de consumo de água

As estimativas de consumo de água foram adaptadas de Pereira et al, (2009), que mostra a injeção de água diária necessária em função do seu peso vivo, o oferecimento inadequado (quantidade e qualidade) de água diminui a CMS e prejudica o funcionamento de funções metabólicas diminuindo o desempenho

animal. Para o mesmo autor a água é o nutriente mais importante e mais abundante no corpo do animal em todas as fases de crescimento, desenvolvimento, produção e saúde animal.

Para a construção da página de consumo de água (figura 06) defini-se a classe STL, com um *widget* criado denominado *\_sil, return Container* responsável em configurar a imagem da tabela. Logo abaixo, é responsável pelo *blackgrand* e toda a configuração dos elementos do *AppBar*.

Estimativa Consumo de Água	
Bovino kg	L/Dia
270	36
370	57
455	78

Figura 06: *Print* tela estimativa de consumo de água.

Fonte: *Design* do próprio autor.

## 2.8 Tabelas de composições

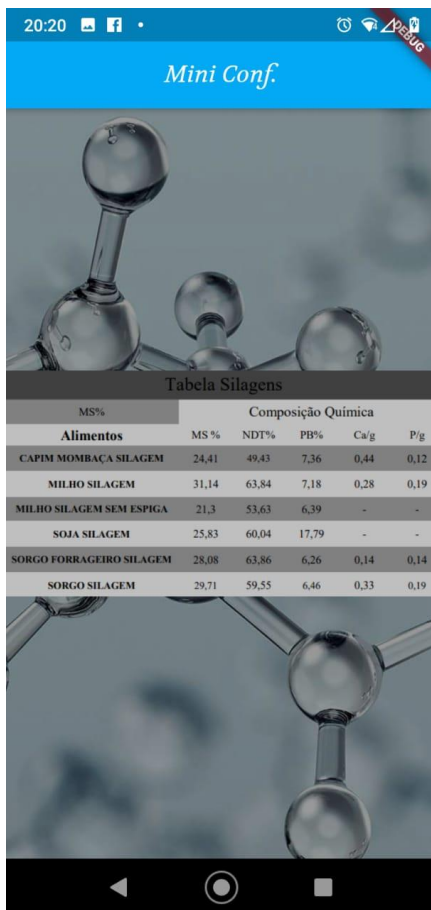
Os dados inseridos nestas tabelas são adaptados do Br-corte 4.0 que por sua vez é alimentado pelas a tabela de composição química e bromatológicas de alimentos (CQBAL). A relação de alimentos nestas inseridos leva em consideração



os que tem maior uma frequência em formulações, podendo alguns serem menos utilizados que outros dependendo da região.

As tabelas inseridas como imagem é fruto da coleta de dados e otimização dos mesmos no *Excel*. A ordem das funções não se difere das já utilizadas, a ordem dos filhos, herdados dos pais ainda é seguida, a tabela é referida apenas como uma imagem inalterável na composição

Os itens que compõem a tela (figura 07), são configurados por uma classe *STL*, recebendo um *blackgrand* com um *container color: Colors.black.withOpacity(0.4)*, um *widget* chamado *\_sil* que leva funções responsáveis pela imagem da tabela.



MS%	Composição Química				
Alimentos	MS %	NDT%	PB%	Ca/g	P/g
CAPIM MOMBACA SILAGEM	24,41	49,43	7,36	0,44	0,12
MILHO SILAGEM	31,14	63,84	7,18	0,28	0,19
MILHO SILAGEM SEM ESPIGA	21,3	53,63	6,39	-	-
SOJA SILAGEM	25,83	60,04	17,79	-	-
SORGO FORRAGEIRO SILAGEM	28,08	63,86	6,26	0,14	0,14
SORGO SILAGEM	29,71	59,55	6,46	0,33	0,19

Figura 07: *Print* tela de tabela de silagens.

Fonte: *Design* do próprio do autor.

Na sequência da configuração encontra-se um *AppBar* com um *text* seguido de um *blackgrand* configurado como os anteriores.

## 2.9 Aspectos nutricionais em confinamentos

Quando se quer traçar melhores estratégias para maximizar um sistema de produção animal, alguns pilares são responsáveis em atuar por todas as fases deste sistema. O confinamento tem a proposta de encurtar o ciclo de produção de carnes explorando ao máximo o potencial genético do rebanho, condicionando-os a um ambiente propício e alimentação condicionada, assim possibilitando melhor qualidade na carne produzida e um menor custo por arroba produzida, isso é intensificação.

Para Benez e Cabral (2015), a pecuária de corte intensiva aponta resultados expressivos para contribuir de maneira significativa na promoção do desenvolvimento do setor de produção de carne bovina no país, já que incentiva a utilização exponencial das diversidades genéticas animal e vegetal.

Reforçando esta expressão, Macitelli (2015) fortalece que a exploração de sistemas intensivos de criação de bovinos vai além da simples ideia de aumento de produtividade e busca por menor preço de produção e melhores remuneração por arroba em períodos de menor disponibilidade de forragens, mas também que esses sistemas de produção caracterizam-se como adjuvantes da agricultura, uma vez que maximizam produtividade aumentam a disponibilidade de áreas para a produção de grãos e outros produtos agrícolas, incentivam o uso de maquinários e mão de obra, além de incentivar o aproveitamento dos resíduos da produção de importantes culturas do panorama agrícola brasileiro.

Para a edificação deste sistema colocar a genética, sanidade, ambiente e nutrição como os pilares é certamente viável já que, estes refletem diretamente na estrutura operacional e produtiva, funcionando como verdadeiros gatilhos para a eficácia e eficiência produtiva tanto quantitativa quanto qualitativa.

Nutricionalmente dentre os confinamentos, tem-se que levar em consideração a existência de diferentes propostas de ganho de peso, e que produzem composições de diferentes estratégias de ganho. É normal tratar ganho de peso como sendo uma coisa só entre os indivíduos de um mesmo piquete, ou seja, correspondendo o ganho de peso vivo diário, ou ganho médio diário, porém a conformação de ossos, músculo e gordura são distintos.

Compreender o consumo qualitativo e quantitativo é de grande importância, logo que esta variável influencia drasticamente o custo de produção. O volume total de nutrientes recebidos pelo animal para o crescimento, saúde e produção são dependentes da relação entre o balanceamento de dietas, manejo nutricional e o consumo voluntário dos alimentos pelos animais (Ferreira, 2019).

A adaptação nutricional dos animais não é o único desafio enfrentado para os recém chegados ao confinamento. Quando os animais são expostos ao novo ambiente de sistema de produção intensivo, em primeiro momento, ocorrerá um período de adaptação ao novo ambiente, considerando que a transição do pasto para o confinamento proporciona grandes mudanças, já que o animal passa de uma dieta rica em fibras para outra mais concentrada, porém, também pela nova dependência gerada ao homem para o atendimento de recursos essenciais como oferta de alimento e água, por exemplo, a oferta dos alimentos em reservatórios, presença de animais de diferentes rebanhos, ausência de sombra e substituição de amplo espaço por área restrita, aliados à alta densidade de animais (Benez e Cabral, 2015).

Um desafio nutricional a ser enfrentado em confinamentos é o consumo irregular de matéria seca. O ajuste inapropriado o balanceamento da oferta de ração, pode fazer com que ocorra um alto consumo ou até mesmo um baixo consumo, ambos provocando desajustes. Os reflexos desses desajustes provocam distúrbios metabólicos, já que esse evento provoca uma sobrecarga nas funções metabólicas dos órgãos do trato digestório, desperdiçando energia que deveria ser convertida em ganho de carcaça. A oscilação do pH no ambiente ruminal é o fator responsável por vários problemas de saúde e desempenho dos animais (Almeida 2005).

Alta participação de concentrado na dieta geralmente proporciona maiores ganhos de peso e conseqüentemente maiores quantidades de gordura na carcaça, pois a composição do ganho em planos nutricionais com objetivos superiores favorece a deposição de gordura.

## 2.10 Adaptação de consumo de matéria seca

Quando se trata de animais destinados a produção de carne, a ocorrência de variações no consumo de alimentos manifesta-se durante seu período de crescimento para buscar o equilíbrio dinâmico do meio interno diante dos acontecimentos constantes apresentados pelas necessidades metabólicas e ambientais.

O CMS é uma das variáveis mais importantes no desempenho animal, principalmente em bovinos destinados para o corte levando em consideração a importância econômica e seu complexo sistema digestivo e suas funções metabólicas.

O CMS é um dos fatores importantes que afetam o desempenho animal, já que fornece ao organismo, os nutrientes e substratos energéticos necessários para que ocorra as reações bioquímicas que contribuem para as oscilações do metabolismo celular, especialmente em bovinos destinados para produção de carne, observando importância econômica e o complexo sistema digestivo com suas funções metabólicas peculiares (Valadares Filho et al., 2016).

Possíveis limitações no consumo de alimentos podem provocar o impedimento na necessidade de suprir as exigências nutricionais. Alterações alimentares na dieta de bovinos de cortes podem promover limitações na sua eficiência, pois, é necessário levar em consideração que grande parte da sua alimentação é responsável apenas por sua manutenção, assim quando alterado pode haver queda em seu rendimento.

Ferreira et al (2013) faz uma consideração que, a uma diferença entre animais se tratando de eficiência de produção que está na capacidade de ingestão de alimentos, em níveis considerados acima das exigências de manutenção, as diferenças são observadas nas quantidades de alimentos consumidos por quilogramas de ganho de peso ou de carcaça produzido.

O consumo pode ser controlado pela qualidade da dieta em termos de digestibilidade e energético, porém ao mesmo tempo que sua relação pode ser positiva também pode ser negativa. Dietas de alta digestibilidade tendem um menor consumo, pois, o animal atenderá com menores níveis de consumo sua demanda

energética. Em outro ponto dietas de baixa qualidade, porém com alta digestibilidade apresentaram um maior consumo de alimentos.

A uma extrema importância na alimentação animal para conhecimento dos mecanismos de controle da seleção e ingestão de alimentos, já que o crescimento, saúde e reprodução do animal depende da qualidade dos nutrientes ingeridos por ele (Oliveira et al., 2017).

Respostas comportamentais também conhecidas como psicogênicas são fatores que inibem ou estimulam o consumo em bovinos, fatores como odor, sabor visual, textura, aparência, o emocional do animal, podem modificar o consumo.

O mecanismo psicogênico é responsável pelas respostas do comportamento do animal em relação a fatores inibidores ou estimuladores envolvendo o ambiente ou alimentos sem ter correlação à energia ou enchimento ruminal (Nascimento et al., 2009).

As características sensoriais dos alimentos e da água tem ação sobre o estímulo do consumo, a muito se estuda a influência de sensações de frescor durante a ingestão dos alimentos, a água pode ajudar na mastigação e deglutição, atendendo também aos mecanismos de homeostáticos, atingindo a satisfação e evitando défices metabólicos energéticos.

Benedetti (2007), consta nos resultados de sua pesquisa que animais submetidos a diferentes dietas com alimentos secos e uma dieta com 70% de umidade no alimento, demonstraram diferenças na relação ingestão de água e CMS obtidos pelos coeficientes parciais de regressão para MS. Apresentando uma diferença de 55,2% na relação 1 litro de água ingerida por kg de CMS.

### **3. Materiais e métodos**

A metodologia empregada no desenvolvimento deste estudo se dá em dois momentos. O primeiro envolveu o desenvolvimento de sistema de *software* baseado na linguagem C++ que opera para leitura de dados dos animais e análises vindas de laboratório a, já o segundo implica na construção de uma interface gráfica e transformação do *software* em C++ para a *Dart in Flutter*, assim desenvolvendo um aplicável para Android.

Este aplicável para parelhos Androides busca facilitar o manejo nutricional para bovinos de corte em confinamento, possuindo um banco de dados com tabelas de composição química e de exigências nutricionais.

Para o desenvolvimento destas tabelas utilizou-se o Excel do pacote office da Microsoft, para tabelas de composição química, mas atualizadas utilizou-se de adaptação de um *software* de nutricional específico para corte que permite a visualização de diversos matérias o BR-Corte 4.0, para exigências nutricionais, utilizou-se estudos do Nutrient Requirements of Beef Cattle E-974.

### 3.1 Code::Blocks

A partir deste momento, começa a se moldar o esqueleto por assim dizer do mini conf. Utilizando o ambiente de desenvolvimento integrado de multiplataforma e código aberto *Code::Blocks* (figura 08), na linguagem C++.

O C++ foi desenvolvido durante a década de 80 com o objetivo de programar uma versão distribuída do núcleo *Unix*. Como o *Unix* era escrito em C, dever-se-ia manter a compatibilidade, ainda que adicionando novos recursos.

A IDE permite *syntax highlighting* e *code folding*, auto completar código em C++ e lista de classes integradas e uma lista de todo. Todos os arquivos abertos são organizados em *tabas*.

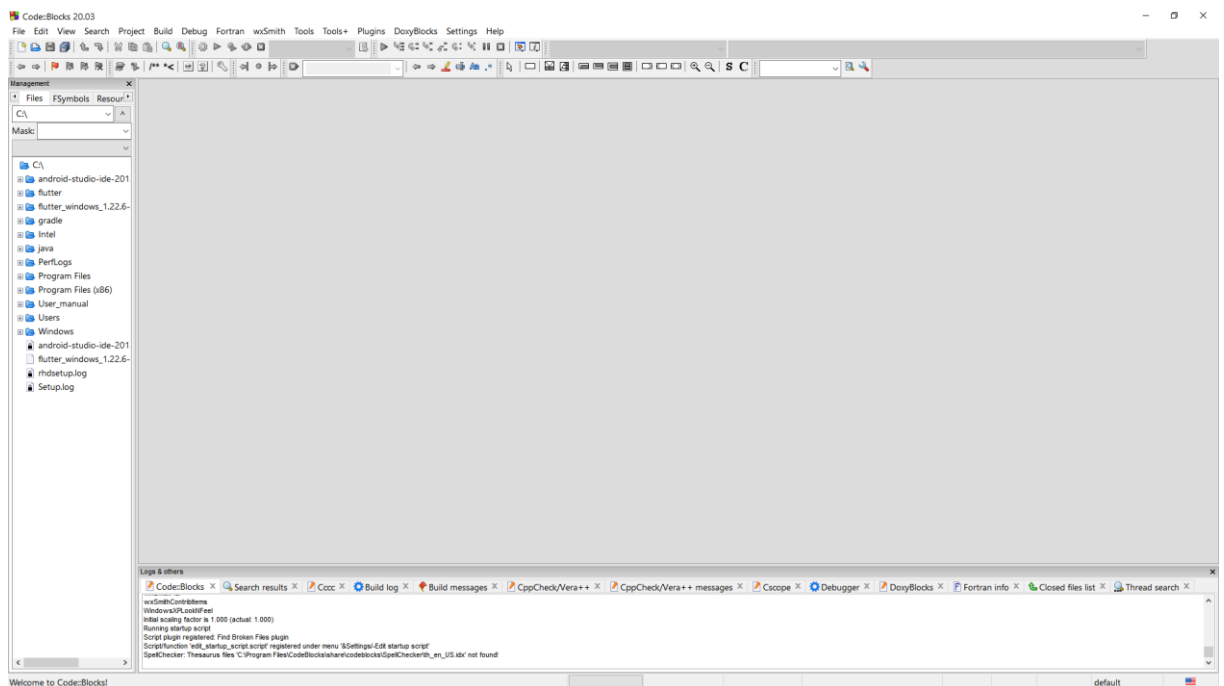


Figura 08 - Ambiente de trabalho *Code::Blocks*.

Fonte – Própria do autor.

O compilador é um programa de sistema que transcreve um código descrito em uma linguagem de alto nível para um programa equivalente em código de máquina (baixo nível) para um processador. Em geral, um compilador não produz diretamente o código de máquina, mas sim um programa em linguagem simbólica semanticamente equivalente ao programa em linguagem de alto nível (Passano, 2019).

O código de análises Mini Conf. em si, segue uma estrutura lógica é direta, com estrutura sequencial algébrica acompanhado de um modelo algébrico X e Y:

X corresponde a;

$$PBD = PBX + PBS (100 - X) + PBF \text{ onde;}$$

PBD = proteína desejada

PBX = proteína de X

PBS = proteína do segundo alimento

PBF = proteína fixada

Y a;

$$100 - X = Y$$

No método algébrico, as proporções de ingredientes para se obter uma mistura com certo teor de nutriente podem ser obtidas através do estabelecimento de equações algébricas e resolução de sistema de equações. As equações algébricas são processos simples de calcular uma mistura de alimentos. Todas as fórmulas utilizadas para os cálculos executados neste aplicativo foram extraídos de (Salman et al., 2020).

### 3.2 Estrutura do programa

A raiz do programa consiste em funções estabelecidas sequencialmente, tendo como particularidade desse fato o Mini Conf. consiste em um programa modular e estruturado.

Assim, a estrutura do Mini Conf. é:

Estrutura em C++

```
[<definições para o pré-processamento – cabeçalhos>]
[<declaração das variáveis>]
[<tipo>] main([<parâmetros>])
{
  /* Este trecho é destinado para o corpo da função, para a declaração de
suas variáveis, seus comandos e funções*/
  return [<valor>];
}
[<tipo>] funçãonome([<parâmetros>])
{...
[<declaração de parâmetros>]
[return; ou return (); ou return(valor);]
}
]
```

### 3.3 Controle dos Fluxos de Entrada e de Saída

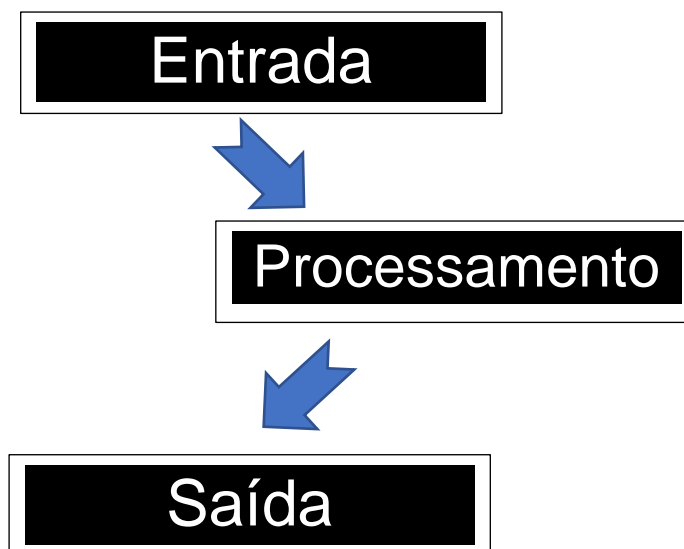


Figura 09 - Fluxograma *Code:Blocks*.

Fonte – Própria do autor.

É por meio dos fluxos de entrada e de saída (figura 09) que os operadores de acesso ao teclado e vídeo poderão ser analisados e executados. Assim sendo



será possível efetuar entrada (*cin – console input*) e saída de dados (*cout – console output*) para todas as operações realizadas no programa.

Sendo; *cin* – trabalha com o teclado, *cout* – trabalha com o monitor.

Neste momento o programa não conta com armazenamento de dados (tabelas ou dados online), as dietas aqui geradas são advindas da alimentação de dados manuais. Aqui o Mini Conf. é considerado apenas um esqueleto porem funcional.

### **3.4 Criação do mobile**

Para criação da aplicação *mobile* optou se pela linguagem *Dart* e pelo *framework Flutter*.

Criado pelo Google, o *Flutter* está na classe dos *frameworks*, ou seja, é um facilitador de desenvolvimento, que possibilita construir aplicativos para *Android*. Neste processo utilizasse de ferramentas como o *Android Studio*, *Visual Studio Code*, *Excel*, *CorelDRAW* e *Adobe Photoshop*.

### **3.5 Ambiente de trabalho**

Para se tornar mais rápido e eficiente, o desenvolvimento do ambiente utiliza se *plugins Flutter e Dart* tanto no sistema *Windows* quanto no *Android Studio* e editor de textos *Visual Studio Code* maximizando assim o desempenho (Figura 10).

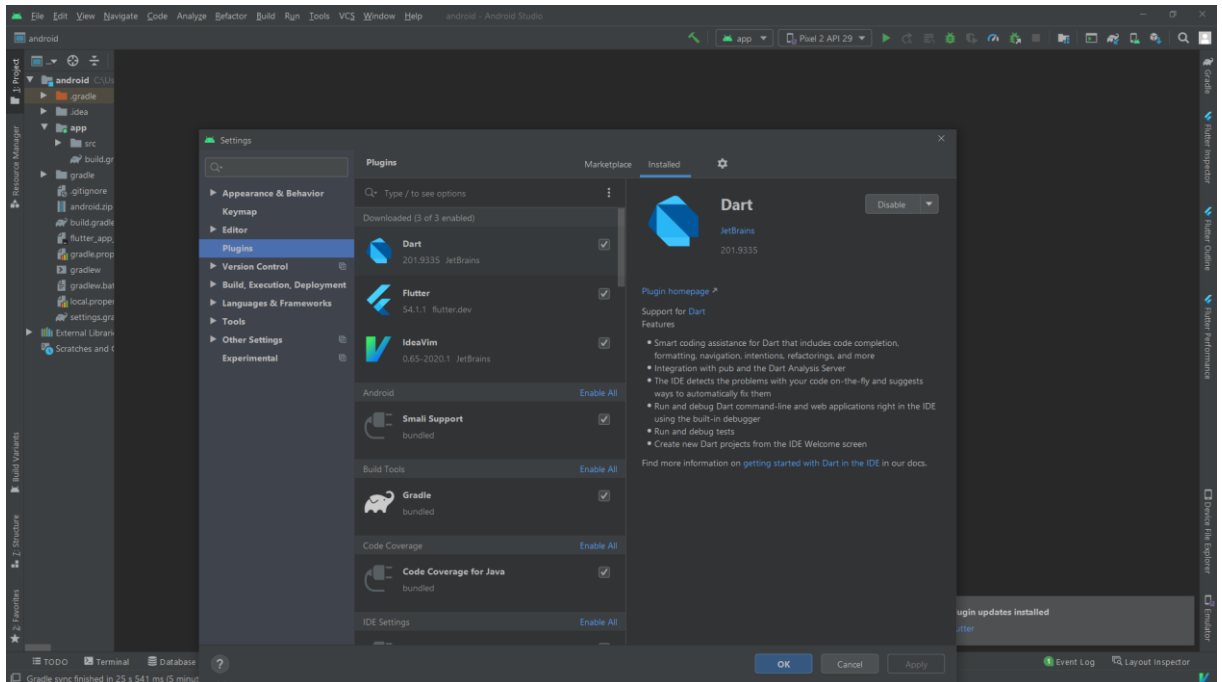


Figura 10: *Plugins Flutter e Dart no Android Studio.*

Fonte: Própria do autor.

O *Android Studio* é o ambiente de desenvolvimento integrado IDE diretamente para o desenvolvimento de *apps Android*. Além do editor de código e das ferramentas de desenvolvedor avançado, um sistema de compilação flexível baseado em *Gradle*, *emulador* rápido com inúmeros recursos, modelos de código e integração com *GitHub* para ajudar a criar recursos comuns de *apps* e importar exemplos de código, *frameworks* e ferramentas de teste (Figura 11).

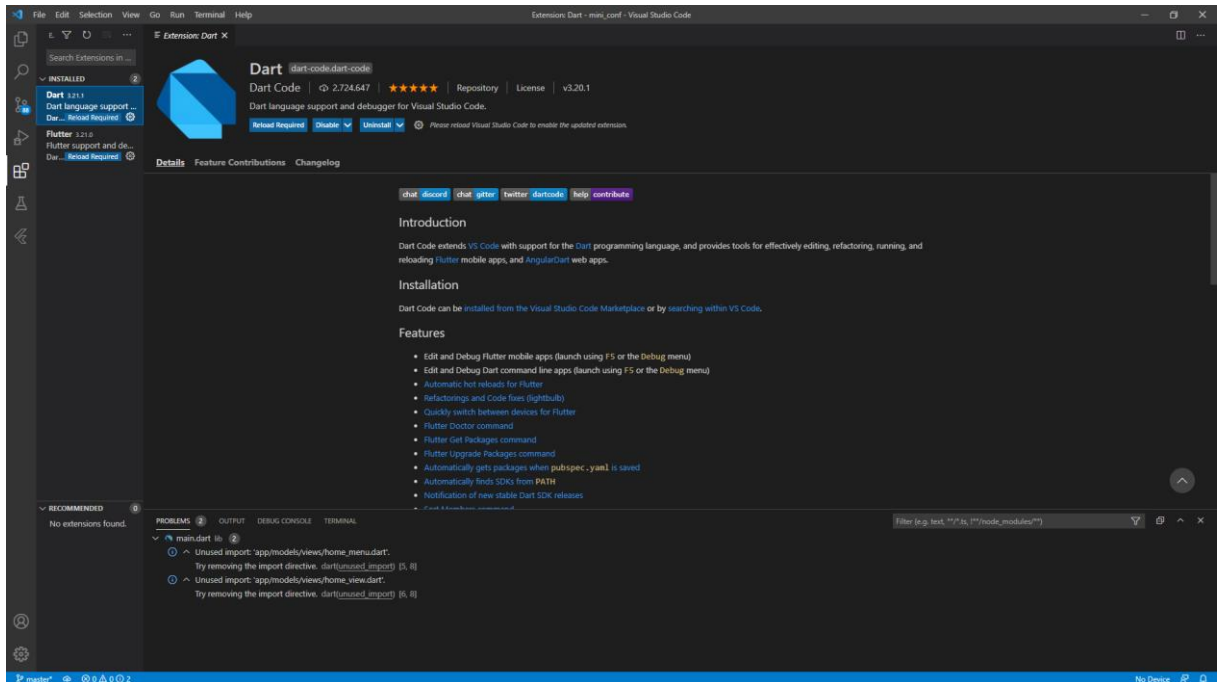


Figura 11: Plugins Flutter e Dart no Visual Studio Code.

Fonte: Própria do autor.

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft. Ele inclui suporte para depuração, controle Git, realce de sintaxe, complementação inteligente de código (Figura 12).

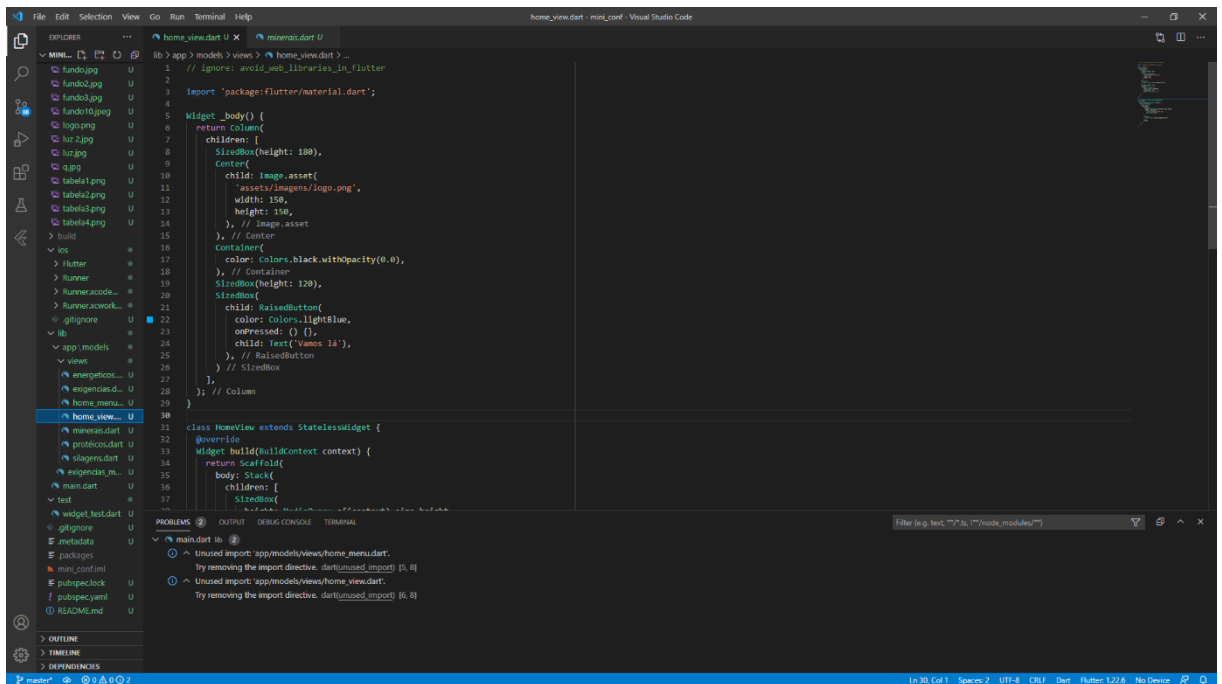


Figura 12: Ambiente de trabalho Visual Studio Code

Fonte: Própria do autor.

Para maximizar o desempenho da máquina, vinculou se o *Android Studio* com o *Visual Studio Code*, desenvolvendo assim uma área de trabalho eficiente sem exigir tanto o processador.

### 3.6 Controle dos fluxos de entrada e de saída dart em flutter

Com a transcrição do esqueleto em C++ para *Dart in flutter* o sistema operacional (figura 13) mudou drasticamente, sendo agora um aplicável *mobile* operável com armazenamento de tabelas de composição química dos alimentos, exigências nutricionais, consumo de água e vazão de água.

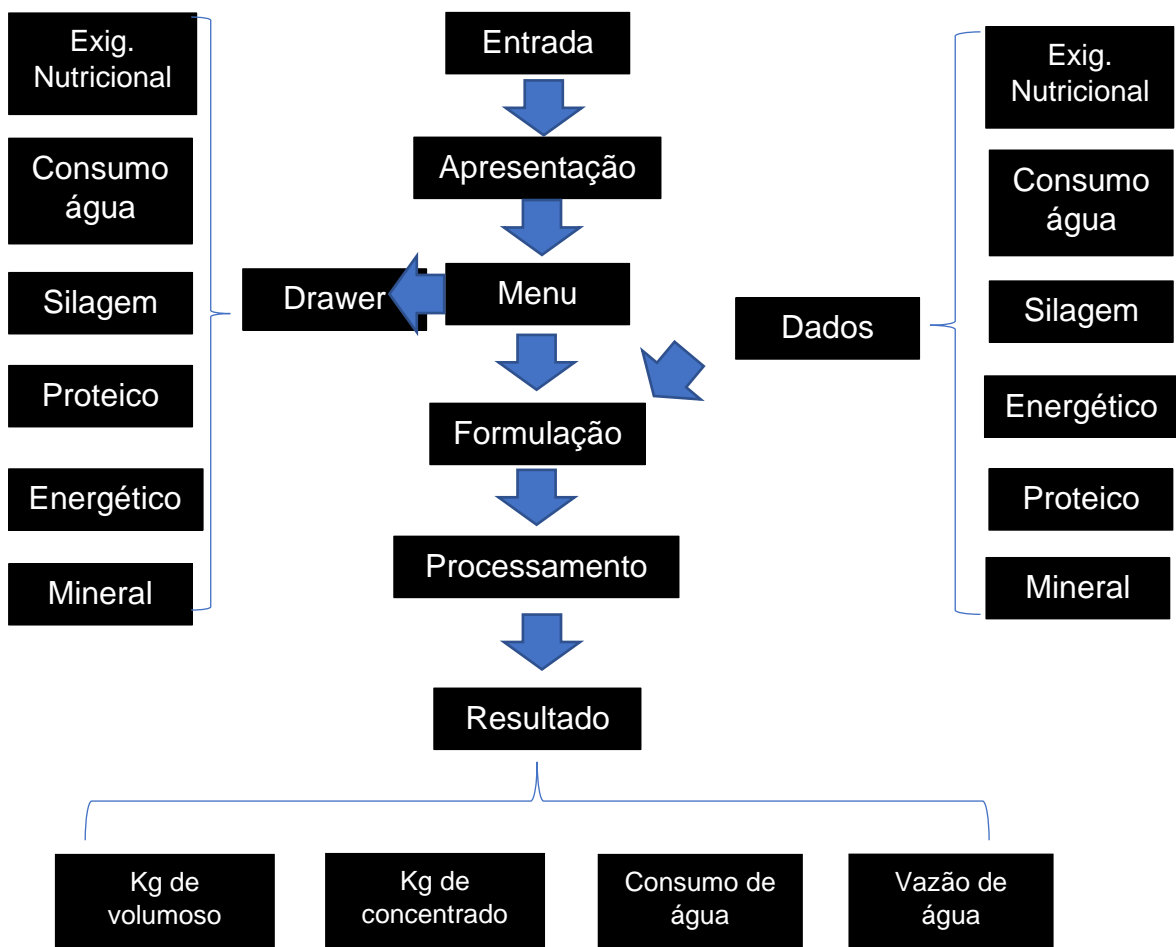


Figura 13: Fluxograma sistema Mini Conf.

Fonte: Própria do autor.

### 3.7 Tabelas

O aplicável *mobile* Mini Conf. conta com tabelas de auxílio desenvolvidas a partir de estudos científicos da plataforma BR-Corte 4.0 e estudos do *Nutrient Requirements of Beef Cattle E-974*.

As tabelas de composição química dos alimentos (Figura 20), foram adaptadas da plataforma BR-Corte 4.0 (Figura 14).

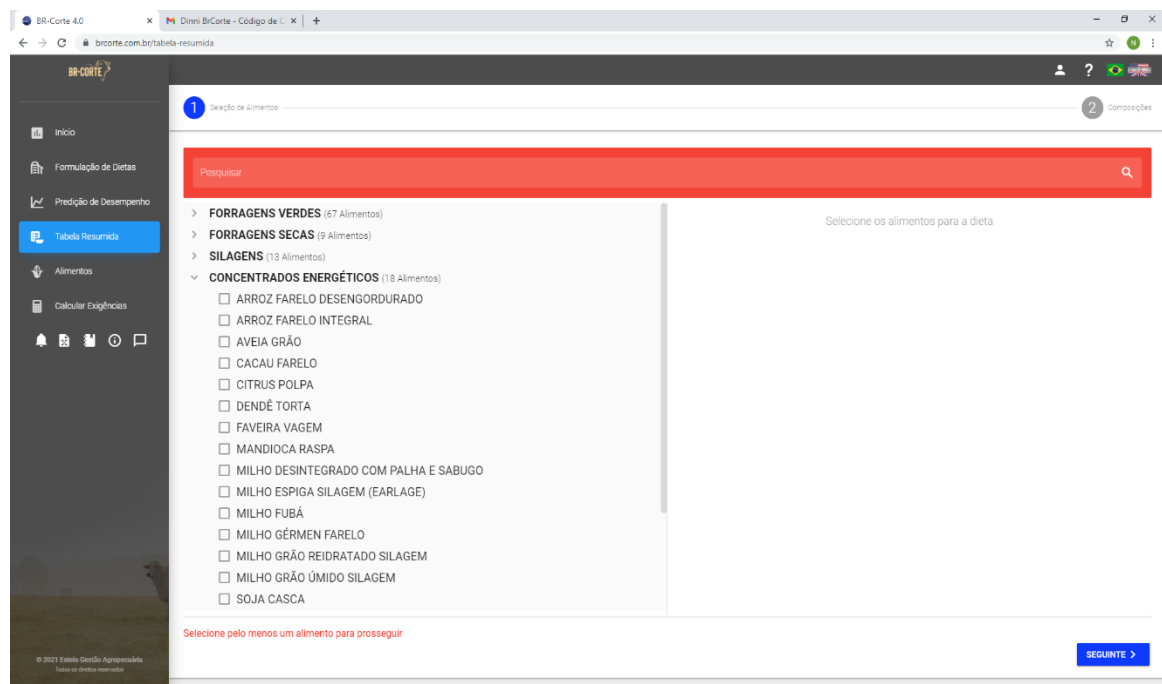


Figura 14: Plataforma BR-Corte 4.0.

Fonte: BR-Corte 4.0 (2021).

O *software* possibilita ao usuário que altere os alimentos que desejam ser trabalhados modificando assim os teores de proteína, energia e/ou minerais na formulação de dietas. A composição dos alimentos é obtida do *software* CQBAL 4.0, versão *on line* das Tabelas Brasileiras de Composição de Alimentos para Ruminantes.

Tabela Silagens					
MS%	Composição Química				
Alimentos	MS %	NDT%	PB%	Ca/g	P/g
CAPIM MOMBAÇA SILAGEM	24,41	49,43	7,36	0,44	0,12
MILHO SILAGEM	31,14	63,84	7,18	0,28	0,19
MILHO SILAGEM SEM ESPIGA	21,3	53,63	6,39	-	-
SOJA SILAGEM	25,83	60,04	17,79	-	-
SORGO FORRAGEIRO SILAGEM	28,08	63,86	6,26	0,14	0,14
SORGO SILAGEM	29,71	59,55	6,46	0,33	0,19

Tabela 01: Tabela de Silagens da composição química.

Fonte: Adaptada do BR-Corte 4.0.

As exigências nutricionais foram adaptadas dos estudos *Nutrient Requirements of Beef Cattle E-974 (2000)*, dados americanos de requerimentos nutricionais para produtividade.

Estudo norte americano que visa a construção de documentos científicos para o auxílio no entendimento das exigências nutricionais de bovinos para produção, maximizando a produtividade com maior assertividade e baixa nos custos.

#### 4. Resultados e discussões

Para construção dos resultados do app é importante que haja o correto preenchimento da tela formulação, esta tela consiste em um formulário preenchido pelo operador em que os dados são analisados pelo módulo de análises, que imprime os resultados na tela subsequente. Este formulário é dividido em classes para melhor entendimento sendo elas: Quantidade de animais e consumo de água dia; Porcentagem de núcleo; Exigências animal, peso vivo (PV), ganho de peso diário (GPV), IMS, NDT, PB, Ca e P; Relação concentrado x volumoso; Concentrado fixado, Porcentagem fixada, PB, NDT, Ca, P; 2º Concentrado, PB, NDT, Ca, P; 3º Concentrado, PB, NDT, Ca, P; Minerais Ca/gramas/kg,

P/gramas/kg; Preços, concentrado fixado, 2º concentrado, 3º concentrado ,  
minerais/ Enchimento.

Para a configuração da tela de formulário (figura 15) utilizou-se a classe STL com um *widget* criado denominado *\_body* com *return ListView* trazendo um *Image.asset* com a imagem do logotipo mini conf. espaçado com *SizedBox*, seguido por 38 *Rows* responsáveis pelos elementos textuais *Text* e *TextField* que tem função em denominar os campos de formulários impressos na tela, todos recebendo *SizedBox* em cima e em baixo para espaçamento e *Image.asset* da imagem de uma linha separando os em classes, um *RaisedButton* que ativa a análise de dados, o *blackgrand* é exibido por um *image.asset* com um *SizedBox* *height: MediaQuery.of(context).size.height* com *fit: BoxFit.cover*.

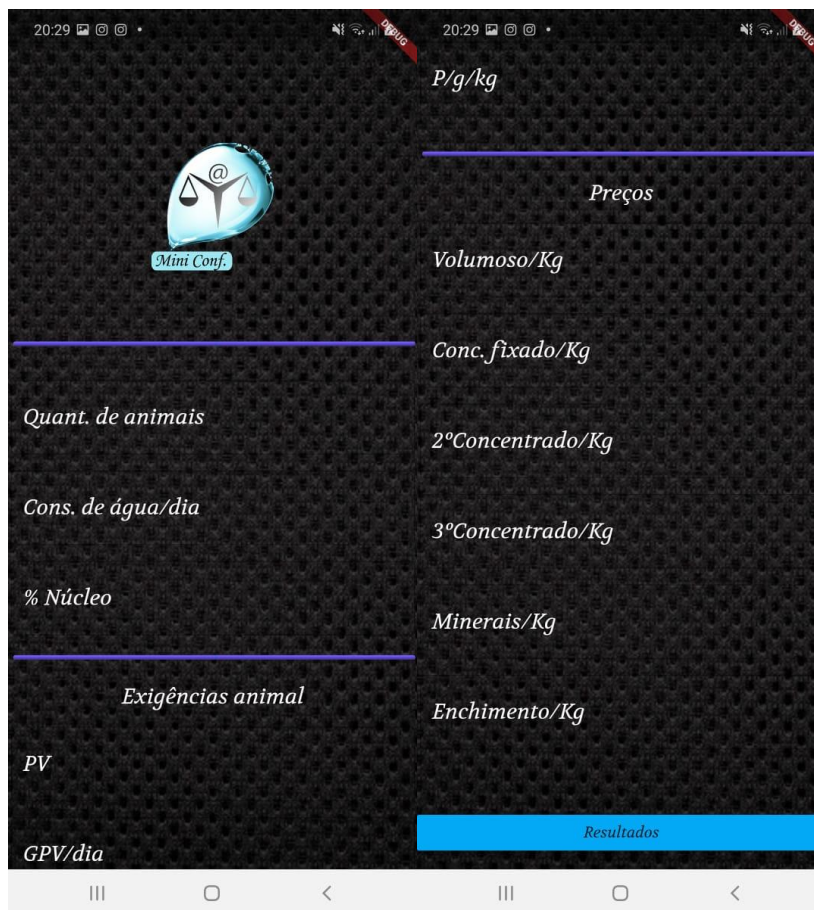


Figura 15: *Print* parte da tela de formulação.

Fonte: *Design* do próprio do autor.

A parte responsável em efetuar os cálculos e gerar as respostas é o módulo de análises onde se encontra o método *void cálculos () {}*, o que liga a coleta

de entrada de dados e saídas de respostas dos cálculos são os *controllers* criados pelo *TextEditingController* exemplo; *final TextEditingController qanimais* é o *controller* responsável por coletar a variável quantidade de animais que o operador irá fornecer.

Simplemente passar a string para o método de cálculos não funcionara, pois, o método não conseguirá ler a *string*, para que ele tenha acesso a esta informação é necessário converter a *string* para *double*. Para melhor didática a constituição exemplificada será necessária;

```
String qua = qanimais.text;
```

```
Double animais = double.tryParse (qua) ?? 1.0;
```

Descrevendo o exemplo acima nota-se que a *string qua* representa o *TextEditingController qanimais*, que por sua vez é o controlador da variável quantidade de animais que é fornecida pelo operador. A baixo sendo convertido para *double* pelo *double.tryParse*, ve-se, que durante os cálculos a *string qua* sera representada pelo *double animais*.

A seguir uma simulação dos cálculos do método *void cálculos () {}*:

Dados informados pelo operador;

Quantidade de animais: 500

Consumo de água/dia: 57

Porcentagem de núcleo: 4

Exigência animal;

PV: 350

GPV/dia: 1.2

IMS: 8.2

NDT: 5.82

PB: 0.9

Ca: 34

P: 19

Relção Concentrado x Volumoso;

Porcentagem volumoso: 60



Porcentagem concentrado: 40

Volumoso

NDT%: 65

PB%: 7.0

Ca%: 0.3

P%: 0.2

Concentrado fixado

Kg fixados: 10

MS: 88

NDT%: 63

PB%: 16

Ca%: 0.16

P%: 1.3

2º concentrado

MS%: 82

NDT%: 80

PB%: 8.0

Ca%: 0.03

P%: 0.27

3º concentrado

MS%: 88

NDT%: 75

PB%: 45

Ca%: 0.3

P%: 0.6

Minerías

Ca/g/kg: 195

P/g/Kg: 130

Preços

Volumoso/kg: 0.35

Concentrado fixado/kg: 0.90

2º Concentrado/kg: 1.2

3º Concentrado/kg: 2.83

Minerías/kg: 6.52

Enchimento/kg: 1.2

Calculo de consumo de água

Consumo de água dia \* Quantidade de animais

$$57 * 500 = 28.000 \text{ L}$$

Calculo relação volumoso x concentrado

IMS \* Porcentagem de volumoso

$$8.2 * 60\% = 4.9$$

IMS \* Porcentagem de concentrado

$$8.2 * 40\% = 3.3$$

Correção do volumoso

IMS volumoso \* NDT% volumoso

$$4.9 * 65\% = 3.2$$

IMS volumoso \* PB% volumoso

$$4.9 * 7.0\% = 0.34$$

IMS volumoso \* Ca% volumoso

$$4.9 * 0.3\% = 3.2 * 1000 = 13.8$$

IMS volumoso \* P% volumoso

$$4.9 * 0.2\% = 3.2 * 1000 = 9.8$$

Calculo do déficit

NDT exigido – NDT corrigido do volumoso

$$5.82 - 3.2 = 2.62$$

PB exigida – PB corrigida do volumoso

$$0.90 - 0.34 = 0.56$$

Ca exigido – Ca corrigida do volumoso

$$34 - 13.8 = 20.2$$

P exigido – P corrigido do volumoso

$$19 - 9.8 = 9.2$$

Calculo de Kg de concentrado

IMS concentrado \* Quantidade de animais

$$3.3 * 500 = 1650$$

Calculo de Kg de volumoso

IMS concentrado \* Quantidade de animais

$$4.9 * 500 = 2450$$

Total de kg

Kg de concentrado + Kg de volumoso

$$1650 + 2450 = 4100$$

Correção PB

PB fixada / 100

$$16 / 100 = 0.16$$

$$2^{\circ} \text{ PB} / 100 = 0.08$$

$$3^{\circ} \text{ PB} / 100 = 0.45$$

Calculo da PB

Déficit de PB \* 100 / IMS concentrado

$$0.56 * 100 / 3.3 = 17.0$$

Porcentagem para PB

100 – Porcentagem de núcleo

$$100 - 4 = 96$$

Nova PB

Calculo da PB \* 100 / Porcentagem para PB;

$$17 * 100 / 96 = 17.7$$

Nova porcentagem para formular

Porcentagem para PB – Porcentagem Kg fixados

$$96 - 10 = 86$$

Porcentagem do 2º concentrado

((Porcentagem kg fixados \* PB fixada) + (Nova porcentagem para formular \* PB 3º corrigida concentrado) – Nova PB) / (- PB corrigida 2º concentrado + PB corrigida 3º concentrado)

$$((10 * 0.16) + (86 * 0.45) - 17.7) / (- 0.08 + 0.45) = 61\%$$

Porcentagem do terceiro

Nova porcentagem – porcentagem do 2º concentrado

$$86 - 61 = 25\%$$

Calculo NDT

Porcentagem kg fixados \* IMS concentrado

$$10\% * 3.3 = 0.3 * \text{MS concentrado fixado}$$

$$0.3 * 88 = 0.2$$

Porcentagem do 2º concentrado \* IMS concentrado

$$61\% * 3.3 = 2.0 * \text{MS 2º concentrado}$$

$$2.0 * 1.64$$

Porcentagem do 3º concentrado \* IMS concentrado

$$25\% * 3.3 = 0.82 * \text{MS 3º concentrado}$$

$$0.82 * 88\% = 0.72$$

Calculo minerais concentrado fixado

$$0.3 * 0.16 * 10 = 0.48$$

$$0.3 * 1.3 * 10 = 3.9$$

Calculo minerais 2º concentrado

$$2.0 * 0.03 * 10 = 0.6$$

$$2.0 * 0.27 * 10 = 5.4$$

Calculo minerais 3º concentrado

$$0.82 * 0.3 * 10 = 2.46$$

$$0.82 * 0.6 * 10 = 4.92$$

Total Ca e P

$$\text{Ca } 0.48 + 0.6 + 2.46 = 3.54$$

$$\text{P } 3.9 + 5.4 + 4.92 = 14.22$$

Falta de Ca e P

$$\text{Ca } 20.2 - 3.54 = 16.66$$

$$\text{P } 9.2 - 14.22 = - 5.02$$

Todas as vezes que a Falta de Ca for < que o Ca exigido o método realizara o cálculo a seguir;

Calculo do núcleo

$$195 * 100 / 1000 = 19.5$$

$$130 * 100 / 1000 = 13$$

$$100 * 16.6 / 19.5 = 85.12$$

$$85.5 * 13 / 100 = 11.06$$

$$85.12 / 1000 / 3.3 * 100 = 2.57$$

$$4 - 2,57 = 1.43$$

Total de nutrientes da dieta

$$\text{NDT } 3.2 + 0.29 + 1.65 + 0.73 = 5,87$$

$$\text{PB } 17.7$$

$$\text{Ca } 13.8 + 3.63 + 16.2 = 33.63$$

$$\text{P } 9.8 + 14.7 + 11.06 = 35.56$$

$$\text{Kg fixado} = 1650 * 10 / 100 = 165$$

$$\text{kg } 2^\circ \text{ concentrado} = 1650 * 61 / 100 = 1006.5$$

$$\text{kg } 3^\circ \text{ concentrado} = 1650 * 25 / 100 = 412.5$$

$$\text{kg Minerais} = 1650 * 2.57 / 100 = 40.09$$

$$\text{kg Enchimentos} = 1650 * 1.43 / 100 = 23.59$$

Porcentagem da dieta

$$\text{Volumoso} = 2450 * 100 / 4100 = 60$$

$$\text{fixado} = 165 * 100 / 4100 = 4.02$$

$$2^{\circ} \text{ concentrado} = 1006.5 * 100 / 4100 = 24.54$$

$$3^{\circ} \text{ concentrado} = 412.5 * 100 / 4100 = 10.06$$

$$\text{Minerais} = 40.09 * 100 / 4100 = 0.97$$

$$\text{Enchimentos} = 23.59 * 100 / 4100 = 0.57$$

Calculo de custo da dieta

$$\text{Kg volumoso} 2450 * 0.35 = 857.50$$

$$\text{kg fixado} 165 * 0.9 = 148.00$$

$$\text{kg } 2^{\circ} \text{ concentrado} 1006.5 * 1.2 = 1207.80$$

$$\text{kg } 3^{\circ} \text{ concentrado} 412.5 * 2.83 = 1167.37$$

$$\text{kg Minerais} 40.09 * 6.52 = 276.44$$

$$\text{kg Enchimentos} 23.59 * 1.2 = 28.32$$

$$\text{Total} = 3685.93$$

Assim como é necessário transformar *string* em *double*, para que os resultados sejam impressos na tela resultados (figura 16) recuperar o valor *double* do método de *void cálculos* para ser impresso na tela precisa ser convertido para *string* não funcionara, pois, o método não conseguirá ler a *string*. Para melhor didática a constituição exemplificada será necessária;

*Reqanimais = animais.toStringAsFixed;*

No exemplo acima, a *string reqanimais* passa a representar o valor da *double animais* que é convertida por um *toStringAsFixed*.

A tela resultados é impressa no aparelho a partir das análises dos dados alimentados pelo usuário, o sistema exibira resultados para PV, GPV, IMS volumoso, IMS concentrado, porcentagem de volumoso, porcentagem de concentrado, quantidade de animais, consumo de água, a porcentagem da composição da ração dia, porcentagem de volumoso, porcentagem concentrado fixado, porcentagem segundo concentrado, porcentagem terceiro concentrado,

porcentagem minerais, porcentagem enchimento, NDT%, PB%, Ca/g e P/g, total IMS, kg volumoso, kg concentrado fixado, kg segundo concentrado, kg terceiro concentrado, kg minerais, kg enchimento, kg total, custo volumoso, custo concentrado fixado, custo segundo concentrado, custo terceiro concentrado custo minerais, custo enchimento.

Para a configuração desta tela utilizou-se a classe STL com um *widget* criado denominado *\_bode* com *return ListView* trazendo um *Image.asset* com a imagem do logotipo mini conf. espaçado com *SizedBox*, seguido por 16 *Rows* responsáveis pelos elementos textuais *Text* e *TextField* que tem função em denominar os resultados impressos na tela, todos recebendo *SizedBox* em cima e em baixo para espaçamento e *Image.asset* da imagem de uma linha separando os em classes, um *RaisedButton* que reinicia uma nova formulação, o *blackgrand* é exibido por um *image.asset* com um *SizedBox* *height: MediaQuery.of(context).size.height* com *fit: BoxFit.cover*.

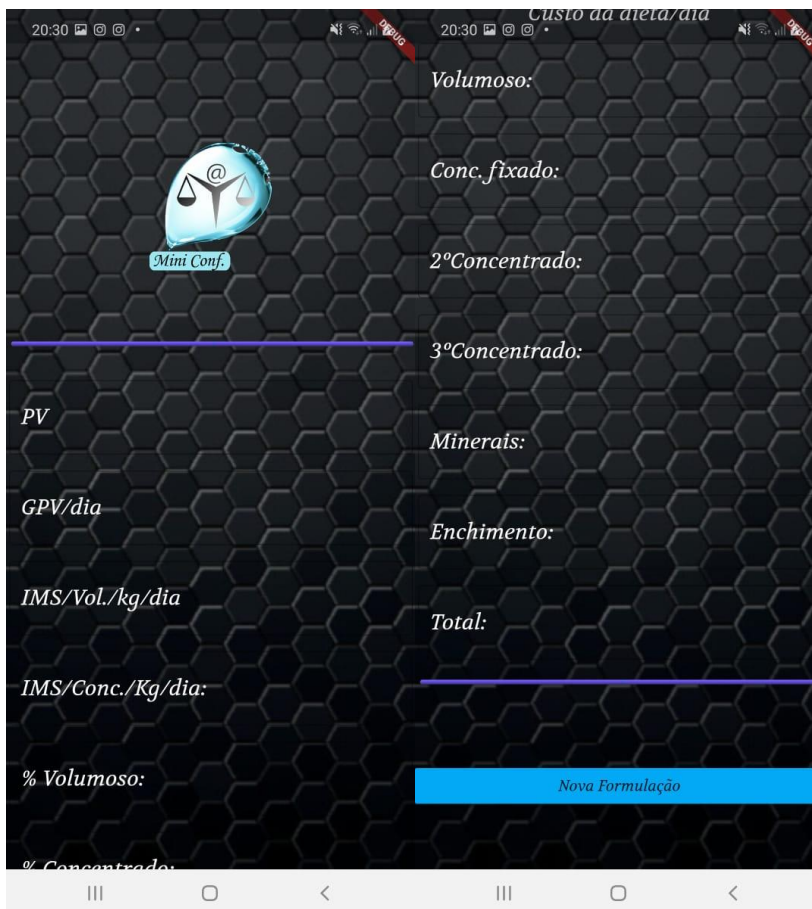


Figura 16: *Print* tela de resultados.

Fonte: *Design* do próprio do autor.

#### 4. CONSIDERAÇÕES FINAIS

A crescente demanda tecnológica, impulsiona a busca por ferramentas cada vez mais viáveis. O *dart in flutter* possibilita a criação de projetos capazes de serem utilizados tanto em androids podendo ser criados versões para IOS abrindo assim possibilidades de mercado.

O Mni Conf. sendo um aplicativo para aparelhos androids, traz características notáveis possuindo seu próprio banco de dados e sendo capaz de estimar o consumo de água, balancear dietas oferecendo as porcentagens dos alimentos e nutrientes kgs e custos diários da dieta, sendo eficaz e atendendo as características de mercado.

Espera-se que este aplicativo possa facilitar o manejo de alimentos para animais confinados, buscando alternativas que viabilizem e tragam parâmetros nutricionais e econômicos.



## 5. REFERÊNCIAS BIBLIOGRAFICAS

ALMEIDA, R. Consumo e eficiência alimentar de bovinos em crescimento. (Tese). Piracicaba: Universidade de São Paulo Escola Superior de Agricultura “Luiz de Queiroz”;2005.

ALVES, S. K. L.; VIANA, P. G.; RAINERI, C.; Utilização de ferramentas digitais na pecuária e extensão rural pubvet [online]. 2019; 13; p.1-9 [acessado em 16 de abril de 2021].  
<https://www.pubvet.com.br/uploads/38742c625f80cb897eaf5ac5da3182eb.pdf>.

BENEDETTI, E. D. Água na Nutrição de Ruminantes (Dissertação). Uberaba: Faculdades Associadas de Uberaba – FAZU; 2007.

BENEZ e CABRAL Terminação de bovinos de corte. Produção e manejo de bovinos de corte in: Kcm; 2015. cap.6, p. 86-109.

BONFIM, P. M.; MARQUES, C. I. Sistema de informações gerenciais voltado a melhoria de desempenho do confinamento bovino: um estudo de caso; 56 Congresso SOBER; 2018, Campinas, SP, Brasil. Campinas: 2018. Asseado em 17 de abril de 2021.

CARVALHO, C. G. Zootecnia de precisão e ferramentas de gestão aplicáveis na bovinocultura. (Monografia). Boa Vista: Universidade Federal de Roraima; 2018.

CLARINDO, L. R. Fontes energéticas e proteicas para bovinos confinados em fase de terminação. (Dissertação). Piracicaba: Universidade de São Paulo Escola Superior de Agricultura “Luis de Queiroz”.2006.

DONATTI, N. R. Desenvolvimento de um sistema de monitoramento de animais, utilizando rede de sensores sem fio, baseado no protocolo ZigBee e tecnologia GPS. (Dissertação). Pirassununga: Universidade de São Paulo Faculdade de Zootecnia e Engenharia de Alimentos; 2017.

DUARTE, P. D. G. Introdução à matemática e à administração financeira intervalar. (Monografia). Natal,RN: Univercidade Federal do Rio Grande do Norte; 2007.

SILVEIRA, O; A gota d'água; São Paulo; 2014.Thiago Tanji [acessado 26 de abril de 2021]. Disponível em <https://revistagalileu.globo.com/Revista/noticia/2014/11/gota-dagua.html>.

FERREIRA, F. S.; NETO, D. M.; PEREIRA, R. L. M.; MELO, F. H. A.; OLIVEIRA, G. L.; NETO, N. T. J. Fatores que afetam o consumo alimentar de bovinos, Arquivos de Pesquisa Animal [Online] 2013, V.2: p. 9-19 [acessado em 27 de abril de 2021].

FERREIRA, S. M. A. Consumo observado e predito pelos sistemas nutricionais em bovinos de corte confinados. (Dissertações). Uberlândia: Universidade Federal Uberlândia Faculdade de Medicina Veterinária, 2019.

FILHO, V. C. S.; SILVA, C. F. L.; GIONBELLI, P. M.; ROTTA, P. P.; MARCONDES, I. M.; CHIZZOTTI, L. M.; PRADOS, F. L. Regulação e predição de consumo de matéria seca. Exigências Nutricionais de Zebuínos Puros e Cruzados BR-CORTE In: 3 ed. Viçosa: Suprema Gráfica; 2016. cap.2, p.15-44.

FILHO, V. C. S.; SILVA, C. F. L.; GIONBELLI, P. M.; ROTTA, P. P.; MARCONDES, I. M.; CHIZZOTTI, L. M.; PRADOS, F. L. Exigências de minerais para bovinos de corte. Exigências Nutricionais de Zebuínos Puros e Cruzados BR-CORTE In: 3 ed. Viçosa: Suprema Gráfica; 2016. cap.2, p.15-44.

NASCIMENTO, L. M. P.; FARJALLA, B. Y.; NASCIMENTO, L. J. Consumo voluntário de bovinos - Bovines voluntary intake. **REDVET Rev. electrón. Vet.** São Paulo, Vol. 10, Nº 10, 2009.

MATOS, P. L. M. Gestão na bovinocultura de corte (Monografia). Rondonópolis. Universidade Federal de Rondonópolis; 2020.

MEDEIROS, R. S.; GOMES, C. R.; BUNGENSTAB, J. D. Nutrição de bovinos de corte Fundamentos e aplicações. [online]. Brasília,DF. Embrapa Gado de Corte. 2015.[ acessado em 05 de maio 2021] Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/120040/1/Nutricao-Animal-livro-em-baixa.pdf>.

OLIVEIRA, C. B.; CAETANO, O. A. G.; JUNIOR, C. B. M.; MARTINS, R. T.; OLIVEIRA, B. C. Mecanismos reguladores de consumo em bovinos de corte. **Nutritime Revista Eletrônica**. Goiânia, Vol. 14, Nº 04, 2017.

PANOSSO, R. A. Linguagem de Programação C++ (Dissertação). São Paulo. Universidade Estadual Paulista “Júlio de Mesquita Filho”.2019.

PEREIRA, R. E.; PATERNIANI, S. E. J.; DEMARCHI, A. A. J. J. A importância da qualidade da água de dessedentação animal. **Revista Brasileira de Engenharia de Biosistema**. 2009. v.3 n.3, p.227-235, [acessado em 3 de maio 2021]. [https://www.researchgate.net/publication/287810325\\_A\\_IMPORTANCIA\\_DA\\_QUALIDADE\\_DA\\_AGUA\\_DE\\_DESSEDENTACAO\\_ANIMAL\\_THE\\_IMPORTANCE\\_OF\\_WATER\\_QUALITY\\_OF\\_ANIMAL\\_WATERING/link/571e140108aead26e71a7ead/download](https://www.researchgate.net/publication/287810325_A_IMPORTANCIA_DA_QUALIDADE_DA_AGUA_DE_DESSEDENTACAO_ANIMAL_THE_IMPORTANCE_OF_WATER_QUALITY_OF_ANIMAL_WATERING/link/571e140108aead26e71a7ead/download).

RODRIGUES, M. O. D. A importância da água na produção de bovinos de corte. (Monografia). Cuiabá: Universidade Federal de Mato Grosso, 2018.

SALMAN, K. A.; OSMARI, K. E.; NETO, F. A. J.; SANTOS, R. G. M.; MATARAZZO, V. S; Manual prático de formulação de ração para vacas leiteiras [online] 2020: p.26 [acessado em 14/04/2021] <https://www.infoteca.cnptia.embrapa.br/infoteca/bitstream/doc/1123902/1/cpafro-18428-doc167.pdf>.

SILVA, F. R.; AGUIAR, O. M. Linguagem C: Introdução ao C em 10 aulas [Online], São Paulo, Editora Alegre, 2016. [acessado em 28 de abril de 2021]. Disponível em

[https://repositorio.ufes.br/bitstream/10/6800/1/Introdu%C3%A7%C3%A3o\\_C\\_10\\_Aulas.pdf](https://repositorio.ufes.br/bitstream/10/6800/1/Introdu%C3%A7%C3%A3o_C_10_Aulas.pdf).

## Anexos

Anexo 01: Tabela de composições químicas de concentrados energéticos.

<b>Tabela Concentrados Energéticos</b>					
<b>MS%</b>	<b>Composição Química</b>				
<b>Alimentos</b>	<b>MS %</b>	<b>NDT%</b>	<b>PB%</b>	<b>Ca/g</b>	<b>P/g</b>
<b>MILHO DESINTEGRADO</b>	87,89	75,88	7,06	0,04	0,22
<b>MILHO FUBÁ</b>	87,96	87,68	9,01	0,03	0,26
<b>MILHO GÉRMEN FARELO</b>	89,91	32,48	11,02	0,03	0,42
<b>SOJA CASCA</b>	90,1	72,5	12,59	0,52	0,16
<b>SORGO GRÃO</b>	88	84,37	9,33	0,07	0,29
<b>TRIGO FARELO</b>	87,65	77,28	16,66	0,17	1,01
<b>MILHO GRÃO REIDRA. SILAGEM</b>	65,76	99,56	9,31	-	-
<b>MILHO GRÃO ÚMIDO SILAGEM</b>	66,72	88,23	9,19	0,03	0,25
<b>SORGO GRÃO REIDRA. SILAGEM</b>	65,27	79,9	9,14	-	-

Fonte: Adaptada do BR-Corte 4.0.

Anexo 02: Tabela de composições químicas de minerais.

<b>Tabela Minerais</b>					
<b>MS%</b>	<b>Composição Química</b>				
<b>Alimentos</b>	<b>MS %</b>	<b>NDT%</b>	<b>PB%</b>	<b>Ca/g</b>	<b>P/g</b>
<b>CALCÁRIO</b>	99,22	-	-	37	0,02
<b>FOSFATO BICÁLCICO</b>	98,44	-	-	24,12	18,53
<b>ÓXIDO DE MAGNÉSIO</b>	98,32	-	-	0,58	-
<b>SAL COMUM</b>	98,97	-	-	-	0,3

Fonte: Adaptada do BR-Corte 4.0.

Anexo 03: Tabela de composições químicas de concentrados proteicos.

<b>Tabela Concentrados Protéicos</b>					
<b>MS%</b>	<b>Composição Química</b>				
<b>Alimentos</b>	<b>MS %</b>	<b>NDT%</b>	<b>PB%</b>	<b>Ca/g</b>	<b>P/g</b>
<b>ALGODÃO CAROÇO</b>	90,64	46,04	23,14	0,27	0,75
<b>ALGODÃO TORTA</b>	90,41	75,6	29,64	0,28	0,58
<b>GIRASSOL FARELO</b>	90,22	71,05	31,47	0,3	0,9
<b>MILHO DESINT. PALHA E SABUGO</b>	87,89	75,88	7,06	0,04	0,22
<b>MILHO FUBÁ</b>	87,96	87,68	9	0,03	0,26
<b>MILHO GÉRMEN FARELO</b>	89,91	90	11	0,03	0,42
<b>RESÍDUO DE CERVEJARIA ÚMIDO</b>	22,28	92,16	25,59	0,33	0,78
<b>DDG</b>	87,5	86,23	23,55	0,05	0,32
<b>WDG</b>	31,8	48,17	32	0,05	0,35
<b>SOJA CASCA</b>	90,1	72,5	12,59	0,52	0,16
<b>SOJA FARELO</b>	88,64	78,59	34,9	0,34	0,59
<b>SOJA FARELO EXTRUSADO</b>	96,73	89,74	40,98	0,07	0,57
<b>SOJA GRÃO</b>	90,92	100	38,47	0,32	0,53
<b>SOJA GRÃO TOSTADA</b>	91,7	20,77	39,1	0,25	0,49
<b>SORGO GRÃO</b>	88	84,37	9,33	0,07	0,29
<b>TRIGO FARELO</b>	87,65	77,28	16,66	0,17	1

Fonte: Adaptada do BR-Corte 4.0.

Anexo 04: Tabela de composições químicas de concentrados proteicos.

Estimativa Consumo de Água	
Bovino kg	L/Dia
270	36
370	57
455	78

Fonte: Pereira et al, (2009).

Anexo 05: *Scripts* do arquivo *Main.dart*.

```
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:mini_conf/app/models/views/agua.dart';
import 'package:mini_conf/app/models/views/energeticos.dart';
import 'package:mini_conf/app/models/views/exigencia.dart';
import 'package:mini_conf/app/models/views/formular.dart';
import 'package:mini_conf/app/models/views/minerais.dart';
import 'package:mini_conf/app/models/views/prot%C3%A9icos.dart';
import 'package:mini_conf/app/models/views/silagens.dart';

import 'app/models/views/home_menu.dart';
import 'app/models/views/home_view.dart';

main() {
  runApp(AppWidget());
}

class AppWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData.light(),
      initialRoute: '/',
      routes: {
        '/': (context) => HomeView(),
        '/Menu': (context) => HomeMenu(),
        '/Exigencia': (context) => Exigencias(),
        '/Agua': (context) => ConsumoAgua(),
      }
    );
  }
}
```

```

    '/Silagem': (context) => Silagens(),
    '/Proteico': (context) => Proteicos(),
    '/Energetico': (context) => Energeticos(),
    '/Mineral': (context) => Minerais(),
    '/Formular': (context) => Formular(),
  },
);
}
}
}

```

Fonte: Desenvolvimento próprio do autor.

Anexo 06: *Scripts* do arquivo *Home View*.

```

// ignore: avoid_web_libraries_in_flutter

// ignore: avoid_web_libraries_in_flutter
import 'dart:html';
// ignore: avoid_web_libraries_in_flutter

import 'package:flutter/material.dart';

Widget _body() {
  return Column(
    children: [
      SizedBox(height: 180),
      Center(
        child: Image.asset(
          'assets/imagens/logo.png',
          width: 150,
          height: 150,
        ),
      ),
      Container(
        color: Colors.black.withOpacity(0.0),
      ),
      SizedBox(height: 120),
      SizedBox(
        child: RaisedButton(
          onPressed: () {},
          child: Text('Vamos lá'),
        ),
      ),
    ],
  );
}

class HomeView extends StatelessWidget {
  @override
  Widget build(BuildContext context) {

```



```

return Scaffold(
  body: Stack(
    children: [
      SizedBox(
        height: MediaQuery.of(context).size.height,
        child: Image.asset(
          'assets/imagens/fundo2.jpg',
          fit: BoxFit.cover,
        ),
      ),
      Container(
        color: Colors.black.withOpacity(0.4),
      ),
      _body()
    ],
  ),
);
}

```

Fonte: Desenvolvimento próprio do autor.

Anexo 07: *Scripts* do arquivo *Home menu*.

```

import 'package:flutter/material.dart';

class HomeMenu extends StatefulWidget {
  @override
  _HomeMenuState createState() => _HomeMenuState();
}

Widget _ted() {
  return Center(
    child: Column(
      children: [
        SizedBox(
          height: 170,
        ),
        Text(
          'Bem vindo ao Mini Conf.',
          style: TextStyle(
            fontFamily: 'Volkhov', fontSize: 30, color: Colors.white),
        ),
        SizedBox(
          height: 110,
        ),
        RaisedButton(
          color: Colors.lightBlue,
          onPressed: () {},
          child: Text('Formular',

```

```

        style: TextStyle(
          fontFamily: 'Volkhov',
        )),
      RaisedButton(
        color: Colors.lightBlue,
        onPressed: () {},
        child: Text('Dicas!',
          style: TextStyle(
            fontFamily: 'Volkhov',
          )),
      ),
      SizedBox(
        height: 40,
      ),
      Center(
        child: Text('(62) 9 96328417',
          style: TextStyle(
            fontFamily: 'Volkhov', fontSize: 15, color: Colors.white))
      ),
    ),
    Text('nutriminiconf@gmail.com',
      style: TextStyle(
        fontFamily: 'Volkhov', fontSize: 15, color: Colors.white))
  ],
),
);
}

Widget _doc() {
  return Column(
    children: <Widget>[
      UserAccountsDrawerHeader(
        decoration: BoxDecoration(
          color: Colors.lightBlue,
          image: new DecorationImage(
            image: AssetImage('assets/imagens/luz.jpg'),
            fit: BoxFit.cover,
          )),
        currentAccountPicture: Image.asset('assets/imagens/logo.png'),
        accountName: Text(
          'Olá este é nosso menu',
          style: TextStyle(fontFamily: 'Volkhov', color: Colors.white),
        ),
        accountEmail: Text(
          'Fique a vontade!',
          style: TextStyle(fontFamily: 'Volkhov', color: Colors.white),
        ),
      ),
      ListTile(
        leading: Icon(

```

```

        Icons.account_balance_outlined,
        color: Colors.white,
    ),
    title: Text(
      'Exigências',
      style: TextStyle(fontFamily: 'Volkhov', color: Colors.white),
    ),
  ListTile(
    leading: Icon(
      Icons.grain_sharp,
      color: Colors.white,
    ),
    title: Text(
      'Consumo de água',
      style: TextStyle(fontFamily: 'Volkhov', color: Colors.white),
    ),
  ),
  ListTile(
    leading: Icon(
      Icons.grass_outlined,
      color: Colors.white,
    ),
    title: Text(
      'Silagens',
      style: TextStyle(fontFamily: 'Volkhov', color: Colors.white),
    ),
  ),
  ListTile(
    leading: Icon(
      Icons.battery_charging_full,
      color: Colors.white,
    ),
    title: Text(
      'Concentrado Energético',
      style: TextStyle(fontFamily: 'Volkhov', color: Colors.white),
    ),
  ),
  ListTile(
    leading: Icon(
      Icons.add_link,
      color: Colors.white,
    ),
    title: Text(
      'Concentrado Protéico',
      style: TextStyle(fontFamily: 'Volkhov', color: Colors.white),
    ),
  ),
  ListTile(
    leading: Icon(
      Icons.workspaces_outline,
      color: Colors.white,
    ),
  ),

```

```

        title: Text(
          'Minerais',
          style: TextStyle(fontFamily: 'Volkhov', color: Colors.white),
        )),
      ],
    );
  }

class _HomeMenuState extends State<HomeMenu> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      drawer: Drawer(
        child: Stack(children: [
          SizedBox(
            height: MediaQuery.of(context).size.height,
            child: Image.asset(
              'assets/imagens/luz 2.jpg',
              fit: BoxFit.cover,
            ),
          ),
          _doc()
        ])),
      appBar: AppBar(
        backgroundColor: Colors.lightBlue,
        title: Center(
          child: Text(
            'Mini Conf.',
            style: TextStyle(fontFamily: 'Volkhov', color: Colors.white),
          )),
      ),
      body: Stack(
        children: [
          SizedBox(
            height: MediaQuery.of(context).size.height,
            child: Image.asset(
              'assets/imagens/fundo3.jpg',
              fit: BoxFit.cover,
            ),
          ),
          Container(
            color: Colors.black.withOpacity(0.4),
          ),
          _ted()
        ],
      ),
    );
  }
}

```

```
}

```

Fonte: Desenvolvimento próprio do autor.

Anexo 08: *Scripts* do arquivo estimativas de consumo de água.

```
import 'package:flutter/material.dart';

Widget _sil() {
  return Container(
    child: Center(
      child: (Image.asset(
        'assets/imagens/agua.png',
      )),
    ),
  );
}

class ConsumoAgua extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.lightBlue,
        title: Center(
          child: Text(
            'Mini Conf.',
            style: TextStyle(fontFamily: 'Volkhov', color: Colors.white),
          ),
        ),
      ),
      body: Stack(
        children: [
          SizedBox(
            height: MediaQuery.of(context).size.height,
            child: Image.asset(
              'assets/imagens/cromo.jpg',
              fit: BoxFit.cover,
            ),
          ),
          _sil(),
        ],
      ),
    );
  }
}

```

Fonte: Desenvolvimento próprio do autor.

Anexo 10: *Scripts* do arquivo formular.

```
import 'package:flutter/material.dart';

Widget _sil() {

```

```

return Container(
  child: Center(child: Image.asset('assets/imagens/tabela1.png')),
);
}

class Silagens extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.lightBlue,
        title: Center(
          child: Text(
            'Mini Conf.',
            style: TextStyle(fontFamily: 'Volkhov', color: Colors.white),
          ),
        ),
      ),
      body: Stack(
        children: [
          SizedBox(
            height: MediaQuery.of(context).size.height,
            child: Image.asset(
              'assets/imagens/cromo.jpg',
              fit: BoxFit.cover,
            ),
          ),
          Container(
            color: Colors.black.withOpacity(0.4),
          ),
          _sil(),
        ],
      ),
    );
  }
}

```

Fonte: Desenvolvimento próprio do autor.

Anexo 10: *Scripts* do arquivo formulário.

```

import 'package:flutter/material.dart';

Widget _body() {
  return ListView(
    children: [
      SizedBox(height: 100),
      Center(
        child: Image.asset(
          'assets/imagens/logo.png',

```

```

        width: 130,
        height: 130,
      ),
    ),
    SizedBox(height: 50),
    Image.asset('assets/imagens/linha.png'),
    Row(
      children: [
        Expanded(
          child: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 3),
            child: Text('Quant. de animais:',
              style: TextStyle(
                fontFamily: 'Volkhov',
                fontSize: 20,
                color: Colors.white))),
        Expanded(
          child: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 1),
            child: SizedBox(
              height: 30,
              child: TextField(
                decoration: InputDecoration(
                  enabledBorder: UnderlineInputBorder(
                    borderSide: BorderSide(color: Colors.white))),
              ),
            ),
          ),
        ),
      ],
    ),
    SizedBox(height: 5),
    Row(
      children: [
        Expanded(
          child: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 3),
            child: Text('Cons. água/dia:',
              style: TextStyle(
                fontFamily: 'Volkhov',
                fontSize: 20,
                color: Colors.white))),
        Expanded(
          child: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 1),
            child: SizedBox(
              height: 30,
              child: TextField(

```

```

        decoration: InputDecoration(
          enabledBorder: UnderlineInputBorder(
            borderSide: BorderSide(color: Colors.white))),
      ),
    ),
  ),
),
],
),
Image.asset('assets/imagens/linha.png'),
SizedBox(height: 5),
Center(
  child: Text('Exigências animal',
    style: TextStyle(
      fontFamily: 'Volkhov', fontSize: 20, color: Colors.white)),
),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('PV:',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('GPV/Kg:',

```



```

        style: TextStyle(
          fontFamily: 'Volkhov',
          fontSize: 20,
          color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('IMS/Kg/dia:',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    ),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [

```

```

Expanded(
  child: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 3),
    child: Text('NDT/Kg/dia:',
      style: TextStyle(
        fontFamily: 'Volkhov',
        fontSize: 20,
        color: Colors.white))),
Expanded(
  child: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 1),
    child: SizedBox(
      height: 30,
      child: TextField(
        decoration: InputDecoration(
          enabledBorder: UnderlineInputBorder(
            borderSide: BorderSide(color: Colors.white))),
      ),
    ),
  ),
),
),
],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('PB/Kg/dia:',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
      Expanded(
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 1),
          child: SizedBox(
            height: 30,
            child: TextField(
              decoration: InputDecoration(
                enabledBorder: UnderlineInputBorder(
                  borderSide: BorderSide(color: Colors.white))),
            ),
          ),
        ),
      ),
    ],
  ),
],

```

```

),
  SizedBox(height: 5),
  Row(
    children: [
      Expanded(
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 3),
          child: Text('Ca/g/dia:',
            style: TextStyle(
              fontFamily: 'Volkhov',
              fontSize: 20,
              color: Colors.white))),
      Expanded(
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 1),
          child: SizedBox(
            height: 30,
            child: TextField(
              decoration: InputDecoration(
                enabledBorder: UnderlineInputBorder(
                  borderSide: BorderSide(color: Colors.white))),
            ),
          ),
        ),
      ),
    ],
  ),
  SizedBox(height: 5),
  Row(
    children: [
      Expanded(
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 3),
          child: Text('P/g/dia:',
            style: TextStyle(
              fontFamily: 'Volkhov',
              fontSize: 20,
              color: Colors.white))),
      Expanded(
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 1),
          child: SizedBox(
            height: 30,
            child: TextField(
              decoration: InputDecoration(
                enabledBorder: UnderlineInputBorder(
                  borderSide: BorderSide(color: Colors.white))),
            ),
          ),
        ),
      ),
    ],
  ),

```

```

        ),
      ),
    ],
  ),
  Image.asset('assets/imagens/linha.png'),
  SizedBox(height: 5),
  Center(
    child: Text('Relação concentrado x volumoso',
      style: TextStyle(
        fontFamily: 'Volkhov', fontSize: 20, color: Colors.white)),
  ),
  Row(
    children: [
      Expanded(
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 3),
          child: Text('Volumoso%',
            style: TextStyle(
              fontFamily: 'Volkhov',
              fontSize: 20,
              color: Colors.white))),
      Expanded(
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 1),
          child: SizedBox(
            height: 30,
            child: TextField(
              decoration: InputDecoration(
                enabledBorder: UnderlineInputBorder(
                  borderSide: BorderSide(color: Colors.white))),
            ),
          ),
        ),
      ),
    ],
  ),
  SizedBox(height: 5),
  Row(
    children: [
      Expanded(
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 3),
          child: Text('Concentrado%',
            style: TextStyle(
              fontFamily: 'Volkhov',
              fontSize: 20,
              color: Colors.white))),

```

```

Expanded(
  child: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 1),
    child: SizedBox(
      height: 30,
      child: TextField(
        decoration: InputDecoration(
          enabledBorder: UnderlineInputBorder(
            borderSide: BorderSide(color: Colors.white))),
      ),
    ),
  ),
),
),
],
),
),
Image.asset('assets/imagens/linha.png'),
SizedBox(height: 5),
Center(
  child: Text('Concentrado fixado',
    style: TextStyle(
      fontFamily: 'Volkhov', fontSize: 20, color: Colors.white)),
),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('PB%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    ),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
            ),
          ),
        ),
      ),
    ],
  ),
),
SizedBox(height: 5),

```

```

Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('NDT%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('Ca%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
          ),
        ),
      ),
    ),
  ],
),

```

```

    ),
  ],
),
 SizedBox(height: 5),
 Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('P%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
          ),
        ),
      ),
    ),
  ],
),
 Image.asset('assets/imagens/linha.png'),
 SizedBox(height: 5),
 Center(
  child: Text('2º Concentrado',
    style: TextStyle(
      fontFamily: 'Volkhov', fontSize: 20, color: Colors.white)),
),
 Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('PB%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    Expanded(
      child: Padding(

```

```

padding: const EdgeInsets.symmetric(horizontal: 1),
child: SizedBox(
  height: 30,
  child: TextField(
    decoration: InputDecoration(
      enabledBorder: UnderlineInputBorder(
        borderSide: BorderSide(color: Colors.white))),
    ),
  ),
),
),
),
),
],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('NDT%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    ),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
            ),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('Ca%',
          style: TextStyle(
            fontFamily: 'Volkhov',

```



```

        fontSize: 20,
        color: Colors.white))),
Expanded(
  child: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 1),
    child: SizedBox(
      height: 30,
      child: TextField(
        decoration: InputDecoration(
          enabledBorder: UnderlineInputBorder(
            borderSide: BorderSide(color: Colors.white))),
      ),
    ),
  ),
),
],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('P%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    ),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
            ),
          ),
        ),
      ),
    ),
  ],
),
Image.asset('assets/imagens/linha.png'),
SizedBox(height: 5),
Center(
  child: Text('3º Concentrado',
    style: TextStyle(

```

```

        fontFamily: 'Volkhov', fontSize: 20, color: Colors.white))),
),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('PB%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('NDT%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
          ),
        ),
      ),
    ),
  ],
),

```

```

    ),
  ),
),
],
),
 SizedBox(height: 5),
 Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('Ca%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
          ),
        ),
      ),
    ),
  ],
),
 SizedBox(height: 5),
 Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('P%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(

```

```

        decoration: InputDecoration(
          enabledBorder: UnderlineInputBorder(
            borderSide: BorderSide(color: Colors.white))),
      ),
    ),
  ),
),
],
),
Image.asset('assets/imagens/linha.png'),
SizedBox(height: 5),
Center(
  child: Text('Minerais',
    style: TextStyle(
      fontFamily: 'Volkhov', fontSize: 20, color: Colors.white)),
),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('PB%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('NDT%',

```

```

        style: TextStyle(
          fontFamily: 'Volkhov',
          fontSize: 20,
          color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('Ca%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    ),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(
                borderSide: BorderSide(color: Colors.white))),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [

```

```

Expanded(
  child: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 3),
    child: Text('P%',
      style: TextStyle(
        fontFamily: 'Volkhov',
        fontSize: 20,
        color: Colors.white))),),
Expanded(
  child: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 1),
    child: SizedBox(
      height: 30,
      child: TextField(
        decoration: InputDecoration(
          enabledBorder: UnderlineInputBorder(
            borderSide: BorderSide(color: Colors.white))),
      ),
    ),
  ),
),
),
],
),
Image.asset('assets/imagens/linha.png'),
SizedBox(height: 5),
Center(
  child: Text('Preços',
    style: TextStyle(
      fontFamily: 'Volkhov', fontSize: 20, color: Colors.white)),
),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('Conc. fixado/Kg:',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(
              enabledBorder: UnderlineInputBorder(

```







```

);
}

class Formular extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: [
          SizedBox(
            height: MediaQuery.of(context).size.height,
            child: Image.asset(
              'assets/imagens/formular.jpg',
              fit: BoxFit.cover,
            ),
          ),
          _body()
        ],
      ),
    );
  }
}

```

Fonte: Desenvolvimento próprio do autor.

Anexo 11: *Scripts* do arquivo resultados.

```

import 'package:flutter/material.dart';

Widget _body() {
  return ListView(
    children: [
      SizedBox(height: 100),
      Center(
        child: Image.asset(
          'assets/imagens/logo.png',
          width: 130,
          height: 130,
        ),
      ),
      SizedBox(height: 50),
      Image.asset('assets/imagens/linha.png'),
      Row(
        children: [
          Expanded(
            child: Padding(
              padding: const EdgeInsets.symmetric(horizontal: 3),
              child: Text('Quant. de animais:',
                style: TextStyle(

```

```

        fontFamily: 'Volkhov',
        fontSize: 20,
        color: Colors.white))))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(),
          ),
        ),
      ),
    ),
  ],
),
Image.asset('assets/imagens/linha.png'),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('NDT%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('PB%',

```

```

        style: TextStyle(
          fontFamily: 'Volkhov',
          fontSize: 20,
          color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('Ca/g:',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    ),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('P/g:',

```

```

        style: TextStyle(
          fontFamily: 'Volkhov',
          fontSize: 20,
          color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(),
          ),
        ),
      ),
    ),
  ],
),
Image.asset('assets/imagens/linha.png'),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('GPV/dia:',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    ),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),

```

```

        child: Text('IMS/Kg/dia:',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('Volumoso%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    ),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),

```

```

        child: Text('Concentrado%',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(),
          ),
        ),
      ),
    ),
  ],
),
Image.asset('assets/imagens/linha.png'),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('1º Concentrado/Kg:',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white))),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(

```

```

padding: const EdgeInsets.symmetric(horizontal: 3),
child: Text('2ºConcentrado/Kg:',
  style: TextStyle(
    fontFamily: 'Volkhov',
    fontSize: 20,
    color: Colors.white)))),
Expanded(
  child: Padding(
    padding: const EdgeInsets.symmetric(horizontal: 1),
    child: SizedBox(
      height: 30,
      child: TextField(
        decoration: InputDecoration(),
      ),
    ),
  ),
),
),
),
],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('Conc. fixado/Kg:',
          style: TextStyle(
            fontFamily: 'Volkhov',
            fontSize: 20,
            color: Colors.white)))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(

```

```

        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('Minerais/Kg:',
            style: TextStyle(
                fontFamily: 'Volkhov',
                fontSize: 20,
                color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(),
          ),
        ),
      ),
    ),
  ],
),
SizedBox(height: 5),
Row(
  children: [
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 3),
        child: Text('Total/Kg:',
            style: TextStyle(
                fontFamily: 'Volkhov',
                fontSize: 20,
                color: Colors.white))),
    Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 1),
        child: SizedBox(
          height: 30,
          child: TextField(
            decoration: InputDecoration(),
          ),
        ),
      ),
    ),
  ],
),
Image.asset('assets/imagens/linha.png'),
SizedBox(height: 5),
Row(
  children: [
    Expanded(

```



```

        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 3),
          child: Text('Consumo de água:',
            style: TextStyle(
              fontFamily: 'Volkhov',
              fontSize: 20,
              color: Colors.white))),
        Expanded(
          child: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 1),
            child: SizedBox(
              height: 30,
              child: TextField(
                decoration: InputDecoration(),
              ),
            ),
          ),
        ),
      ],
    ),
    Image.asset('assets/imagens/linha.png'),
    SizedBox(height: 5),
    Row(
      children: [
        Expanded(
          child: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 3),
            child: Text('Custo da dieta:',
              style: TextStyle(
                fontFamily: 'Volkhov',
                fontSize: 20,
                color: Colors.white))),
          Expanded(
            child: Padding(
              padding: const EdgeInsets.symmetric(horizontal: 1),
              child: SizedBox(
                height: 30,
                child: TextField(
                  decoration: InputDecoration(),
                ),
              ),
            ),
          ),
        ],
      ),
    Image.asset('assets/imagens/linha.png'),
    SizedBox(height: 60),
    RaisedButton(

```

```
        color: Colors.lightBlue,
        onPressed: () {},
        child: Text('Nova Formulação',
            style: TextStyle(
                fontFamily: 'Volkhov',
            )),
        SizedBox(height: 60),
    ],
);
}

class Resultados extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Stack(
        children: [
          SizedBox(
            height: MediaQuery.of(context).size.height,
            child: Image.asset(
              'assets/imagens/resultados.jpg',
              fit: BoxFit.cover,
            ),
          ),
          _body()
        ],
      ),
    );
  }
}
```

Fonte: Desenvolvimento próprio do autor.