

UNIVERSIDADE FEDERAL DO PARANÁ
CURSO DE AGRONOMIA

Adriana Geray Artigas
Pierre Geovane Graff
Sabrina Giovana Clarindo
William Santiago de Mendonça

CÁLCULO NUTRICIONAL DE GADO DE CORTE CONFINADO

CURITIBA

2018

Adriana Geray Artigas (GRR20105605)

Pierre Geovane Graff

Sabrina Giovana Clarindo

William Santiago de Mendonça (GRR20185065)

CÁLCULO NUTRICIONAL DE GADO DE CORTE CONFINADO

Relatório apresentado à disciplina
Fundamentos de Programação de
Computadores do Curso de Graduação
em Agronomia da Universidade Federal
do Paraná.

Orientador: Prof. Jackson Antônio do Prado Lima

Curitiba, novembro de 2018

1 INTRODUÇÃO

O presente trabalho busca estimar a quantidade de alimento seco somado ao concentrado fornecido para o bovino, para que suas necessidades sejam supridas. Para isso foi desenvolvido um programa, realizado na linguagem Python. Os dados utilizados foram retirados de uma apostila de nutrição animal, nesta estavam presentes necessidades nutricionais dos animais, bem como a composição química de diversos alimentos.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Tem como objetivo facilitar o calculo nutricional, que por vezes pode ser muito demorado e trabalhoso. Com a escolha do animal desejado pelo produtor, seu peso e alimentos a serem oferecidos, o programa utiliza de seu banco de dados para efetuar os cálculos de algumas das exigências, não é exato, porém muito aproximado.

3. DESENVOLVIMENTO

Inicialmente foi criado um banco de dados, as informações foram retiradas de uma apostila de nutrição animal. Os dados foram agrupados em lista e em matrizes, para que sua utilização fosse facilitada.

Com a entrada, o programa percorreu cada lista\matriz, extraindo os dados necessários.

```
for c in range((len(classes))):
    print("{}-{}".format(c, classes[c]))
clas=int(input(">"))
```

```
for p in range((len(pesos[clas]))):
    print("{}-{}".format(p, pesos[clas][p]))
pes=int(input(">"))
for n in range((len(valores[clas][pes]))):
    necessidades.append(valores[clas][pes][n])
```

```
if tipo_alimento=="feno":
    ali=0
    for f in range((len(fenos))):
        print("{}-{}".format(f, fenos[f]))
    ali=int(input(">"))
    for c in range((len(alimentos_vol[0][ali]))):
        nutrientes_fornecidos.append(alimentos_vol[0][ali][c])
    for g in fenos:
        alimento.append(g)
elif tipo_alimento=="forragem":
    ali=0
    for f in range((len(forragens))):
        print("{}-{}".format(f, forragens[f]))
    ali=int(input(">"))
    for c in range((len(alimentos_vol[1][ali]))):
        nutrientes_fornecidos.append(alimentos_vol[1][ali][c])
    for g in forragens:
        alimento.append(g)
elif tipo_alimento=="silagem":
    ali=0
    for f in range((len(silagens))):
        print("{}-{}".format(f, silagens[f]))
    ali=int(input(">"))
    for c in range((len(alimentos_vol[2][ali]))):
        nutrientes_fornecidos.append(alimentos_vol[2][ali][c])
    for g in silagens:
        alimento.append(g)
```

```
conc=0
for c in range((len(alimentos_c))):
    print("{}-{}".format(c, alimentos_c[c]))
conc=int(input(">"))
nutrientes_concentrados=[]
for num in range((len(porcentagens[conc]))):
    nutrientes_concentrados.append(porcentagens[conc][num])
```

Por serem muito grandes, as listas não foram adicionadas no presente relatório

Subsequentemente, os cálculos necessários foram reproduzidos no programa, por se tratar de uma regra de três composta, diversas variáveis foram usadas.

```
volumoso_pb=(nutrientes_fornecidos[1])/100
volumoso_ndt=(nutrientes_fornecidos[4])/100
concentrado_pb=(nutrientes_concentrados[1])/100
concentrado_ndt=(nutrientes_concentrados[4])/100
necessidade_pb=(necessidades[1])
necessidade_ndt=(necessidades[3])
novo_vpb=volumoso_pb*volumoso_ndt
novo_vndt=volumoso_ndt*volumoso_pb
novo_n_pb=necessidade_pb*volumoso_ndt
novo_n_ndt=necessidade_ndt*volumoso_pb
novo_cpb=concentrado_pb*volumoso_ndt
novo_cndt=concentrado_ndt*volumoso_pb
novo_vpb=novo_vpb-novo_vndt
novo_cpd=abs(novo_cpb-novo_cndt)
novo_n_pb=abs(novo_n_pb-novo_n_ndt)
kg_c=(novo_n_pb)/novo_cpb
kg_v=((necessidade_pb)-(kg_c*concentrado_pb))/volumoso_pb
kg_v=abs(kg_v)
#calculo de suplementação
sup=["Acido fosforico H3PO4","Calcáreo CaCO3"]
suplementos_100=[0.361,0.385]
sup_p=0
sup_ca=0
ca_vol=kg_v*((nutrientes_fornecidos[5])/100)
ca_c=kg_c*((nutrientes_concentrados[5])/100)
p_vol=kg_v*((nutrientes_fornecidos[6])/100)
p_c=kg_c*((nutrientes_concentrados[6])/100)
p_fornecido=p_vol+p_c
ca_fornecido=ca_vol+ca_c
p_faltante=(necessidades[5])-p_fornecido
ca_faltante=(necessidades[4])-ca_fornecido
sup_p=p_faltante/(suplementos_100[0])
sup_ca=ca_faltante/(suplementos_100[1])
```

Após o calculo de algumas exigências, foi feito o calculo da suplementação, como se visualiza acima.

Por fim, foi elaborada uma saída em que somente o necessário é mostrado, pois em alguns casos, a suplementação não se faz necessária.

4.CONCLUSÃO

Pode se concluir que a nutrição animal é uma área muito importante e extremamente programável, pois seu cálculos são muito extensos e repetitivos.

Quanto a programação em si, nota-se que existe uma certa dificuldade de transcrever conceitos para a linguagem do computador.

5.REFERENCIAS BIBLIOGRÁFICAS

BÉRTOLI, Claudia Damo. **Nutrição Animal Aplicada e Alimentação dos Animais Domésticos**, 2010. Instituto Federal Catarinense *Campus* Camboriú