

UNIVERSIDADE FEDERAL DO PARANÁ  
CURSO DE AGRONOMIA

JOÃO PEDRO DE BARROS LEINECKER  
EDUARDA KERUK POSSOLI  
THIAGO HENRIQUE DE OLIVEIRA

Curitiba  
2018

JOÃO PEDRO DE BARROS LEINECKER

EDUARDA KERUK POSSOLI

THIAGO HENRIQUE DE OLIVEIRA

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

## **SUMÁRIO**

<b>1 Introdução .....</b>	<b>4</b>
<b>2 Objetivo Geral.....</b>	<b>5</b>
<b>2.1 Objetivos específicos.....</b>	<b>5</b>
<b>3 Desenvolvimento.....</b>	<b>6</b>
<b>5 Conclusão.....</b>	<b>14</b>
<b>6 Referências .....</b>	<b>15</b>

## **1 INTRODUÇÃO**

Este trabalho visa criar um programa em python 3.4 que solicite ao usuário um conjunto de sete informações referentes a capacidade de uso do solo, retornando ao usuário como resposta uma orientação da possível aptidão do solo analisado.

Dentro da temática dos recursos naturais renováveis, viemos a partir de informações referentes ao solo, reclamar uma possível aptidão como forma de orientar tomadas de decisões em áreas de difícil acesso a um profissional habilitado.

## **2 OBJETIVO GERAL**

Construir um programa em python 3.4 .

### **2.1 OBJETIVOS ESPECÍFICOS**

Adquirir noções básicas em python 3.4.

Construir uma problemática envolvendo conhecimentos agronômicos e linguagem de programação.

Correlacionar informações agronômicas em linguagem de programação.

Produzir um programa capaz de auxiliar na tomada de decisões do dia a dia no campo.

### 3 DESENVOLVIMENTO

O programa consiste em solicitar sete informações referentes a capacidade de uso do solo, que estão divididos em fertilidade do solo(f) com duas informações (condutividade elétrica do solo [ce] e saturação de bases [v%]), disponibilidade de água(h), disponibilidade de oxigênio no solo(o), erosividade(e) e por fim mecanização(m) esta também com duas informações a serem introduzidas pelo usuário (profundidade do solo [m2] e a porcentagem de pedregosidade no perfil do solo[m1]). Estas informações serão ranqueadas (0-4) e interpoladas pelo programa que retornará ao usuário a possível aptidão do solo que pode ser uma das seguintes opções: lavoura, área de preservação, pastagem e reflorestamento.

## Ex. do programa:

```
C:\Python34\python.exe
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:43:06) [MSC v.1600 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> # f - fertilidade do solo (CE e V%)
... # f - fertilidade do solo (CE e V%)
... # h - disponibilidade de agua
... # o - oxigenação do solo
... # e - erosão
... # m - mecanização
...
>>> #inicialização das variáveis auxiliares
... f = 100 #variável para fertilidade do solo
>>> h = 0 #variável para disponibilidade de água
>>> h1 = 100 #variável para disponibilidade de água
>>> o1 = 100 #variável para oxigenação do solo
>>> e1 = 100 #variável para erosão
>>> m1 = 100 #variável para mecanização
>>>
>>> print("\t\tCurso de Agronomia\n")
        Curso de Agronomia

>>> print("\t\tTrabalho da Disciplina de Programação\n")
        Trabalho da Disciplina de Programação

>>> print("\t\tCapacidade e Aptidão do Solo\n")
        Capacidade e Aptidão do Solo


>>> print("\tAcademicos: EDUARDA, JOÃO E THIAGO\n")
        Academicos: EDUARDA, JOÃO E THIAGO

>>>
>>> while(f==100):
...     # leitura do CE - condutibilidade eletrica
...     ce = float(input("Qual a condutividade elétrica do solo?\n"))
...     # leitura do V% - saturação de bases
...     v = float(input("\nQual a saturação por bases do solo?\n"))
...     # verificação do nível de fertilidade
...     if ce < 4 and v > 80:
...         f = 0
...     elif ce < 4 and v > 50:
...         f = 1
...     elif 4 < ce < 8 and 35 < v < 50:
...         f = 2
...     elif 8 < ce < 15 and v < 35:
...         f = 3
...     elif ce > 15 and v < 35:
...         f = 4
...     else: print("\nERRO: Valor de condutividade elétrica e/ou saturação por bases do solo inválidos\n")
...
Qual a condutividade elétrica do solo?
# leitura do h - Disponibilidade de agua
```

Exemplo de como foi resolvido a interpolação das informações sobre fertilidade do solo:

```
>>> while(f==100):
..     # leitura do CE - condutibilidade eletrica
..     ce = float(input("Qual a condutividade elétrica do solo?\n"))
..     # leitura do V% - saturação de bases
..     v = float(input("\nQual a saturação por bases do solo?\n"))
..     # verificação do nível de fertilidade
..     if ce < 4 and v > 80:
..         f = 0
..     elif ce < 4 and v > 50:
..         f = 1
..     elif 4 < ce < 8 and 35 < v < 50:
..         f = 2
..     elif 8 < ce < 15 and v < 35:
..         f = 3
..     elif ce > 15 and v < 35:
..         f = 4
..     else: print("\nERRO: Valor de condutividade elétrica e/ou saturação p
```



 C:\Python34\python.exe

Qual a condutividade elétrica do solo?

# leitura do h - Disponibilidade de agua

Traceback (most recent call last):

File "<stdin>", line 3, in <module>

ValueError: could not convert string to float: '# leitura do h - Disponibili

>>> while(h1==100):

.. h = int(input("\nQual a disponibilidade de água, h pode ser:\n0 - Não  
STIAGEM \n"))

.. if h==0:

.. h1 = 0

.. elif h == 1:

.. h1 = 1

.. elif h == 2:

.. h1 = 2

.. elif h == 3:

.. h1 = 3

.. elif h == 4:

.. h1 = 4

.. else: print("\nERRO: Valor de disponibilidade de h2o inválido\n")

..

```
Python 3.4.3 (v3.4.3:9b73f1c3e601, Feb 24 2015, 22:4
Type "help", "copyright", "credits" or "license" for
>>> # f - fertilidade do solo (CE e V%)
.. # f - fertilidade do solo (CE e V%)
.. # h - disponibilidade de agua
.. # o - oxigenação do solo
.. # e - erosão
.. # m - mecanização
..
>>> #inicialização das variáveis auxiliares
.. f = 100 #variável para fertilidade do solo
>>> h = 0 #variável para disponibilidade de água
>>> h1 = 100 #variável para disponibilidade de água
>>> o1 = 100 #variável para oxigenação do solo
>>> e1 = 100 #variável para erosão
>>> m1 = 100 #variável para mecanização
>>>
>>> print("\t\tCurso de Agronomia\n")
        Curso de Agronomia

>>> print("\t\tTrabalho da Disciplina de Programação
        Trabalho da Disciplina de Programação

>>> print("\t\tCapacidade e Aptidão do Solo\n")
        Capacidade e Aptidão do Solo

>>> print("\tAcademicos: EDUARDA, JOÃO E THIAGO\n")
        Academicos: EDUARDA, JOÃO E THIAGO

>>>
>>> while(f==100):
..     # leitura do CE - condutibilidade eletrica
..     ce = float(input("Qual a condutividade elétri
..     # leitura do V% - saturação de bases
..     v = float(input("\nQual a saturação por bases
```

As informações referentes a disponibilidade de água, oxigenação do solo e erosividade apresentaram uma maior facilidade de interpolação que as informações referentes a mecanização e fertilidade do solo, as quais necessitavam de um maior numero de respostas possíveis. Como podemos analisar :]

```
C:\Python34\python.exe
Qual a disponibilidade de água, h pode ser:
0 - Não há definição
1 - DE 3 A 5 MESES DE ESTIAGEM
2 - DE 4 A 6 MESES DE ESTIAGEM
3 - De 7 a 9 MESES DE ESTIAGEM
4 - MAIS DE 9 MESES DE ESTIAGEM
# leitura do e - Declividade
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
ValueError: invalid literal for int() with base 10: '# leitura do e - Declividade'
>>> while(e1==100):
...     e = float(input("\nQual o percentual(%) de declividade (0 a 45%)?\n"))
...     if e < 3:
...         e1 = 0
...     elif 3 <= e < 8:
...         e1 = 1
...     elif 8 <= e < 13:
...         e1 = 2
...     elif 13 <= e < 20:
...         e1 = 3
...     elif 20 <= e <= 45:
...         e1 = 4
...     else: print("\nERRO: Valor de declividade inválido\n")
...
Qual o percentual(%) de declividade (0 a 45%)?
# leitura do o - oxigenação do solo
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
ValueError: could not convert string to float: '# leitura do o - oxigenação do solo'
>>> while(o1==100):
...     o = int(input("\nQual a oxigenação do solo?\n0 - DRENADO\n1- MODERADAMENTE DRENADO\n2 - MAL DRENADO\n3 - MAIS QUE MAL DRENADO\n4 - INUNDAÇÕES FREQUENTES\n"))
...     if o == 0:
...         o1 = 0
...     elif o == 1:
...         o1 = 1
...     elif o == 2:
...         o1 = 2
...     elif o == 3:
...         o1 = 3
...     elif o == 4:
...         o1 = 4
...     else: print("\nERRO: Valor de oxigenação inválido\n")
...
```

C:\Python34\python.exe

Qual a oxigenação do solo?

0 - DRENADO

1- MODERADAMENTE DRENADO

2 - MAL DRENADO

3 - MAIS QUE MAL DRENADO

4 - INUNDAÇÕES FREQUENTES

#leitura da mecanização m

Traceback (most recent call last):

File "<stdin>", line 2, in <module>

ValueError: invalid literal for int() with base 10: '#leitura da mecanização m'

>>> while(m1==100):

... m = int(input("\nQual a quantidade de pedras no perfil do solo?\n0 -

... if m == 0:

... m1 = 0

... elif m == 1:

... m1 = 1

... elif m == 2:

... m1 = 2

... elif m == 3:

... m1 = 3

... elif m == 4:

... m1 = 4

... else: print("\nERRO: Valor de percentual de pedras inválido\n")

...

Qual a quantidade de pedras no perfil do solo?

0 - NÃO PEDREGOSA

1 - MODERADAMENTE PEDREGOSA

2 - PEDREGOSA

3 - MUITO PEDREGOSA

4 - EXTREMAMENTE PEDREGOSA

# validando para que o solo é adequado apartir das respostas do usuário


Traceback (most recent call last):

File "<stdin>", line 2, in <module>

ValueError: invalid literal for int() with base 10: '# validando para que o

>>> print("\n\t RESULTADO: \n")

Para obtermos o resultado esperado, foi necessário estabelecer uma serie de condições que variaram de 0 a 4 sendo zero para o cultivo intensivo da área na forma de lavoura, e quatro para área de preservação ambiental, como podemos ver no exemplo abaixo:

 C:\Python34\python.exe

RESULTADO:

```
>>> if f == 0 and h == 1 and o == 1 and e1 == 0 and m == 1:
...     print("\t\tLAVOURA\n")
... elif f == 1 and h == 2 and o == 1 and e1 == 1 and m == 2:
...     print("\t\tLAVOURA\n")
... elif f == 2 and h == 3 and o == 2 and e1 == 2 and m == 3:
...     print("\t\tLAVOURA\n")
... elif f == 2 and h == 2 and o == 3 and e1 == 2 and m == 2:
...     print("\t\tPASTAGEM\n")
... elif f == 3 and h == 3 and o == 3 and e1 == 3 and m == 3:
...     print("\t\tPASTAGEM\n")
... elif f == 4 and h == 4 and o == 4 and e1 == 4 and m == 3:
...     print("\t\tPASTAGEM\n")
... elif f == 2 and h == 2 and o == 1 and e1 == 3 and m == 2:
...     print("\t\tREFLORESTAMENTO\n")
... elif f == 3 and h == 3 and o == 1 and e1 == 3 and m == 3:
...     print("\t\tREFLORESTAMENTO\n")
... elif f == 4 and h == 4 and o == 2 and e1 == 4 and m == 3:
...     print("\t\tREFLORESTAMENTO\n")
... else: print("\nAREA DE PRESERVAÇÃO AMBIENTAL\n")
... 
```

## **Conclusão**

Conclui-se que a linguagem python pode ser uma excelente ferramenta para a elaboração de programas que auxiliem na otimização e eficiência da atividade profissional nomeadamente ao eng. Agrônomo. Apesar da dificuldade em aplicar esta ferramenta, é praticamente inevitável imaginar o futuro profissional sem utiliza-la, para tanto este trabalho de pesquisa e desenvolvimento é estratégico para progresso desta área do conhecimento.

## REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (ABNT). Informação documentação - Trabalhos acadêmicos - Apresentação. **NBR14724**. Rio de Janeiro, 2011.

AMADEU, M. S. U. et al. **Manual de normalização de documentos científicos de acordo com as normas da NT**. Curitiba, 2015.