



**UNIVERSIDADE FEDERAL DO PARANÁ**  
**CURSO DE AGRONOMIA**

Bianca Maria Chastalo

July Caroline Lopes de Andrade

**PRODUTIVIDADE MÉDIA DE GRÃOS: SOJA OU MILHO EM PYTHON**

**CURITIBA**

**2018**

**BIANCA MARIA CHASTALO - GRR20180796**

**JULY CAROLINE LOPES DE ANDRADE - GRR20186699**

## **PRODUTIVIDADE MÉDIA DE GRÃOS: MILHO OU SOJA EM PYTHON**

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 - OBJETIVOS .....</b>	<b>5</b>
2.1 OBJETIVO GERAL .....	5
<b>3 - DESENVOLVIMENTO .....</b>	<b>6</b>
3.1 O TRABALHO .....	6
3.2 FÓRMULAS UTILIZADAS .....	9
<b>4 - CONCLUSÃO .....</b>	<b>10</b>
<b>REFERÊNCIAS .....</b>	<b>11</b>

## 1 - INTRODUÇÃO

Este relatório tem o objetivo de apresentar o projeto final do trabalho “Produtividade média de grãos: Milho ou Soja” em Python, demonstrando as etapas de como foi realizado e seu resultado final.

Inicialmente iríamos fazer um programa que informasse ao produtor de grãos (milho ou soja) a data provável da colheita com base em informações que seriam informadas por ele e também sua produtividade média, porém, não foi possível desenvolvermos um programa que estimasse a provável data da colheita, pois, não encontramos dados suficientes e concretos, já que isso é algo que varia muito dependendo da região, do clima, e outros fatores. Dessa forma fizemos um programa que calcula apenas a produtividade média dos grãos.

## **2 - OBJETIVOS**

### **2.1 OBJETIVO GERAL**

O objetivo geral do nosso trabalho foi fazer um programa direcionado à quem já plantou alguns dos grãos (milho ou soja), pensando em uma maior comodidade para a pessoa que deseja saber a produtividade. Essa comodidade se dá ao fato de que a pessoa apenas informa ao programa alguns dados e informações na lavoura. Os dados são utilizados em uma fórmula já desenvolvida (a qual daremos mais informações ao decorrer do trabalho). E então é retornada a produtividade média para quem estiver usando o programa.

### 3 - DESENVOLVIMENTO

#### 3.1 O TRABALHO

O programa conta com 3 funções, sendo duas funções apenas para cálculos e mais uma que trata do menu. Uma 4ª função aparece, mas é aquela que faz com que o programa seja executado, apenas chamando a função menu.

A primeira função trata dos cálculos para estimar a produtividade média de soja. Quando essa função é chamada, ela primeiramente pede dados ao produtor.

```

10 def produtividade_soja():
11     """
12     Função que calcula produtividade de soja
13     Dentro da função os dados necessários para a conta serão pedidos
14     """
15
16     # Solicita alguns dados do usuário
17     plantas10m = float(input("\nDigite a quantidade de plantas em dez metros: "))
18     esp_m = float(input("Digite o espaçamento em metros: "))
19     quantVag = float(input("Digite a quantidade total de vagens em dez plantas consecutivas: "))

```

Figura 1- Função para dados de entrada (cálculo da produtividade da soja)

Depois são feitos os cálculos necessários para estimar a produtividade da soja em sacos/ha. E logo após retorna o resultado para o usuário, utilizando apenas 2 casas decimais após a vírgula.

Após dar o resultado, a função menu é chamada para que o usuário possa escolher se quer fazer um novo cálculo ou sair do programa.

```

20
21     # Cálculo de médias com os valores informados
22     plantasHa = ((plantas10m / 10) / esp_m) * 10
23     mediaVagens = quantVag / 10
24
25     # Conta para definir a quantidade de sacos/ha
26     # Valores padrão - "2.5": valor (médio) de sementes por vagem/ "170": peso (médio) de mil grãos
27     sacos = (plantasHa * mediaVagens * 2.5 * 170) / 60000
28
29     print("\nCom base nos valores informados, a média de produtividade da soja será de {:.2f} sacos/ha.\n".format(sacos))
30
31     # Chama função menu() para o usuário escolher fazer outro cálculo ou sair do programa
32     menu()
33     return sacos

```

Figura 2 - Continuação com os cálculos e finalização retornando à produtividade

A segunda função é sobre a produtividade média do milho. Nessa função tivemos complicações porque precisávamos informar ao usuário uma tabela que serviria de referência e não sabíamos como fazer para que ela fosse exibida. Felizmente, com a ajuda do professor Jackson e com o exemplo de relatório fornecido à nós, foi possível seu desenvolvimento.

Após mostrar a tabela, o programa pede um dado que será informado com base na mesma.

```
def produtividade_milho():
    """
    Função que calcula produtividade de milho
    Dentro da função os dados necessários para a conta serão pedidos
    """

    # Mostra uma tabela para ajudar o usuário a informar os próximos dados
    print("\nEspacamento(cm)\tComprimento para ter 4m\n50\t1\t1\t6,6m\n70\t1\t1\t5,7m\n80\t1\t1\t5m\n90\t1\t1\t4,4m\n100\t1\t1\t4m")

    espigas4m = float(input("\nUsando a tabela como referência, informe o número de espigas em 4m:"))
```

Figura 3 – Tabela para função da produtividade do milho

Pedimos informações ao usuário e dizemos que os valores devem ser informados sendo separados por espaço e essas informações são armazenadas em uma lista vazia. Para serem armazenadas na lista usamos a função `.split()` que quebra os valores informados para colocar na lista.

Criamos também uma condição para que o usuário não informe mais informações do que o necessário. Após isto, alguns cálculos são desenvolvidos.

```
# Cria lista vazia
graos = []

while len(graos) != 3:
    graos = [float(g) for g in input("Com base em 3 espigas, informe a quantidade de grãos de cada uma, separando as quantidades por espaço:").split()]
    if len(graos) != 3:
        print("Você deve informar 3 valores!")

somaGraos = len(graos) # Conta a quantidade de valores na lista
mediaGraos = sum(graos) / somaGraos # Soma os valores da lista e calcula média
```

Figura 4 - Cálculo para a produtividade do milho

Criamos mais uma lista e mais uma condição com a lógica idêntica à lista mostrada acima. Mais cálculos são feitos com a mesma lógica também, e finalmente estimamos a produtividade do milho em kg/ha e logo devolvemos ela ao usuário.

```

1. Mesma lógica da lista acima
fileiras = []
while len(fileiras) != 3:
    fileiras = [float(f) for f in input("Com base nas mesmas espigas, informe a quantidade de fileiras de cada uma, separando as quantidades por espaço:").split()]
    if len(fileiras) != 3:
        print("Você deve informar apenas 3 valores!")
somafileiras = len(fileiras)
mediafileiras = sum(fileiras) / somafileiras

# Conta para definir a média de produtividade
# Valor padrão - "0.7": fator de correção da transformação de bushels/acre para kg/ha
prod = espigas4m * mediafileiras * mediaGraos * 0.7

print("\nCom base nos dados informados, a média de produtividade do milho com 15,5% de umidade será de {:.2f} kg/ha.\n".format(prod))

# Chama função menu() para o usuário escolher fazer outro cálculo ou sair do programa
menu()
return prod

```

Figura 5 – Última parte do cálculo de produtividade do milho

A terceira função trata do menu do programa. Nessa função mostramos ao usuário opções para escolher entre o milho e a soja ou então sair do programa.

Depois fizemos condições para chamar a função correta de acordo com a escolha do usuário.

Nessa parte tivemos bastante dificuldade para fazer as condições, tentamos de várias formas e todas davam algum erro. Depois fizemos algumas mudanças e obtivemos o resultado esperado.

```
def menu():
    """
    Função do menu do programa
    """
    # Mostra uma mensagem de boas-vindas e as opções
    print("Vamos estimar produtividade? Escolha a opção que corresponde ao grão plantado ou digite outro número para sair.\n\t1 - Soja\t2 - Milho")
    opcao = int(input("\nOpção:"))

    # Se a opcao for 1, chama a função produtividade_soja()
    if opcao == 1:
        produtividade_soja()
    # Se a opcao for 2, chama a função produtividade_milho()
    elif opcao == 2:
        produtividade_milho()
    # Se a opcao não for 1 nem 2, a mensagem será exibida e o programa encerrado
    else:
        print("Volte sempre!")
```

Figura 6 – Função do menu do programa

A última função é a `if __name__ == "__main__"` que chama apenas a função menu quando executar o programa.

No começo não estávamos entendendo muito bem para que servia essa função e estávamos inserindo todas as outras funções nela. Quem nos ajudou a entender o erro foi o professor Jackson.

```
if __name__ == "__main__":
    """
    Quando executar o programa Python esse trecho de código será executado
    """

    # Chama a função menu
    menu()
```

Figura 7 – Função `if __name__ == "__main__"`



### 3.2 FÓRMULAS UTILIZADAS

Cálculo de produtividade da soja:

$$\frac{\text{Nº Plantas por Metro}}{\text{Espaçamento (metros)}} \times 10 = \text{Nº Plantas por Hectare (em milhares)}$$

Para o cálculo, utilizamos a fórmula abaixo:

$$\frac{\text{Plantas por ha} \left( \frac{\text{mil}}{\text{ha}} \right) \times \text{Vagens por Planta} \times \text{Semente por Vagem} \times \text{Peso de Mil Sementes} \left( \frac{\text{g}}{1000} \right)}{60000} = \text{sc/ha}$$

Retirado de: <<http://www.pioneersementes.com.br/blog/46/estimando-a-produtividade-na-cultura-da-soja>>

Cálculo de produtividade do milho:

1 – Conte o número de espigas em 4m²:

Espaçamento entre linhas (cm)	Comprimento para se obter 4m²
50	8,0 m
60	6,6 m
70	5,7 m
80	5,0 m
90	4,4 m
100	4,0 m

$$(\text{nº de espigas em } 4\text{m}^2) \times (\text{nº de fileiras de grãos}) \times (\text{nº de grãos por fileira}) \times 0,70^* = \text{kg/ha com 15,5\% de umidade}$$

\*Fator de correção do método e transformação de bushels/acre para kg/ha.

$$\text{Produtividade (toneladas/ha a 15,5\% de umidade)} = \frac{[(\text{NE} \times \text{P}) / \text{EM}]}{1000}$$

**NE:** Número médio de espigas em 10m lineares

**P:** Peso médio de grãos por espiga corrigido para 15,5% de umidade, obtido pela média do peso de grãos de 3 espigas coletadas (gramas)

**EM:** Espaçamento entre linhas (m)

Retirado de: <<http://www.pioneersementes.com.br/blog/26/como-estimar-a-produtividade-do-milho>>

#### **4 - CONCLUSÃO**

A partir deste trabalho concluímos que elaborar um trabalho complexo e com pessoas que não temos muito contato gera um efeito nada satisfatório, e dessa forma cria-se uma barreira e alguns problemas que de alguma forma afetam o desenvolvimento do trabalho. Mas em relação ao trabalho em si, pode-se dizer que foi bem cansativo e muito estressante pelo fato de não ser algo que estamos acostumados a lidar. Programação é uma disciplina essencial para quem quer ser um profissional completo e também requer total atenção e dedicação, além de muita força de vontade para aprender. Infelizmente o semestre é muito curto para tanta informação, mas com certeza todo conhecimento e aprendizado será levado e muito utilizado já que trabalhamos com o raciocínio lógico nessa matéria, e isso é importante tanto para nossa futura profissão quanto para a vida.

## REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (ABNT). Informação documentação - Trabalhos acadêmicos - Apresentação. NBR14724. Rio de Janeiro, 2011. Acessado em 17 de novembro de 2018 às 15:32

<<https://docs.ufpr.br/~pdalzoto/2018BIOTECManualNormasUFPR.pdf>>  
Acessado em: 18 de novembro de 2018 às 11:50

<<http://www.pioneersementes.com.br/blog/46/estimando-a-produtividade-na-cultura-da-soja>> Acessado em: 20 de novembro de 2018 às 20:59

<<http://www.pioneersementes.com.br/blog/26/como-estimar-a-produtividade-do-milho>> Acessado em: 20 de novembro de 2018 às 21:00