



UNIVERSIDADE FEDERAL DO PARANÁ
CURSO DE AGRONOMIA

Gabrieli Polochenski
Jéssica Maria Moreira dos Santos
Thaiz Milenna Dias

PYTHON DIRECIONADO À UM ESTACIONAMENTO

CURITIBA
2018

Gabrieli Polochenski (GRR20185972)
Jéssica Maria Moreira dos Santos (GRR 20180832)
Thaiz Milenna Dias (GRR20187231)

PYTHON DIRECIONADO À UM ESTACIONAMENTO

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

1 INTRODUÇÃO.....	3
2 OBJETIVO.....	4
2.1 OBJETIVO GERAL	4
3 DESENVOLVIMENTO.....	5
4 CONCLUSÃO.....	10
5 REFERÊNCIAS.....	11

1 INTRODUÇÃO

Esse relatório apresenta informações relativas ao trabalho realizado, tendo como tema “Python direcionado à um estacionamento”, o mesmo foi realizado em trio.

A linguagem Python foi concebida em 1989 pelo holandês Guido van Rossum, visto que ele estava desenvolvendo a linguagem ABC no CWI em Amsterdã – Holanda, e estava encontrando deficiências nessa linguagem. Tentando suprir esses problemas visto com o ABC, o holandês criou o Python com base em C.

Python é uma linguagem expressiva, em que é fácil traduzir o raciocínio em um algoritmo. Em aplicações científicas, o raciocínio é essencialmente complicado — essa é a natureza das ciências. É um problema adicional para o cientista ter que se preocupar com, além do assunto básico de sua pesquisa, a correção do programa em detalhes pouco relevantes: alocação de memória, gerenciamento de recursos, etc. Python faz isso tudo automaticamente de maneira muito eficiente

2 OBJETIVOS

2.1 OBJETIVO GERAL

O objetivo é a obtenção de um programa que seja capaz de prestar suporte administrativo à um estabelecimento comercial, no caso um estacionamento, com o intuito final de controlar o lucro obtido diariamente.

O programa é capaz de distinguir o valor a ser cobrado do cliente, através de critérios como o porte do automóvel, período de tempo em que o automóvel permaneceu estacionado no mesmo, dia da semana em que foi solicitado o serviço, tendo variação de valores em dias uteis, finais de semana e feriados. E também é capaz de gerar um desconto para clientes que apresentarem alguma deficiência e/ou são idosos.

3 DESENVOLVIMENTO

3.1 O Trabalho

Ao iniciar-se o código está presente a primeira função, que tem como objetivo a obtenção do lucro diário do estabelecimento, direcionando o cálculo com os critérios principais estabelecidos pela mesma, sendo eles o dia da semana e a quantidade de horas em que o automóvel utilizou os serviços do estabelecimento. Tendo variação de preços entre os dias úteis e finais de semana/feriado, além de dar desconto para ambas situações: a idosos e deficientes, retornando o lucro com o auxílio de if dentro de if .

Nas primeiras linhas do código estão declaradas as funções: a primeira será usada para obter o lucro diário da empresa direcionando o cálculo com certos critérios estabelecidos pela empresa que tem preços diferentes para dia útil ou se em fim de semana e feriado, além de dar desconto para ambas situações a idosos e deficientes, retornando o lucro com o auxílio de if dentro de if.

```
def lucrodinario(quem,numerohoras,dia):
    if dia==1:#se dia normal
        if quem==3:
            lucro=numerohoras*7#7 reais agora
        else:
            lucro=numerohoras*5#5 reais p idosos/defici
    else:#se fim de semana
        if quem ==3:
            lucro=numerohoras*9 #pra outros
        else:
            lucro=numerohoras*7#idosos/deficientes
    return lucro
```

A segunda função retornará o valor a ser pago pelo cliente através da entrada de horas (sendo a hora de entrada e a de saída) semelhantemente a primeira função, mas sendo armazenada na matriz, para cadastros quando esta está sendo preenchida.

```
def quantoapagar(horaentrada,horasaida,quem,dia):
    varquanto=horasaida-horaentrada
    if dia==1:#se dia normal
        if quem==3:
            varquanto*=7#7 reais agora
        else:
            varquanto*=5#5 reais p idosos/defici
    else:#se fim de semana
```

```

    if quem ==3:
        varquanto*=9 #pra outros
    else:
        varquanto*=7
    return varquanto

for v in range (1):
    chegada.append(str(input("placa: ")))
    dia=int(input("(1) (dia util)\n(2) (fim de semana)\n"))
    quem=int(input("(1) idoso\n(2) com deficiência\n(3) outros\n>"))

categoria=int(input("categoria:\n(1) carro\n(2) moto\n(3) caminhonete\n>"))
    horaentrada=int(input("hora de entrada: "))
    horasaida=int(input("hora de saida: "))
    quanto=quantoapagar(horaentrada,horasaida,quem,dia) #chama a
função de quanto a pagar
    chegada.append(quem)
    chegada.append(horaentrada)
    chegada.append(horasaida)
    chegada.append(quanto) #armazenamento
    matriz.append(chegada)

```

São então declaradas as variáveis e listas que serão usadas durante o código.

```

lucro=0#estou usando na def
moto=[]#lista para saber quantas vagas disponiveis a motos
carro=[]#lista para saber quantas vagas disponiveis a carros
caminhonete=[]#lista para saber quantas vagas disponiveis a
caminhonetes
cont=0
cont2=0
var=0 #não me chop para receber o valor da def de lucro
diario
soma=0 #não mecho soma é de somatorio dos lucros
sim=input("gostaria de saber o lucro do dia?s/n>:\n>")

```

O código inicia-se aparecendo ao usuário lhe questionando se gostaria de saber o lucro diário, dando as opções: sim ou não. Se sim, entra-se no laço de repetição. O mesmo apenas se se houver o acionamento da opção de não, quando lhe é perguntado se há mais clientes.

```

while sim=="s":
    dia=int(input("(1) (dia util)\n(2) (fim de semana)\n"))
    quem=int(input("(1) idoso\n(2) com deficiência\n(3) outros\n>"))

categoria=int(input("categoria:\n(1) carro\n(2) moto\n(3) caminhonete\n>"))
    numerohoras=int(input("horas permanência: "))
    var=lucrodiario(quem,numerohoras,dia) #aqui ele chama a def
    soma=soma+lucrodiario(quem,numerohoras,dia)
    sim=input("próximo cliente?s/n>:\n>")

```

Este laço servirá para a realização do somatório de todos os lucros obtidos (retornados pela função `lucrodiario`), a cada vez que retornamos dentro do laço, ou seja, enquanto o mesmo não é quebrado.

```

numerohoras=int(input("horas permanência: "))
var=lucrodiario(quem,numerohoras,dia)
soma=soma+lucrodiario(quem,numerohoras,dia)
sim=input("próximo cliente?s/n:\n>")

def lucrodiario(quem,numerohoras,dia):
    if dia==1:#se dia normal
        if quem==3:
            lucro=numerohoras*7#7 reais agora
        else:
            lucro=numerohoras*5#5 reais p idosos/defici
    else:#se fim de semana
        if quem ==3:
            lucro=numerohoras*9 #pra outros
        else:
            lucro=numerohoras*7#idosos/deficientes
    return lucro
print(soma)#após a quebra do laço ele apresenta o resultado

```

Tanto com a quebra do laço quanto a opção da primeira pergunta (que tem o intuito de questionar o main em relação a saber o lucro diário) for não, o programa irá dar a opção de realizar o cadastramento de horista ou de mensalista com opção 1 ou 2 usando if e else novamente.

```

agendamento=int(input("(1)cadastrar um horista \n(2)cadastrar um mensalista\n>"))
if agendamento==1:
    for v in range(1):
        chegada.append(str(input("placa: ")))
        dia=int(input("(1) (dia util)\n(2) (fim de semana)\n"))
        quem=int(input("(1)idoso\n(2)com deficiência\n(3)outros\n>"))
        categoria=int(input("categoria:\n(1)carro\n(2)moto\n(3)caminhonete\n>"))
        horaentrada=int(input("hora de entrada: "))
        horasaida=int(input("hora de saida: "))
        quanto=quantoapagar(horaentrada,horasaida,quem,dia)
        chegada.append(quem)
        chegada.append(horaentrada)
        chegada.append(horasaida)
        chegada.append(quanto)
        matriz.append(chegada)
        if categoria==1:
            carro.append(1)
        elif categoria==2:

```



```

        moto.append(1)
    else:
        caminhonete.append(1)
    pergunta=(str(input("gostaria de ver o numero de vagas
disponiveis?s/n:\n>")).lower())
    if pergunta=="s":
        pergunta2=int(input("(1)total de vagas\n(2)apenas
carro\n(3)apenas moto\n(4)apenas caminhonete\n>"))
        if pergunta2==1:
            #print(100-(len(moto)+len(carro)+len(caminhonete))) #numero
total de vagas fixo
            print(100-(len(matriz)))
        elif pergunta2==2:
            print(40-(len(carro)))
        elif pergunta2==3:
            print(20-(len(moto)))
        else:
            print(40-(len(caminhonete)))

    else:#para mensalistas
        horasaida=0
        varquanto=0
        quanto=0
        horaentrada=0
        cadastro2=[0,0,0]
        chegada2=[]
        matriz2=[]
        carro2=[]
        moto2=[]
        caminhonete2=[]

    for v in range (1):
        chegada2.append(str(input("placa: ")))
        dia=int(input("(1) (dia util)\n(2) (fim de semana)\n"))
        quem=int(input("(1)idoso\n(2)com deficiência\n(3)outros\n>"))
        categoria=int(input("categoria:\n(1)carro\n(2)moto\n(3)caminhonet
e\n>"))
        horaentrada=int(input("hora de entrada: "))
        horasaida=int(input("hora de saida: "))
        chegada2.append(quem)
        chegada2.append(horaentrada)
        chegada2.append(horasaida)
        chegada2.append(quanto)
        matriz2.append(chegada)
        if categoria==1:
            carro2.append(1)
        elif categoria==2:
            moto2.append(1)
        else:
            caminhonete2.append(1)

```

Funcionando de forma semelhante para ambas as opções, o código opera preenchendo uma lista e salvando em outra lista, para poder deste modo obter o total de clientes que deram entrada, e assim descontar de um número fixo de vagas para saber o quanto de vagas ainda há disponíveis, sendo o número de vagas para mensalistas independentemente do número de vagas para horistas.

Para apresentar este resultado o programa perguntará ao usuário se o mesmo deseja ver o número de vagas. Se sim, será perguntado em relação ao número de vagas totais, carro, moto ou caminhonete, como estabelecido, para realizar este cálculo é diminuído o valor fixo de vagas para cada situação do valor retornado quando usado o len (nome da lista).

Após isso, ou se a opção de ver número de vagas disponíveis não for, será encerrado o programa.

```
pergunta=(str(input("gostaria de ver o numero de vagas
disponiveis?s/n:\n>")).lower())
if pergunta=="s":
    pergunta2=int(input("(1)total de vagas\n(2)apenas carro\n(3)apenas
moto\n(4)apenas caminhonete\n>"))
    if pergunta2==1:
        #print(100-(len(moto)+len(carro)+len(caminhonete))) #numero
total de vagas fixo
        print(60-(len(matriz2)))
    elif pergunta2==2:
        print(30-(len(carro2)))
    elif pergunta2==3:
        print(15-(len(moto2)))
    else:
        print(15-(len(caminhonete2)))
```

4 CONCLUSÃO

No decorrer do desenvolvimento deste trabalho houve vários desafios a serem superados, pois com o estabelecimento de um tema em relação a uma empresa, tivemos que buscar maneiras de criarmos algo útil, que possa ser utilizado no dia-a-dia, algo que poderia satisfazer tal cliente. Além das dificuldades evidentes para relacionar e desenvolver os conhecimentos adquiridos ao longo da disciplina, principalmente em relação a matrizes.

Apesar das dificuldades citadas acima, o trabalho nos ampliou a visão do campo da programação, nos demonstrando a importância de um programa que substitua o uso manual de ferramentas, nos proporcionando mais praticidade e confiança nos resultados a serem obtidos. Além do programa ter sido criado de um modo mais simples em relação ao nosso objetivo, pode ele ser relativamente útil para nosso desenvolvimento tanto de equipe quanto de prática, até à formatação final.

5 REFERENCIA

Disponível em:< <http://pyscience-brasil.wikidot.com/python:python-oq-e-pq>>.
Acessado dia 26 de novembro de 2018, às 19h.