

Projeto final: Desafio Python

Aluno: João Vitor Pereira
Trilha Back-end

Engage

▶ Big Idea: Programação

Programação para desenvolver o projeto em questão.

▶ Questão Essencial: Como melhorar meus conhecimentos de programação?

Para formular uma resposta para essa questão essencial eu tive que fazer alguns testes após algumas pesquisas na internet, e a melhor forma que encontrei para melhorar meus conhecimentos foi: Fazendo e pesquisando. A forma que eu encontrei e que melhor deu certo foi tentar fazer o código completamente sozinho, aí anotar em um bloco de notas todas as minhas dúvidas na ordem em que iam surgindo. Após anotar, eu pesquisava todas e logo em seguida aplicava respondendo as questões essenciais das big ideias que surgiam com as dúvidas.

▶ Desafio: Desenvolver um programa com os tópicos da disciplina de Nivelamento de Programação.

▶ Requisitos nível intermediário (Legolas):

- Uso de bibliotecas de terceiros
- Entre 150 e 300 linhas de código

Nota

Requisitos nível intermediário (Legolas):

Uso de bibliotecas de terceiros • Entre 150 e 300 linhas de código

Motivo para escolha dessa categoria: eu realmente não sabia muito sobre programação orientada a objetos, nunca tinha feito uma interface e nem importado tantas bibliotecas.

Quando o projeto estava finalizado eu senti que o meu nível já era melhor do que eu esperava, porém durante o processo eu senti bastante dificuldade. Varias coisas estão “amadoras” nesse projeto, erros simples. Portanto, ainda não posso me considerar nível avançado.

Direcionamento

Questões guias

- ▶ Questões sobre o domínio que o software vai atender
- ▶ Questões sobre o que você não sabe de programação mas que será necessário para resolver o desafio
- Como fazer uma interface em python?
- Como manipular botões em python?
- Como fazer uma 'historia' seguir de maneira linear

Meu programa

Act: Fase de desenvolvimento do programa

- ▶ • Divida o desenvolvimento em etapas, por exemplo:
- ▶ • Identifique as funções do seu programa
- ▶ • Crie todas as funções identificadas, mas na implementação use um print com uma mensagem: “Função X em Construção”
- ▶ • Faça toda a navegação do seu programa utilizando as funções definidas anteriormente, conforme foi mostrado em aula
- ▶ • Implemente cada uma das funções em construção

Meu programa

Primeiramente, eu dividi as etapas após a definição da ideia inicial, então a tomada de decisão sobre a direção já foi uma das etapas, a primeira.

- ▶ Etapa 1: direção
- ▶ Etapa 2: esqueleto do projeto
- ▶ Etapa 3: implementação da interface
- ▶ Etapa 4: testes e melhoramento

Etapas do desenvolvimento do projeto

► Primeira etapa (direção/definição):

A primeira etapa foi a definição para a direção do projeto. Para isso eu abordei questões que queria incluir no meu projeto. A ideia principal era desenvolver uma interface para o programa. Eu não sei muito sobre programação orientada a objetos e isso seria necessário para fazer a interface que eu imaginei. Portanto, precisaria aplicar as metodologias de pesquisa para um melhor aprendizado

Então, após definir o que iria fazer, decidi começar fazendo a historia interativa com programação estruturada e após isso fazer a interface.

Questões guias:

- Como fazer uma interface em python? Utilizando o PyQt5
- Como manipular botões em python? Com programação orientada a objetos
- Como fazer uma 'historia' seguir de maneira linear? Sistema de contador.

Etapas do desenvolvimento do projeto

► Segunda etapa (esqueleto do projeto):

Essa etapa consistiu em fazer o programa estruturado. A ideia era fazer a historia, com inputs, sem usar funções nesse momento.

Escrevi o roteiro em um doc. no Word, e para cada uma ou duas linhas eu colocava como uma linha do texto em Sting em uma variável no python.

Coloquei as variáveis dinâmicas como nome e etc. Fiz um sistema de condicionais para definir os rumos da historia, com opções A e B que deveriam ser digitadas no input. Fiz um sistema de adicionar itens com um dicionário e por ultimo importei a biblioteca Random para aplicar resultados aleatórios.

Nessa parte eu não tive dificuldades, já sabia fazer tudo.

Agora era hora de pensar em como fazer a interface

```
#unicornio pode aparecer
import random
x = random.randint(1,5)

if x > 2:
    print("Ele começou a ouvir um barulho")
    print("escolheu um item para tentar se c
    # escolha de item -- mostra o inventario
    print("seu inventario:")
    for i in itens.keys():
        print(i, ":", itens[i])
    escolha_item = input('\033[31m' + "(eco

itemmao = []
#escolha de itens dentro do inventario
if escolha_item == "A":
    pair2 = itens["item1"]
    itemmao.append(pair2)
    for i in itemmao:
        print('\033[32m' + f"{nome} escol

elif escolha_itens == "B":
    pair2 = itens["item2"]
    itemmao.append(pair2)
```

```
print(Texto3)
nome = input('\033[31m' + 'Escreva o nome na caixa de texto: ' + '\033[0;0m')

Texto4 = '\033[32m'+f'E o homem respondeu: meu nome é {nome}'+'\033[0;0m' #verde
print(Texto4)

Texto5 = f"-Levante-se! (disse o guarda)"
print(Texto5)

Texto6 = f"-Você não pode perambular pelo reino dessa maneira, vou conduzi-lo para a c
print(Texto6)

idade = input('\033[31m' + 'Escreva a idade na caixa de texto: ' + '\033[0;0m')

Texto7 = '\033[32m'+f'E então o homem respondeu: eu tenho {idade} anos'+'\033[0;0m'
print(Texto7)

while jogar != "sim"

este2.py x
-Qual é o seu nome vassalo imundo? (perguntou o guarda)
Escreva o nome na caixa de texto: joao
E o homem respondeu: meu nome é joao
-Levante-se! (disse o guarda)
-Você não pode perambular pelo reino dessa maneira, vou conduzi-lo para a cadeia do castelo i
Escreva a idade na caixa de texto: |
Control  TODO  Python Packages  Python Console
```


Etapas do desenvolvimento do projeto

▶ Terceira etapa (esqueleto do projeto):

Essa etapa consistiu em fazer uma pesquisa sobre como funcionavam interfaces em python.

Abri o YouTube e vi que dava para fazer com o PyQt5, uma biblioteca para configurar interface. Vi também que existia um programa chamado QT designer, e poderia facilitar a manipulação dos objetos dentro do código. Após isso, baixei o QT designer e comecei a fazer a estrutura da interface, adicionei alguns botões outras ferramentas.

Em seguida, pesquisei como trabalhar para implementar funções para botões dentro do python. Caí em programação orientada a objetos (sabia somente o básico) e comecei a ver diversos exemplos. Em seguida, comecei a testar com outros programas um pouco mais simples, antes de implementar na minha interface previamente pronta.

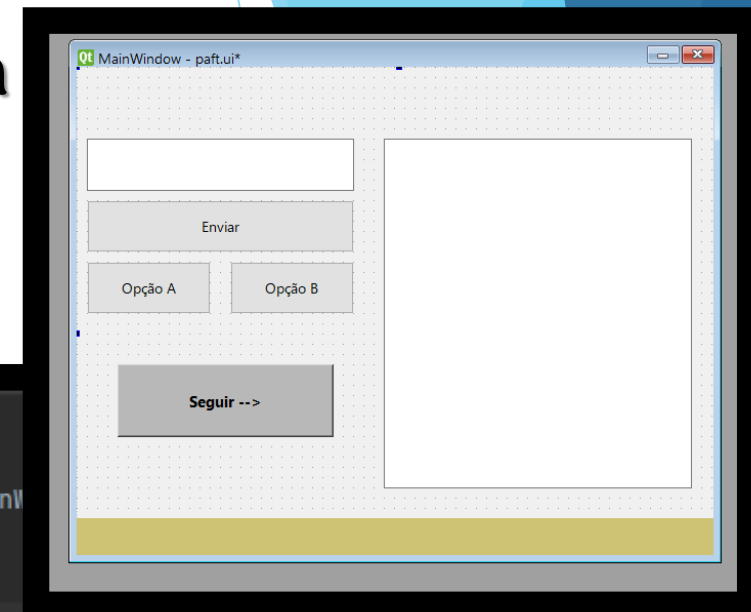
Após alguns teste eu entendi que precisava criar funções dentro da class, e na função construtora eu definia as ações dos botões

Interface básica

```
import sys
from paft import *
from PyQt5.QtWidgets import QMainWindow
import random
import time

class Paft(QMainWindow, Ui_MainWindow):
    def __init__(self, parent=None):
        super().__init__(parent)
        super().setupUi(self)

        #botao seguir
        self.pushButton.clicked.connect(self.segui)
        #botao para saber do contador
        #botaob
        self.btnA.clicked.connect(self.btna)
        #botaob
        self.btnB.clicked.connect(self.btb)
```



Class

Etapas do desenvolvimento do projeto

► Terceira etapa (esqueleto do projeto):

O maior problema agora era fazer a historia ir seguindo conforme eu apertava um botão, então eu desenvolvi o seguinte algoritmo:

Tem um contador, cada pagina dentro da interface é um numero do contador. Cada vez que o usuário aperta o botão para seguir, o contador recebe uma unidade a mais, então, se o valor do contador for igual ao valor da pagina o texto na label é alterado (mais tarde eu implementaria o sistema para voltar, que é exatamente o contrario).

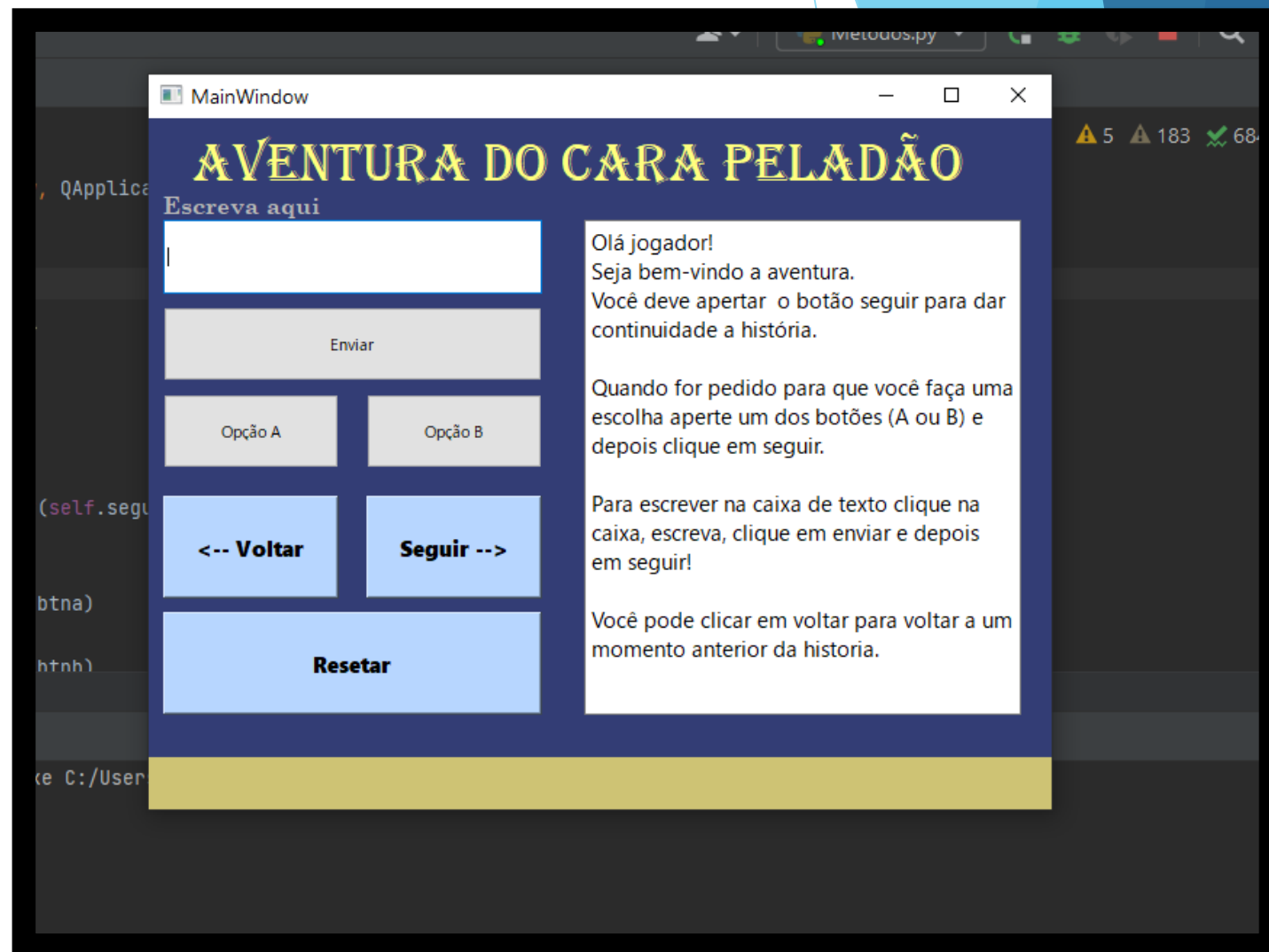
```
def seguir(self):  
    self.x = random.randint(1, 5)  
  
    self.contador = self.contador + 1  
  
    if self.contador == 0:  
        self.textEdit.setText("Para jogar aperte o botão 'seguir'")  
  
    elif self.contador == 2:  
        self.textEdit.setText(self.Texto1 + self.Espaco + self.Texto2)  
  
    elif self.contador == 3:  
        self.textEdit.setText(self.Texto3)  
  
    elif self.contador == 4:  
        self.textEdit.setText(self.Texto4)  
  
    #aqui começa a historia!!!
```

Etapas do desenvolvimento do projeto

▶ Quarta etapa (testes e melhoramento):

Agora eu precisaria definir a interface e adicionar algumas coisas a mais. Coloquei alguns botões a mais, um título para a historia (que é como ela começa, achei bem chamativo) e depois revisei algumas vezes.

O resultado final ficou bem funcional.



Final do projeto

Após terminado eu importei como um executável usando o pyinstaller

Para abrir o projeto é necessário entrar na pasta ProjetoPAFT > Build > Metodos > Metodos (que é um arquivo.exe)

`pythonProject\ProjetoPAFT\dist\Metodos`

Lá também tem o arquivo que usei para o código estruturado, ele é o Prototipo esqueleto

Ou da para ir pelo atalho

