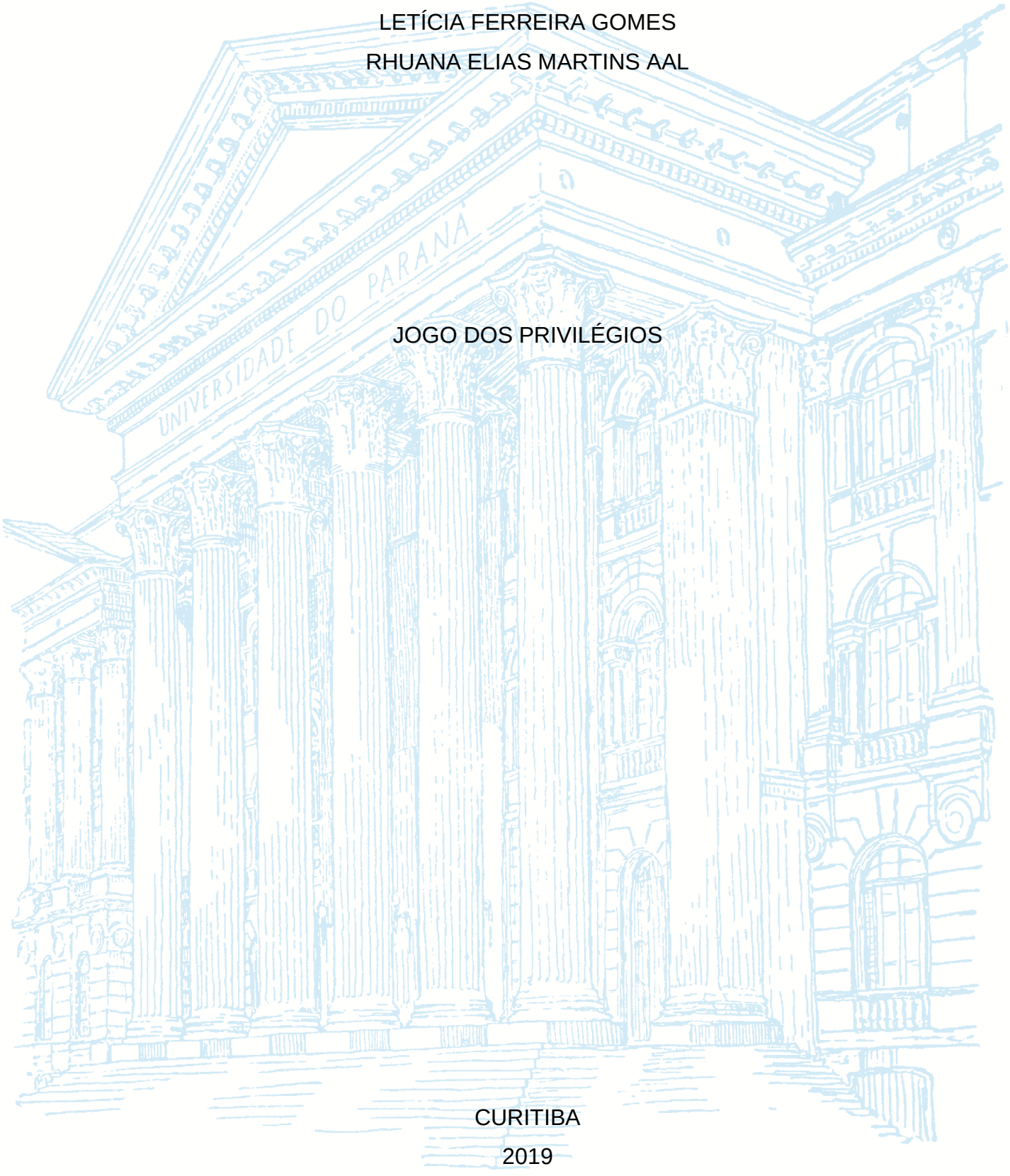
UNIVERSIDADE FEDERAL DO PARANÁ

DEBORAH SILVA BORGES
LETÍCIA FERREIRA GOMES
RHUANA ELIAS MARTINS AAL



JOGO DOS PRIVILÉGIOS

CURITIBA
2019

DEBORAH SILVA BORGES (GRR20160258)
LETÍCIA FERREIRA GOMES (GRR20160252)
RHUANA ELIAS MARTINS AAL (GRR20160297)

JOGO DOS PRIVILÉGIOS

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

Orientador: Prof. Jackson Antônio do Prado Lima

CURITIBA

2019

RESUMO

Esse trabalho é um relato do desenvolvimento de um programa desenvolvido em Python, sob o contexto da disciplina Fundamentos de Programação de Computadores, ofertada ao curso de Matemática da Universidade Federal do Paraná. O programa é intitulado “jogo dos privilégios”, pode participar mais de um jogador e tem como objetivo promover a reflexão da desigualdade presente na sociedade. O programa funciona da seguinte forma: afirmações aparecem na tela inicial do programa, de forma que cada jogador deve responder verdadeiro ou falso, de acordo com a sua realidade. Foram utilizados conceitos de classe, função, controles condicionais, laços de repetição e a importação de algumas bibliotecas.

Palavras-chave: Python. Jogo dos Privilégios. Fundamentos de Programação de Computadores.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Este trabalho é baseado na dinâmica “corrida de privilégios”, conhecida através de vídeos que carregam esse nome ou outros semelhantes. Esta corrida funciona da seguinte forma: são escolhidas, de forma aleatória, pessoas que vivem em diferentes realidades, essas pessoas ficam lado a lado formando uma linha, no contexto da corrida elas estariam na linha de partida; a seguir, aquele que estará guiando a corrida faz perguntas para todos e, a cada resposta afirmativa, é dado uma quantidade de passos para frente ou para trás. Essas perguntas feitas não têm relação com as escolhas próprias de cada pessoa e sim com as oportunidades que os participantes tiveram durante a vida. Essa dinâmica evidencia a existência da desigualdade que, infelizmente, existe em todas as sociedades e os privilégios que alguns acabam tendo sobre os demais.

Somos inocentes, ou até mesmo ignorantes, se pensarmos que todas as pessoas são iguais e têm as mesmas oportunidades na vida. Claro que todas as pessoas têm capacidade de correr atrás do que querem e lutar por aquilo que almejam, mas essa corrida e essa luta não são iguais para todos, alguns a iniciam mais a frente. As pessoas que saem à frente dos outros chamamos de privilegiadas, visto que o privilégio é a vantagem que se têm sobre outras pessoa.

Em *Python*, inspirado nos vídeos citados anteriormente, realizamos um programa para que as pessoas possam refletir sobre seus privilégios, ou a ausência deles, bem como sobre as tantas desigualdades que enfrentamos (social, racial, de gênero, etc) com perguntas semelhantes a da dinâmica apresentada anteriormente, e percebam a diferença e a desigualdade que existem entre as pessoas.

2 METODOLOGIA

Em nosso trabalho utilizamos duas bibliotecas: *os* e *random*, que iremos detalhar posteriormente. Além disso utilizamos as classes de programação, que não foi um conteúdo visto em sala de aula, por isso iremos descrever brevemente o que é e como construir uma classe, e apresentar quais foram as funções desenvolvidas no programa.

2.1 BIBLIOTECA OS

A biblioteca *os* foi utilizada com a intenção de “limpar” o programa, isto é, através do comando *os.system(“cls”)*, após a pergunta ser respondida não irá mais aparecer nem a pergunta nem a resposta, permitindo que tudo fosse “limpo” e surgisse a nova pergunta, ou seja, ela limpa a tela do programa em execução.

2.2 BIBLIOTECA RANDOM

A biblioteca *random* do Python é uma biblioteca muito conhecida e utilizada, pois ela permite gerar números aleatórios, embaralhar listas, sortear itens de sequências, em programas de jogos é possível criar inimigos de forma aleatória, e assim por diante.

Dentro da biblioteca random existem alguns métodos que podem ser utilizados para facilitar ainda mais o desenvolvimento de programas. Por exemplo, randrange(stop), permite você obter um número inteiro aleatório de 0 até antes do seu limite, ou seja, até antes do número que você colocar para substituir a palavra “stop”. Outro método é o randint(a,b), que fornece um número inteiro aleatório entre os números a e b. O método shuffle(x), embaralha a lista x. O método choice(seq) retorna uma escolha aleatória da sequência seq. Fora esses listados até aqui existem outros métodos que podemos usufruir a partir da biblioteca random.

Em nosso programa nós utilizamos a biblioteca random como na FIGURA 1, para gerar um ato de “sorte” ou de ”azar” na vida dos jogadores, a fim de mostrar que na vida estamos sujeitos a situações inesperadas, como ficar doente, que indica uma situação de “azar”, ou ser promovido no trabalho, que indica uma situação de “sorte”. Definimos o *random.choice([True, True, False, False, …, False])*,

considerando que se a escolha for *True* informa uma situação de sorte ou azar, e se a escolha for *False* não informa nenhuma dessas situações. Para que as respostas dos jogadores tenham mais influência do que os atos de sorte ou azar na pontuação existem mais opções *False* do que *True*.

FIGURA 1 – RANDOM NO PROGRAMA

```
elif random.choice([True, True, False,False,False,False,False,False,False,False]):
    r = random.randint(0,len(jogadores)-1)
    s = random.randint(0,1)
    print("\n")
    print(jogadores[r].nome,", na vida acontecem situações que não temos controle... \n ")
    if s == 1:
        print("*"*5, "SORTE!", "*"*5)
        t = random.randint(0,len(sorte)-1)
        print(sorte[t])
        jogadores[r].aumenta_pontos_progresso()


    elif s == 0:
        print("*"*5, "AZAR!", "*"*5)
        t = random.randint(0,len(azar)-1)
        print(azar[t])
        jogadores[r].diminui_pontos_progresso()
```

FONTE: Parte do programa.

Quando for mais de um jogador é necessário determinar qual deles irá receber a situação de sorte ou azar. Sendo assim, utilizando o *randint*, com os limites indo de zero até (*len(jogadores)-1)*, iremos escolher um dos jogadores de forma aleatória.

Utilizamos o *random.randint(0,1)* para determinar, de forma aleatória, se a situação será de sorte ou de azar, de modo que se a escolha for 1 indica uma situação de sorte, e se for 0 indica uma situação de azar.

Criamos uma lista para armazenar as diversas situações de azar e outra para armazenar as diversas situações de sorte, listadas no início do programa. Sendo assim utilizamos o *random.randint* para escolher uma dessas situações, já que os eventos na nossa vida acontecem assim, de forma aleatória e em momentos inesperados.

2.3 CLASSE

Para explicar o nosso programa precisamos diferenciar classe de objeto em Python. Utilizando uma analogia temos que a classe seria como uma receita de bolo, e o

objeto o bolo em si, resultado da receita. Então, tudo aquilo que podemos construir a partir da classe damos o nome de objeto.

Existem classes que já existem no Python, é o caso, por exemplo, de *str*, *int* e *Turtle*. Todavia, para resolver alguns problemas precisamos criar objetos de dados relacionados ao nosso problema, isto é, precisamos criar nossas próprias classes.

Vamos utilizar um exemplo, então para mostrar como criar uma classe. Suponha, então, que iremos criar uma classe *Pessoa*, de forma que seus atributos serão *nome* e *idade*. Na FIGURA 2 podemos ver como seria a implementação dessa classe na linguagem Python.

FIGURA 2 – CLASSE PESSOA

```
1  class Pessoa:
2      def __init__(self, nome, idade):
3          self.nome = nome
4          self.idade = idade
5
6      def setNome(self, nome):
7          self.nome = nome
8
9      def setIdade(self, idade):
10         self.idade = idade
11
12     def getNome(self):
13         return self.nome
14
15     def getIdade(self):
16         return self.idade
```

FONTE: DevMedia (2017).

As regras de sintaxe de uma classe são semelhantes a de outros comandos. Começamos com a palavra *class*, seguido pelo nome da classe e terminado com dois pontos, como é possível ver na Linha 1 do nosso exemplo (FIGURA 1).

A definição do construtor da classe é um método com o nome especial: __init__. Como todo método em Python a declaração começa com *def*, a seguir se tem, entre parênteses, os parâmetros, que inclui , em primeiro lugar, obrigatoriamente o parâmetro *self*, que é definido automaticamente para referenciar

o objeto recém formado e que precisa ser iniciado. É possível na Linha 2 da FIGURA 1.

Em nosso exemplo foram criados os métodos *get* e *set* de todos os atributos da classe Pessoa, com o objetivo de retornar ou modificar os atributos dessa classe, respectivamente.

Em nosso programa criamos a classe Jogador, onde os atributos iniciais são: nome, pontos e progresso, como é possível ver na FIGURA 3 a seguir.

FIGURA 3 - CLASSE JOGADOR

```
class Jogador:

    def __init__(self, nome):
        self.nome = nome
        self.pontos = 0
        self.progresso = "  "

    def aumenta_pontos_progresso(self):
        self.progresso = self.progresso + "#"
        self.pontos = self.pontos + 1
    def diminui_pontos_progresso(self):
        if self.pontos >0:
            self.pontos = self.pontos - 1
            self.progresso = "#" * self.pontos
```

FONTE: Acervo do programa sem os comentários.

Dentro dessa classe também foram criadas duas funções: aumenta_pontos_progresso e diminui_pontos_progresso.

- Aumenta_pontos_progresso: tem o objetivo de atualizar os pontos e o progresso dessa classe, de forma a aumentar a pontuação;
- Diminui_pontos_progresso: tem o objetivo de atualizar os pontos e o progresso dessa classe, de forma a diminuir a pontuação, se a mesma for maior que zero, ou faz permanecer nula, caso não tenha nenhuma pontuação.

2.4 OUTRAS CARACTERÍSTICAS DO PROGRAMA

Fora as funções que já foram apresentadas juntamente com a classe Jogador, em nosso programa também consta a função *imprime_barra()*, que não recebe parâmetro nenhum e tem o objetivo de apresentar a pontuação dos jogadores, como é possível observar na FIGURA 4, a seguir.

FIGURA 4 - FUNÇÃO IMPRIME_BARRA

```
def imprime_barra():
    print("-"*40)
    print("PONTUAÇÃO \n")
    for i in range(num_jog):
        print(jogadores[i].nome," ",jogadores[i].progresso)
        print("\n")
    print("-"*40)
```

FONTE: Acervo da pesquisa.

Essa função irá imprimir a pontuação de cada jogador, onde cada ponto é indicado pelo símbolo “#”, e é delimitada por uma linha, formada por “-”.

No restante do trabalho contamos com laços de repetição e controles condicionais, conceitos bem trabalhados na disciplina de Fundamentos de Programação de Computadores.

3 RESULTADOS

Esse programa, apesar de simples, tem a intenção de permitir que as pessoas possam jogar com seus colegas, com a quantidade que desejarem, e refletir a respeito do contexto no qual estão inseridos. Tem a intenção de fazer com que as pessoas entendam que, mesmo tão próximas, vivem realidades muito diferentes, e se tornem pessoas mais simpáticas, amigáveis, compreensivas e se coloquem mais no lugar do outro.

É um ótimo programa para ser inserido no contexto da sala de aula, fazendo com que os alunos, de diferentes idades, aprendam sobre empatia e, também, sobre programação.

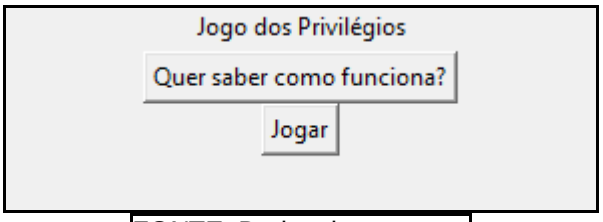
Ao realizar esse programa nos desafiamos a conhecer comandos, métodos, funções do Python com o intuito de deixar o programa mais simples para atingirmos o que desejávamos.

Classe nos auxiliou a criar o programa de forma mais prática, além das inúmeras listas que utilizamos, pois, ao invés de construir, por exemplo, um "*if*" para cada situação de azar, elencamos todos em uma lista e depois utilizamos um "*for*" para escolher os elementos, de forma que cada elemento indica uma possível situação de azar.

4 DIFICULDADES ENCONTRADAS

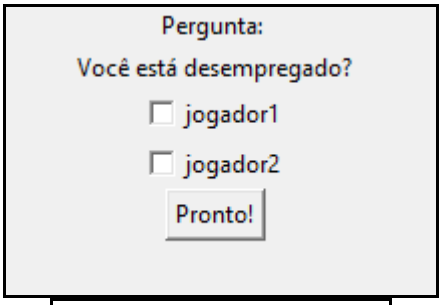
Para o jogo dos privilégios, inicialmente queríamos montar um jogo com a possibilidade de aparecer em janelas, com opções de botões para cada participante clicar na sua resposta. A princípio estávamos conseguindo desenvolver a estrutura base para criar janelas e botões, como é possível observar na FIGURA 5 e na FIGURA 6.

FIGURA 5 - TENTATIVA DE BOTÕES



FONTE: Dados das autoras.

FIGURA 6 - TENTATIVA DE JANELAS



FONTE: Dados das autoras.

Não estávamos conseguindo fazer o que desejávamos de maneira prática, as janelas não abriam como queríamos, muita das vezes abria uma janela para cada pergunta ou para cada botão. Relacionar o que desejávamos fazer com as janelas e os botões estavam tomando muito o nosso tempo, de forma que abrimos mão da utilização dos mesmos para realizar o jogo como queríamos no prazo.

5 SUGESTÕES DE TRABALHOS

Inicialmente queríamos propor um programa que permitisse desenhar fractais, de modo que fractais são figuras de extrema beleza e que obedecem um padrão, e são muito estudados na matemática, principalmente na área de Sistemas Dinâmicos. Além disso um fractal tem a característica de ter uma simples lei de formação, de forma que a cada iteração o todo vai se parecendo com partes menores do mesmo. Como por exemplo, o Triângulo de Sierpinski, o floco de neve de Koch, a esponja de Menger, o conjunto de Cantor e assim por diante.

Desejávamos desenhar um fractal de modo fosse possível escolher o número de iterações. Estávamos utilizando a biblioteca *Turtle* para desenhar o fractal. Infelizmente encontramos algumas dificuldades ao longo do programa, devido ao fractal de base caleidoscópica que foi o escolhido para representar. Entretanto, se bem desenvolvido, acreditamos que será um ótimo trabalho, além de conhecer um pouco mais sobre o que é um fractal.

Outra sugestão é aprimorar o nosso programa, colocando as janelas e os botões, ou construindo como um jogo de corrida, pois o jogo dos privilégios também é conhecido como “corrida dos privilégios”.

REFERÊNCIAS

BUZZFEED BRASIL. Descubra se você é privilegiado. **BuzzFeed**, 2015. Disponível em: <https://www.buzzfeed.com/br/clarissapassos/voce-eh-privilegiado?bfsource=bfocompareof>

CASA DO CÓDIGO. Apostila Python e orientação a objetos. **Caelum**. Disponível em: < https://www.caelum.com.br/apostila-python-orientacao-objetos/orientacao-a-objetos/#classes-e-objetos >

CASTRO, A. Caminhada do privilégio. **Alex Castro**, 2016. Disponível em: <https://alexcastro.com.br/caminhada-do-privilegio/ >

INSTITUTO IDENTIDADES DO BRASIL. Descubra o que é Jogo do Privilégio Branco. **Sim à igualdade racial.** Disponível em: <http://simaigualdaderacial.com.br/site/?mergulhe_no_tema=vantagem-racial-jogo-do-privilegio-branco>

LOUREIRO, M. A. Sobre méritos e privilégios: uma corrida por 100 dólares que viralizou. **Instituto Loureiro**, 2017. Disponível em: < http://institutoloureiro.com.br/meritos-privilegios/>

MILLER, B.; RANUM, D. Classes e Objetos - Fundamentos. Disponível em: <https://panda.ime.usp.br/pensepy/static/pensepy/13-Classes/classesintro.html>

SANTOS. Python Class: Como criar minha primeira classe em python. **DevMedia**, 2017. Disponível em: <https://www.devmedia.com.br/como-criar-minha-primeira-classe-em-python/38912>