

TESTE DE PERFORMANCE 2



Engenharia da Computação
Projeto em Arquitetura de Computadores,
Sistemas Operacionais e Redes
Kaio Henrique Silva da Cunha
Prof.: Alcione Dolavale

Fortaleza, CE
28/09/2021

Atividade 1

Descreva de maneira detalhada sobre o entregável.

R:

Nesse trecho o módulo pygame, que exibe a tela com as barras gráficas, e o psutil, que busca informações do computador, são importados. Esse passo é importante e deve ficar no início do código. Logo em seguida, é definida a fonte e a tela com largura e altura de 800 e 600, respectivamente. Outro importante passo, é a definição das variáveis de cores, representadas pelos 3 valores do modelo RGB. Essa variáveis serão utilizadas em seguida para exibir as barras gráficas.

```
kaio_henrique_PB_TP2.py ×
1  import pygame
2  import psutil
3
4  pygame.font.init()
5  font = pygame.font.Font(None, 32)
6
7  # Iniciando a janela principal
8  largura_tela = 800
9  altura_tela = 600
10 tela = pygame.display.set_mode((largura_tela, altura_tela))
11 pygame.display.set_caption("Uso de Recursos")
12 pygame.display.init()
13
14 preto = (0, 0, 0)
15 vermelho = (255, 0, 0)
16 branco = (255, 255, 255)
17 azul = (0, 0, 255)
18
19 # Mostar uso de memória
20 def mostra_uso_memoria():
21
```

Feito isso, é necessário criar o relógio, que vai buscar as informações do computador num tempo estipulado pela variável “cont”. O laço while verifica a

existência de um evento, como um clique no botão de fechar a janela, para continuar ou não com a exibição do gráfico. A segunda verificação confere a variável `cont` e chama a função `mostra_uso_memoria`. Nesse passo, é possível inserir também as outras funções.

```
68
69 # Cria relógio
70 clock = pygame.time.Clock()
71
72 cont = 60
73
74 terminou = False
75
76 while not terminou:
77     # Checar os eventos do mouse aqui:
78     for event in pygame.event.get():
79         if event.type == pygame.QUIT:
80             terminou = True
81     # Fazer a atualização a cada segundo:
82     if cont == 60:
83         mostra_uso_memoria()
84         cont = 0
85     # Atualiza o desenho na tela
86     pygame.display.update()
87     # 60 frames por segundo
88     clock.tick(60)
89     cont = cont + 1
90
91     # Finaliza a janela
92 pygame.display.quit()
93
```

Finalmente, é hora de criar as funções que vão buscar os dados do computador. Na função abaixo é definida a variável `mem`, que busca as informações da memória.

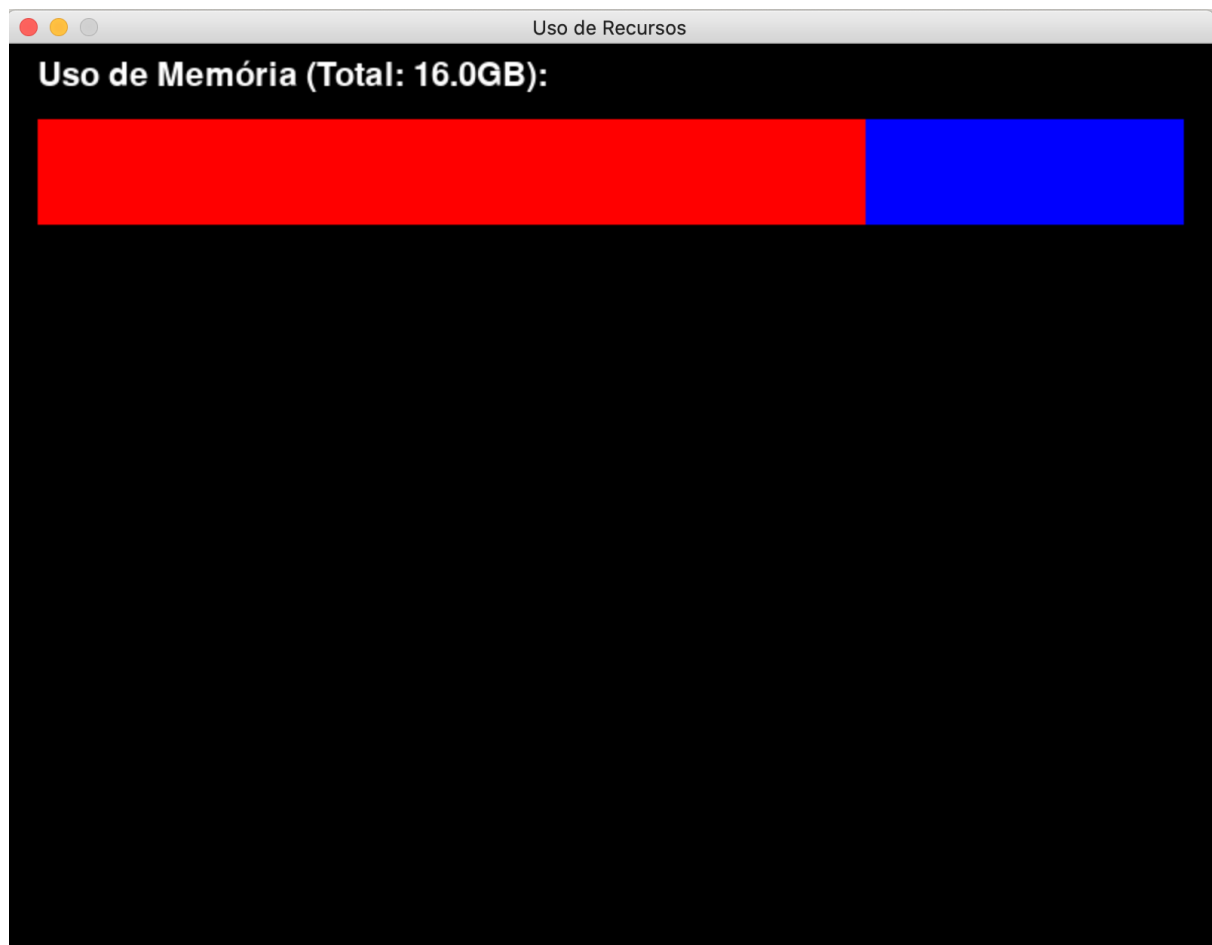
Depois, a largura da tela é definida e é colocado um fundo preto. A função `pygame.draw.rect` cria um retângulo azul com a largura anteriormente especificada. Em seguida, `larg` passa a ter o valor da porcentagem de memória em uso e utilizada como a largura do retângulo vermelho.

```
19 # Mostar uso de memória
20 def mostra_uso_memoria():
21
22     #busca o valor da memória em uso
23     mem = psutil.virtual_memory()
24     #define a largura da barra azul/vermelha
25     larg = largura_tela - 2*20
26     #define a cor de fundo da tela: preto
27     tela.fill(preto)
28     pygame.draw.rect(tela, azul, (20, 50, larg, 70))
29     #muda o valor da variável larg para representar a porcentagem utilizada
30     larg = larg*mem.percent/100
31     #cria a tarja vermelha, que vai depender do tamanho de larg
32     pygame.draw.rect(tela, vermelho, (20, 50, larg, 70))
33     #calcula e arredonda o uso de memória total
34     total = round(mem.total/(1024*1024*1024),2)
35     #exibe o valor utilizado de memória(total)
36     texto_barra = "Uso de Memória (Total: " + str(total) + "GB):"
37     #define o texto da barra com uma string que informa o total de memória utilizada
38     text = font.render(texto_barra, True, branco)
39     #exibe o texto devidamente posicionado e formatado
40     tela.blit(text, (20, 10))
41
```

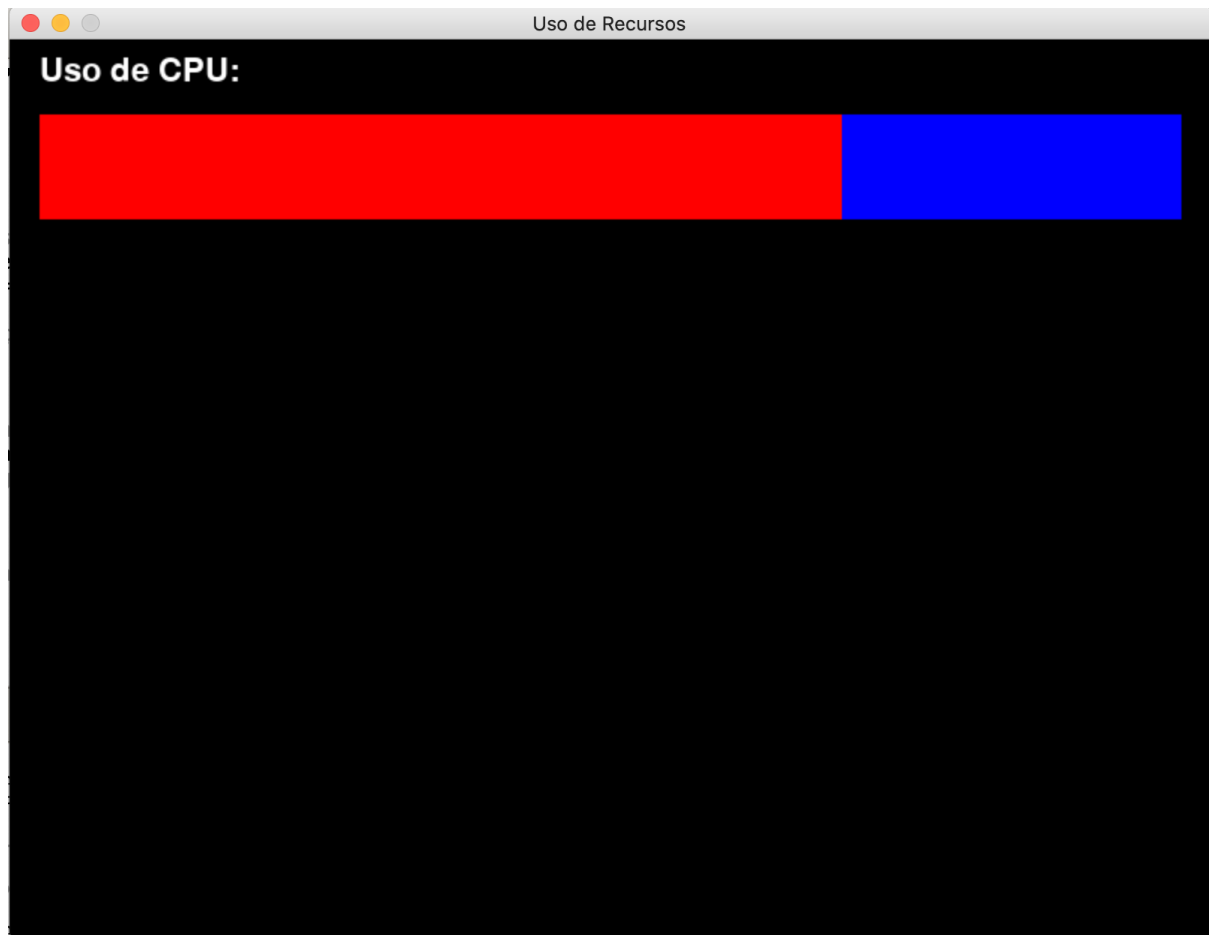
As outras funções variam ligeiramente:

```
40     tela.blit(text, (20, 10))
41
42 # Mostrar o uso de disco local
43 def mostra_uso_disco():
44
45     disco = psutil.disk_usage('.')
46     larg = largura_tela - 2*20
47     tela.fill(preto)
48     pygame.draw.rect(tela, azul, (20, 50, larg, 70))
49     larg = larg*disco.percent/100
50     pygame.draw.rect(tela, vermelho, (20, 50, larg, 70))
51     total = round(disco.total/(1024*1024*1024), 2)
52     texto_barra = "Uso de Disco: (Total: " + str(total) + "GB):"
53     text = font.render(texto_barra, True, branco)
54     tela.blit(text, (20, 10))
55
56 # Mostrar uso de CPU:
57 def mostra_uso_cpu():
58
59     capacidade = psutil.cpu_percent(interval=0)
60     larg = largura_tela - 2*20
61     tela.fill(preto)
62     pygame.draw.rect(tela, azul, (20, 50, larg, 70))
63     larg = larg*capacidade/100
64     pygame.draw.rect(tela, vermelho, (20, 50, larg, 70))
65     text = font.render("Uso de CPU:", True, branco)
66     tela.blit(text, (20, 10))
67
68
```

A exibição dos resultados ficam da seguinte forma:







A função para buscar o IP não funcionou localmente mesmo tentando vários tipos diferentes de redes, além de eth0. Por isso, a fiz no Google Colab.

```
#rede
import psutil

dic_interfaces = psutil.net_if_addrs()
print(dic_interfaces['eth0'][0].address)
```

172.28.0.2

Atividade 2

Descreva de maneira teórica sobre o IP e porque se faz necessária a utilização dos mesmos para conexões com outros dispositivos.

R: Os IPs são utilizados para identificar dispositivos numa rede e são fornecidos pelos serviços provedores de internet.

Referências

Material da disciplina no Moodle. Etapa 2. Acesso em ago. 2021.