

Infnet

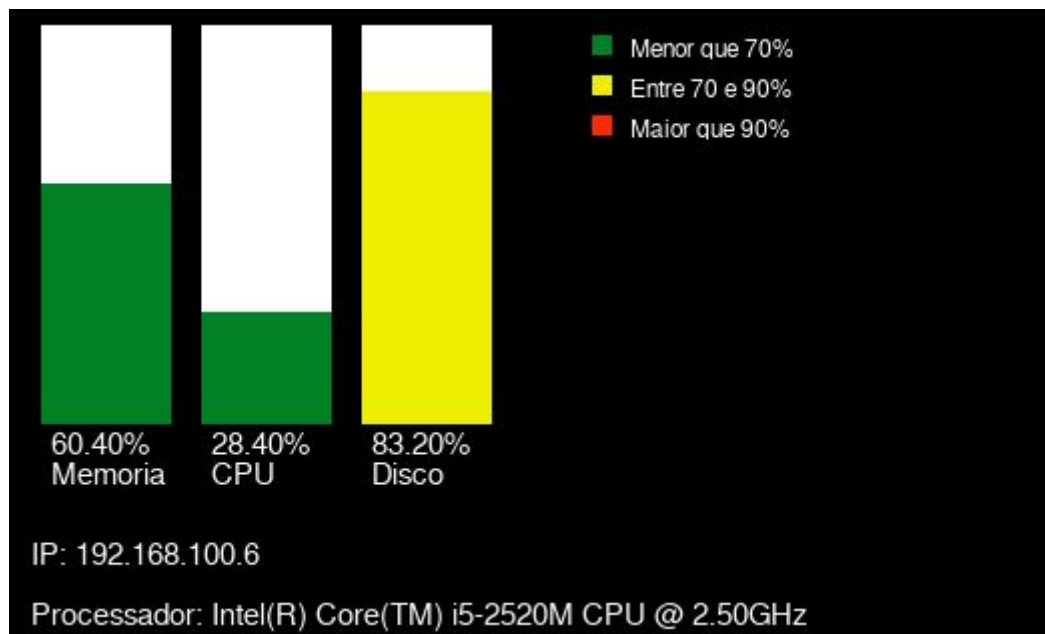
Engenharia de Software

Projeto de Bloco - Arquitetura de
Computadores, Sistemas Operacionais e
Redes

Rodrigo Pinto Coelho Gomes
Prof: Adriano

TP2

1 - O projeto foi criado com a intenção de monitorar a utilização do Disco, CPU e memória do computador. Além de informar o endereço IP da máquina.



2-

Módulos e ferramentas:

- **pygame** - criar a interface gráfica para a visualização.
- **psutil** - listar informações sobre o sistema computacional
- **platform** - listar informações do sistema operacional
- **time** - biblioteca para trabalhar com tempo (usar o sleep)

```
import platform
import psutil
import pygame, sys
from pygame.locals import *
import time
```

3- Fui capaz de obter as informações da memória, CPU e Disco graças a biblioteca do Python chamada psutil.

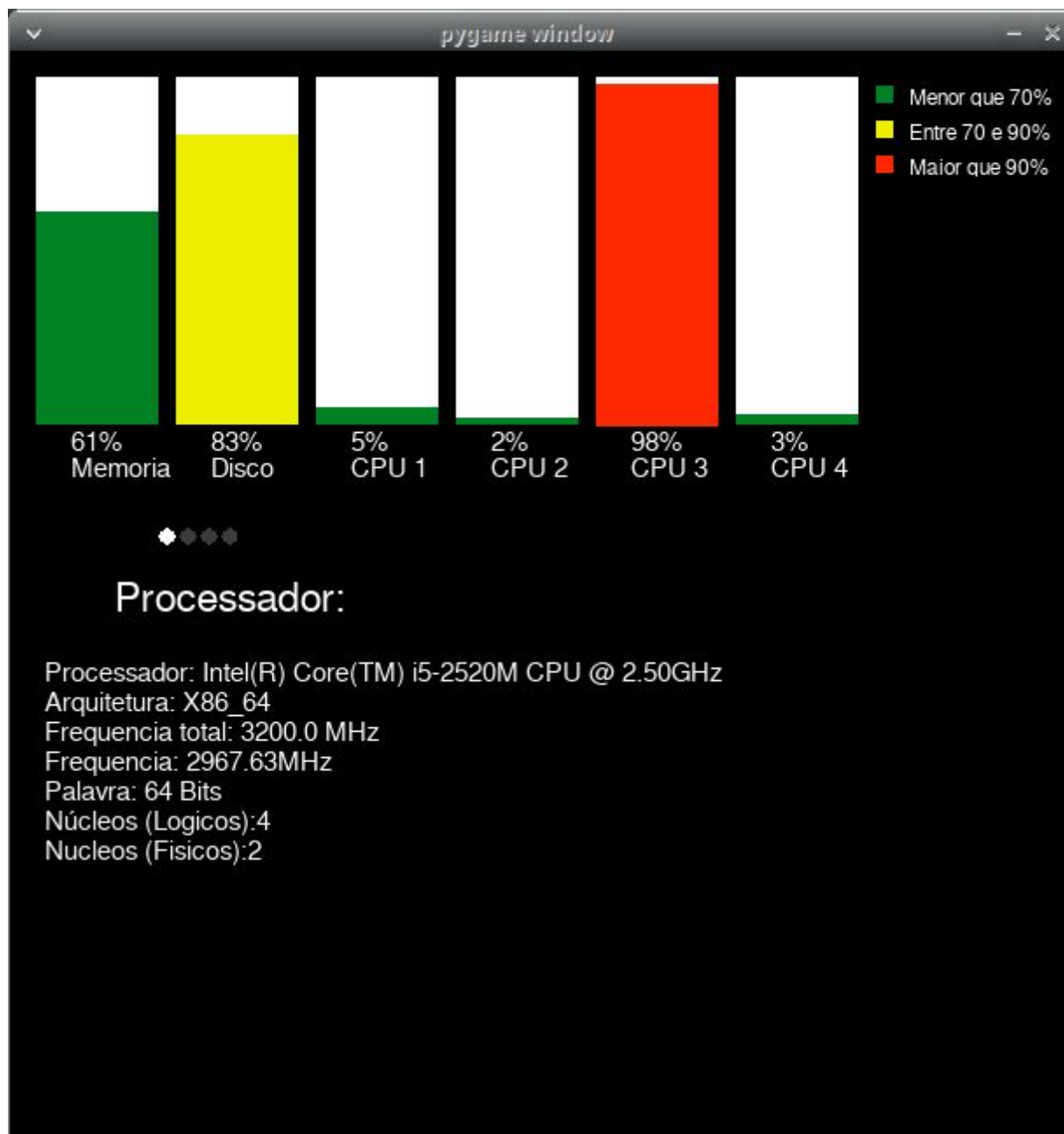
```
pct_memoria = psutil.virtual_memory().percent
pct_cpu = psutil.cpu_percent()
pct_disco = psutil.disk_usage('.').percent
```

4 - IP significa Internet Protocol. É um número associado a cada aparelho conectado a uma rede para comunicação. An **IP address** serves two main functions: host or network interface identification and location addressing. Usamos também o psutil para coletar dados do IP. Porém a chave de rede varia de acordo com o sistema operacional, por isso devemos checar o sistema operacional para acessarmos a chave correta.

```
rede=''
sistema = platform.system()
if sistema == 'Linux':
    rede='wlp3s0'
elif sistema == 'Windows':
    rede='wlan0'
else:
    rede='eth0'

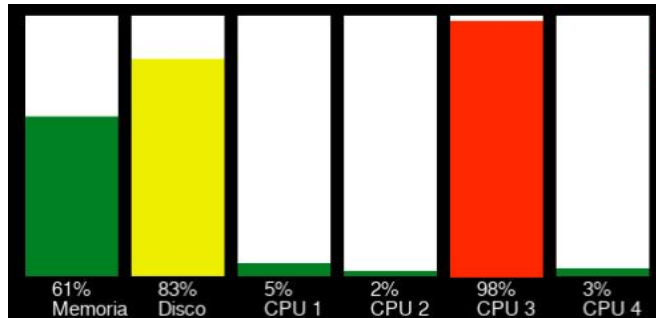
ip = 'IP: ' + psutil.net_if_addrs()[rede][0].address
```

TP3



QUESTÃO 1:

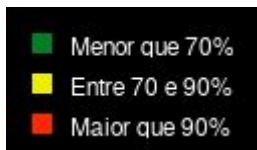
Barras:



As barras de memória e de disco foram colocadas no começo pois elas são fixas. Em seguida as barras de CPU aparecem de acordo com o número de CPUs que o computador atreves da funcao `psutil.cpu_count()`. Com isso, o número de CPUs foi considerado para fazer a largura da tela. Foram feitas através de uma função que recebe o nome, a porcentagem e o index da barra para que o alinhamento de acordo com o index possa ser feito corretamente. As barras de memoria e de Disco são fixas, as de cpus variam de acordo com o computador, foi feito um for nas cpus para desenhar todas.

```
desenhar_barra('Memoria',pct_memoria/100,0)
desenhar_barra('Disco',pct_disco/100,1)
for i,cpu in enumerate(psutil.cpu_percent(interval=1,percpu=True)):
    desenhar_barra('CPU '+str(i+1),cpu/100,2+i)
```

Legenda:



Somente textos com quadrados do lado com a respectiva cor, vale lembrar que o tamanho dessa superfície também foi considerada quando fazer o cálculo da largura da tela

```
def desenhar_legenda():

    def desenhar_legenda_unica(cor,label,cont):
        #Seta a font e tamanho
        myfont = pygame.font.SysFont("Arial", 12)
        #Seta a surface
        surface = pygame.surface.Surface((TAMANHO_LEGENDA+10, 12))
        #Seta o texto
        text= myfont.render(label, 1, WHITE)
        #Desenha o quadrado da cor
        pygame.draw.rect(surface,cor,(0,0,10,10))
        #Escreve o texto
        surface.blit(text, (19 , 0))
        #Desenha a surface na tela
        DISPLAY.blit(surface, ( LEGENDA_X, 20+cont*20 ))

    desenhar_legenda_unica(GREEN,'Menor que 70%',0)
    desenhar_legenda_unica(YELLOW,"Entre 70 e 90%",1)
    desenhar_legenda_unica(RED,'Maior que 90%',2)
```

Informações:

Processador:

Processador: Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz
Arquitetura: X86_64
Frequencia total: 3200.0 MHz
Frequencia: 3065.47MHz
Palavra: 64 Bits
Núcleos (Logicos):4
Nucleos (Fisicos):2

As informações foram feitas através do módulo psutil e cpuinfo. Cada informação foi colocada em uma lista de acordo com a aba e cada aba foi colocada em uma lista com todas as abas assim podemos chamar a função de desenhar as informacoes dinamicamente. Algumas das informações variam de acordo com o SO, com isso e criado uma lista principal (informacoes que todos os SOs contém) e uma lista secundária (de acordo com cada SO). E no final elas são concatenadas.

```
#Memoria
mem = psutil.virtual_memory()
memoria_total="Memoria Total:"+ inGB(mem.total)+ " GB"
memoria_disponivel="Memoria Disponivel:"+ inGB(mem.available)+ " GB"
memoria_livre="Memoria Livre:"+ inGB(mem.free)+ " GB"
memoria_usada="Memoria Usada:"+ inGB(mem.used)+ " GB"

net= psutil.net_if_addrs()[rede][array]
ip= 'IP: ' + net.address
netmask='NetMask: ' + net.netmask
family='Family: ' + str(net.family)
ptp='Ptp: ' + str(net.ptp)
memoria_os=[]
net_os=[]
if sistema == 'Linux':
    memoria_buffers="Buffers:"+ inGB(mem.buffers)+ " GB"
    memoria_cached="Cached:"+ inGB(mem.cached)+ " GB"
    memoria_shared="Compartilhada:"+ inGB(mem.shared)+ " GB"
    memoria_slab="Slab:"+ inGB(mem.slab)+ " GB"
    memoria_ativa="Memoria Ativa:"+ inGB(mem.active)+ " GB"
    memoria_inativa="Memoria Inativa:"+ inGB(mem.inactive)+ " GB"
    memoria_os=[memoria_buffers,memoria_cached,memoria_shared,memoria_slab,memoria_ativa,memoria_inativa]
    net_os=['BroadCast: '+net.broadcast]
elif sistema == 'Windows':
    print("nenhum extra no windows")
elif sistema == 'Darwin':
    memoria_ativa="Memoria Ativa:"+ inGB(mem.active)+ " GB"
    memoria_inativa="Memoria Inativa:"+ inGB(mem.inactive)+ " GB"
    memoria_wired="Wired: "+inGB(mem.wired)+ " GB"
    memoria_os=[memoria_ativa,memoria_inativa,memoria_wired]
    net_os=['BroadCast: '+net.broadcast]
```

Indexação:



A indexação foi feita capturando os cliques da seta esquerda, direita e barra de espaço. Definimos uma variável index que pode ser aumentado ou diminuído seguindo a regra maior que zero para poder diminuir o index e menor que 3 para poder aumentar o index, o index=4, a barra especial. Para acessarmos o resumo salvamos uma cópia do index e colocamos o index=4. Assim podemos usar a cópia do index para retornar para a respectiva aba quando ativamos e desativamos o resumo com a barra de espaço. A bola branca significa que a respectiva aba que está selecionada. As 4 juntas são o resumo.

```
processador_info=['Processador:',nome,arquitetura,frequencia_total,frequencia,bits,cpuscount,cpuscount]
memoria_info=['Memoria: ', memoria_total,memoria_disponivel,memoria_usada,memoria_livre]+memoria_os
disco_info=['Disco:',disco_total,disco_usado,disco_livre]
net_info=['Rede:',ip,netmask,family,ptp]+net_os
resumo_info=['Resumo: ',nome,cpuscount,memoria_total,memoria_disponivel,memoria_usada,memoria_livre,d
infos=[processador_info,memoria_info,disco_info,net_info,resumo_info]
```

QUESTÃO 2:

Existem diversos tipos de arquitetura diferentes de CPUs, podemos citar como exemplo as cpus x86 e ARM. A diferença começa na tecnologia empregada para a construção desses dois tipos de processadores, sendo que o x86 utilizam técnicas que permitem uma maior velocidade na execução das tarefas, enquanto por outro lado os processadores baseados em ARM são construídos de acordo com RISC (Reduced Instruction Set Computer), que como o nome já diz, tem como objetivo serem mais simples.

QUESTÃO 3:

Palavra é a unidade natural de informação utilizada em cada processador, sendo que cada processador trabalha com seu tamanho de palavra fixo e definido, por exemplo processadores de 32 bits e 64 bits, a palavra é usada para realizar a comunicação entre a memória principal e a CPU e também define qual quantidade de bits que serão processador por vez pela CPU.

QUESTÃO 4:

Núcleo físicos são as partes realmente existentes fisicamente na CPU para a realização dos processos, enquanto os núcleos lógicos são partes virtuais criadas para aumentar a velocidade de processamento das tarefas, os núcleos lógicos emulam a existência de uma maior quantidade de núcleos do que realmente existem no computador. Os núcleos lógicos são criados para ajudar os núcleos físicos quando esses estão sobrecarregados, eles recebem as tarefas a serem processadas e executam como se fossem núcleos físicos.