



Disciplina:	Sistemas Operacionais		
Aluno:	Pedro Moraes Mendes	R.A.:	2303470
Email:	pmendes@alunos.utfpr.edu.br		

RELATÓRIO DO DASHBOARD

1. COMO EXECUTAR O PROJETO

O projeto pode ser executado, simplesmente compilando o arquivo `app.py`, contanto que todos os outros arquivos `.py` necessários estejam no diretório. Um exemplo de execução é:

```
> python3 app.py.
```

Para acessar a página do navegador, coloque o endereço que será mostrado quando executar o arquivo `app.py`, que geralmente será no endereço `http://127.0.0.1:8050`

2. PÁGINA INICIAL

A página inicial se trata da página que exibe informações sobre os processos. No topo, há uma breve descrição do projeto. Abaixo, estão as tabs, que servem para separar por páginas cada conteúdo (Processos, Hardware, Sistema de Arquivos e Terminal). Nota-se que há um radio button logo abaixo das tabs de páginas. Elas servem para parar a atualização de dados, e retomar, caso tenha parado. Por padrão, a tab selecionada é a de processos.



Figura 1 - Topo da página principal

3. PROCESSOS

Na tab de processos, encontramos primeiramente uma tabela, que mostra a saída do comando `top` no terminal, que foi parseado e transformado em um dataframe do pandas para alimentar a tabela. A tabela mostra as principais informações do `top`, onde cada coluna pode ser ordenada por ordem crescente/descrescente utilizando as setas mostradas em (1), e os resultados são paginados, podendo navegar entre eles usando os controles em (2),



como mostra na Figura 2 a seguir:

(1)

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPO+	COMANDO
2851	pedromm	20	0	3220960	211216	95496	S	37.5	10.4	0:46.91	firefox
1836	pedromm	20	0	239264	46424	42716	S	25	2.3	0:05.13	Xwayland
1325	pedromm	20	0	4934112	175864	61740	S	18.8	8.7	1:14.76	gnome-shell
5389	pedromm	20	0	21972	4264	3444	R	6.2	0.2	0:00.01	top
1	root	20	0	168080	9716	5240	S	0	0.5	0:04.79	systemd
2	root	20	0	0	0	0	S	0	0	0:00.04	kthreadd
3	root	0	-20	0	0	0	I	0	0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0	0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0	0	0:00.00	netns
7	root	0	-20	0	0	0	I	0	0	0:00.00	kworker/0:0H-events_highpri

(2)

Figura 2 - Tabela de saída do comando top, mostrando processos em execução

Abaixo da tabela, existem 2 gráficos. O primeiro mostra o consumo de CPU por processo, exibindo os 10 primeiros, e o segundo mostra o consumo de memória por processo, também sendo os 10 primeiros. O eixo X mostra o nome dos processos e o eixo Y mostra a % de consumo, como ilustra a Figura 3:



Figura 3 - Gráficos de consumo de CPU e memória por processo.

4. HARDWARE

Selecionando a tab de hardware, iremos encontrar uma seção com as informações do sistema, como mostra a Figura 4:



Informações do sistema

SISTEMA: Linux	VERSÃO: #62-Ubuntu SMP
NODE: pedromm-VirtualBox	MÁQUINA: x86_64
RELEASE: 5.15.0-56-generic	PROCESSADOR: x86_64

Figura 4 - Informações do sistema.

Ao lado, poderá encontrar uma seção mostrando o uso de CPU, mostrando o consumo em % de cada core, além do consumo total, como ilustrado na Figura 5 à seguir:

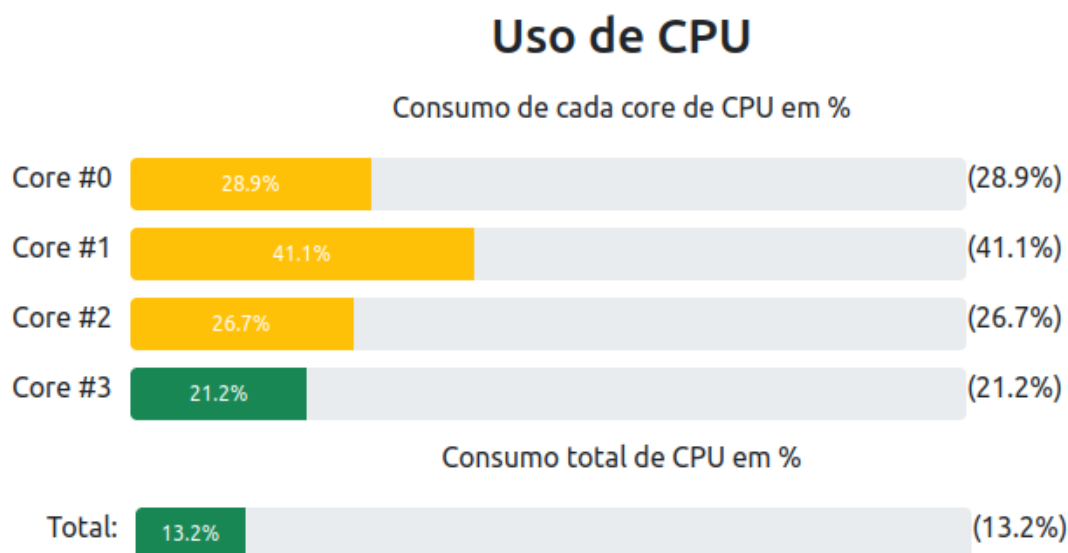


Figura 5 - Consumo de CPU total e de cada core.

Por último, são mostradas as informações referente a memória principal e memória de SWAP, tais como espaço total, disponível, usado e %, além de uma barra ilustrativa, como mostra a Figura 6:

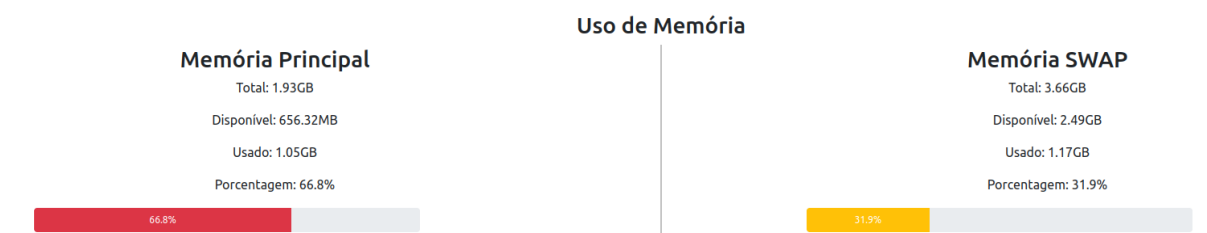


Figura 6 - Seção de consumo de memória.



Por fim, é importante observar que do mesmo modo que na tab de processos, pode-se parar a atualização, bem como retomar. Ademais, as cores dos gráficos seguem a legenda:

- Verde: até 25%
- Amarelo: de 25% até 50%
- Vermelho: Mais que 50%

5. SISTEMA DE ARQUIVOS

Essa página consiste em tabelar todas as partições do sistema, mostrando seu nome, local onde está montado, o sistema de arquivos, espaço total, usado, livre e %, respectivamente. Esta página é estática e não atualiza informações. A figura 7 mostra a tabela:

Sistemas de arquivos e Partições

Dispositivo	Mount	Sistema de Arquivos	Espaço Total	Usado	Livre	%
/dev/sda3	/	ext4	33.66GB	21.73GB	10.18GB	68.1
/dev/loop1	/snap/bare/5	squashfs	128.00KB	128.00KB	0.00B	100
/dev/loop0	/snap/core20/1695	squashfs	63.25MB	63.25MB	0.00B	100
/dev/loop2	/snap/core20/1738	squashfs	63.25MB	63.25MB	0.00B	100
/dev/loop3	/snap/firefox/2154	squashfs	237.62MB	237.62MB	0.00B	100
/dev/loop4	/snap/firefox/2088	squashfs	237.62MB	237.62MB	0.00B	100
/dev/loop6	/snap/gtk-common-themes/1535	squashfs	91.75MB	91.75MB	0.00B	100
/dev/loop5	/snap/gnome-3-38-2004/115	squashfs	346.38MB	346.38MB	0.00B	100
/dev/loop7	/snap/gnome-3-38-2004/119	squashfs	346.38MB	346.38MB	0.00B	100
/dev/loop8	/snap/hunspell-dictionaries-1-7-2004/2	squashfs	37.12MB	37.12MB	0.00B	100
/dev/loop10	/snap/snap-store/599	squashfs	46.00MB	46.00MB	0.00B	100
/dev/loop9	/snap/snap-store/638	squashfs	46.00MB	46.00MB	0.00B	100
/dev/loop11	/snap/snapd/17883	squashfs	49.62MB	49.62MB	0.00B	100
/dev/loop12	/snap/snapd-desktop-integration/14	squashfs	384.00KB	384.00KB	0.00B	100
/dev/sda3	/var/snap/firefox/common/host-hunspell	ext4	33.66GB	21.73GB	10.18GB	68.1
/dev/loop13	/snap/snapd-desktop-integration/43	squashfs	384.00KB	384.00KB	0.00B	100
/dev/sda2	/boot/efi	vfat	511.96MB	5.24MB	506.73MB	1

Figura 7 - Tabela de partições.

6. TERMINAL

Nesta página, há um pseudo terminal, onde há um campo para digitar um comando, e uma textarea para a saída do comando digitado. É **imprescindível** desativar a atualização da página no radio button mencionado anteriormente para executar os comandos.

Ressalto também que os comandos tem efeito único, ou seja, não é possível executar um comando de navegação e trocar de diretório, salvo se utilizar comandos concatenados, exemplo: `cd..; ls`. A figura 8 ilustra um exemplo da execução do comando `ls -l`.



Terminal de comandos

Simulador de um terminal que processa comandos bash.

Entrada de comandos

ls -l

Saída de comandos

```
total 52
-rw-rw-r-- 1 pedromm pedromm 2543 dez 12 15:57 app.py
-rw-rw-r-- 1 pedromm pedromm 106 dez 12 14:14 bash.py
-rw-rw-r-- 1 pedromm pedromm 1718 dez 12 16:05 data_table.py
-rw-rw-r-- 1 pedromm pedromm 1431 dez 12 16:13 filesystem.py
-rw-rw-r-- 1 pedromm pedromm 6506 dez 3 23:05 hardware.py
drwxrwxr-x 2 pedromm pedromm 4096 dez 12 16:13 __pycache__
-rw-rw-r-- 1 pedromm pedromm 730 dez 12 16:08 read_top.py
-rw-rw-r-- 1 pedromm pedromm 1248 dez 3 23:19 terminal.py
-rw-rw-r-- 1 pedromm pedromm 14781 dez 13 15:48 top-output.csv
```

Figura 8 - Terminal de comandos.

7. ENTRADA E SAÍDA

Como pedido pelo professor, deveríamos explicar como implementaríamos a parte de IO do sistema operacional. Uma ideia seria buscar e parsear as informações do arquivo /proc/ioports, e mostrar em uma tabela ou lista, as faixas de endereço reservadas para cada dispositivo que aparece no arquivo.

8. BIBLIOTECAS UTILIZADAS

As bibliotecas utilizadas foram:

- dash
- dash_bootstrap_components
- psutil
- plotly.graph_objects
- pandas
- os