



PSutil

Live de Python # 206



1. PSutil

Um pouco sobre a biblioteca

2. Utilitários do sistema

CPU, Memória, sensores, rede e disco

3. Processos

Analisando processos ativos

4. Dashboard

Vamos criar um mini dashboard



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3



Acássio Anjos, Ademar Peixoto, Alexandre Harano, Alexandre Souza, Alexandre Takahashi, Alexandre Villares, Alex Lima, Alynne Ferreira, Alysson Oliveira, Ana Carneiro, Ana Padovan, Andre Azevedo, André Rocha, Aquiles Coutinho, Arnaldo Turque, Bloquearsites Farewall, Bruno Barcellos, Bruno Freitas, Bruno Guizi, Bruno Oliveira, Bruno Ramos, Caio Nascimento, César Almeida, Christiano Moraes, Clara Battesini, Daniel Freitas, Daniel Haas, Danilo Segura, Dartz Dartz, David Kwast, Delton Porfiro, Dhyeives Rodovalho, Diego Farias, Diego Guimarães, Dilenon Delfino, Dino Aguilar, Douglas Bastos, Douglas Braga, Douglas Martins, Douglas Zickuhr, Eli Júnior, Emerson Rafael, Érico Andrei, Eugenio Mazzini, Euripedes Borges, Evandro Avellar, Everton Silva, Fabio Barros, Fábio Barros, Fabio Castro, Fábio Thomaz, Felipe Rodrigues, Fernanda Prado, Fernando Rozas, Fernando Sousa, Flávio Meira, Flavkaze Flavkaze, Franklin Silva, Gabriel Barbosa, Gabriel Simonetto, Geandreson Costa, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Gustavo Dettenborn, Gustavo Suto, Heitor Fernandes, Henrique Junqueira, Igor Taconi, Ismael Ventura, Israel Gomes, Italo Silva, Jair Andrade, Jairo Lenfers, Janael Pinheiro, Jean Marcio, João Lugão, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jônatas Silva, Jorge Plautz, Jose Mazolini, Juan Gutierrez, Juliana Machado, Julio Silva, Kaio Peixoto, Kaneson Alves, Leandro Botassio, Leandro Miranda, Leonardo Cruz, Leonardo Mello, Leonardo Nazareth, Lucas Adorno, Lucas Mello, Lucas Mendes, Lucas Oliveira, Lucas Polo, Lucas Teixeira, Lucas Valino, Luciano Silva, Luciano Teixeira, Luiz Junior, Luiz Lima, Luiz Paula, Maiquel Leonel, Marcelino Pinheiro, Marcelo Matte, Márcio Martignoni, Marco Mello, Marcos Gomes, Marco Yamada, Maria Clara, Marina Passos, Mario Deus, Mateus Lisboa, Matheus Silva, Matheus Vian, Mirian Batista, Murilo Andrade, Murilo Cunha, Murilo Viana, Natan Cervinski, Nicolas Teodosio, Osvaldo Neto, Otávio Barradas, Patricia Minamizawa, Patrick Felipe, Paulo Braga, Paulo Tadei, Pedro Duarte, Pedro Henrique, Pedro Pereira, Peterson Santos, Priscila Santos, Rafael Lopes, Rafael Rodrigues, Rafael Romão, Ramayana Menezes, Reinaldo Silva, Renan Moura, Renato Veirich, Riverfount Riverfount, Robson Maciel, Rodrigo Brandao, Rodrigo Ferreira, Rodrigo Freire, Rodrigo Junior, Rodrigo Vaccari, Rodrigo Vieira, Rogério Sousa, Ronaldo Silva, Rui Jr, Samanta Cicilia, Sara Selis, Thalysson Bogéa, Thiago Araujo, Thiago Borges, Thiago Bueno, Thiago Curvelo, Thiago Moraes, Thiago Oliveira, Thiago S, Thiago Souza, Tiago Minuzzi, Tony Dias, Victor Wildner, Vinícius Bastos, Vinicius Figueiredo, Vítor Gomes, Vitor Luz, Vlademir Souza, Vladimir Lemos, Walter Reis, Wellington Abreu, Wesley Mendes, William Alves, Willian Lopes, Wilson Neto, Yury Barros



Obrigado você



A biblioteca

PS
util



Criada por **Giampaolo Rodola**, @giampaolo, em 2009. PSutil é uma abreviação para "**Process and System**" utilities.

Uma biblioteca multiplataforma pra obter informações sobre Processos e utilização do sistema operacional.

Uma implementação em python de diversos programas de linha de comando dos sistemas UNIX. ps, top, isof, netstat, ifconfig, who, e etc..

Suporta: Linux, Windows, Mac OS, BSDs, Solaris e AIX.

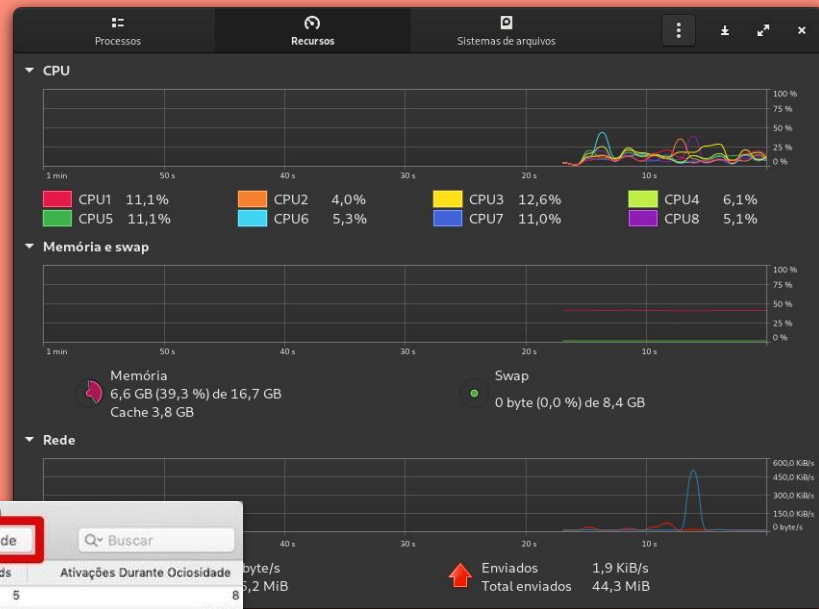
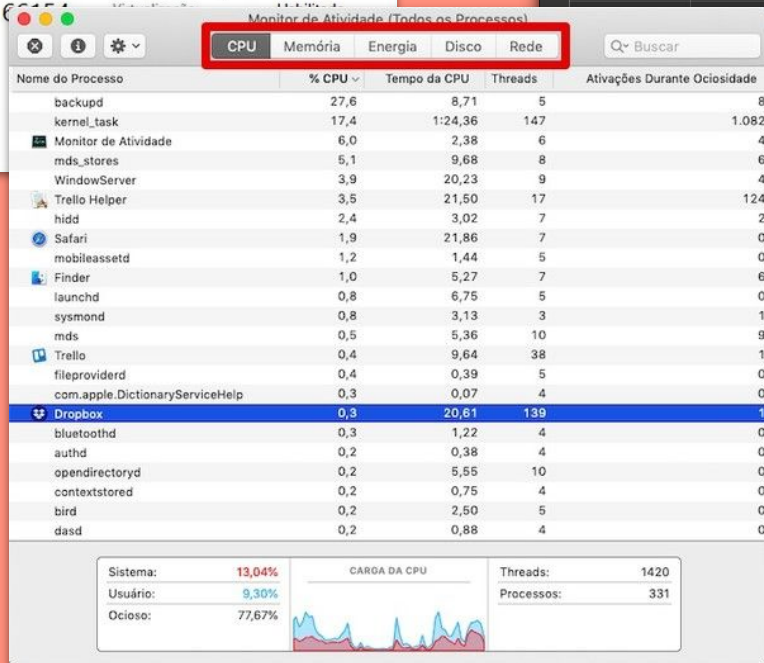
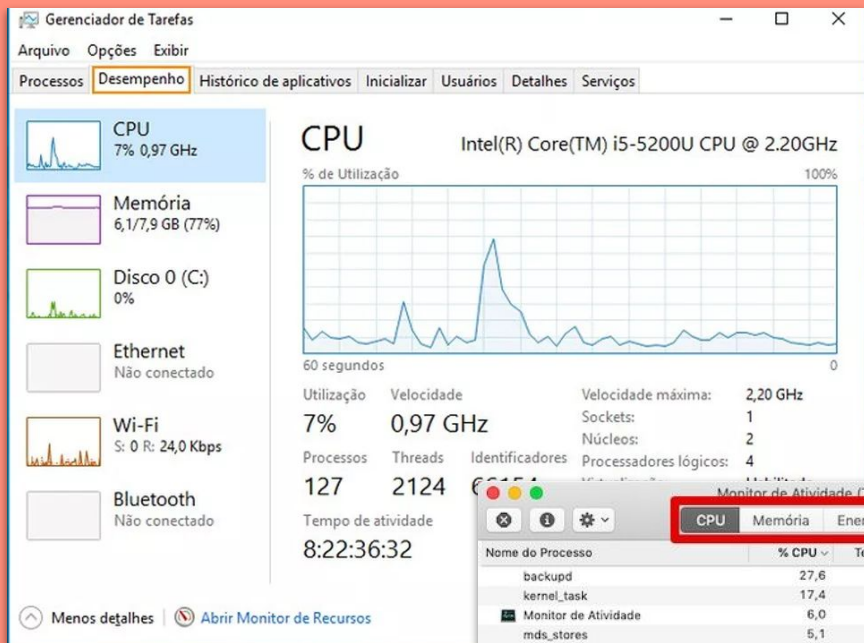
```
pip install psutil
```



Instalação



Onde uma biblioteca assim é útil?





ZABBIX



Nagios®

S
util

Utilitários do
sistema

Memória



O PSUtil consegue recuperar dados de memória virtual e memória swap.

```
1  from psutil import virtual_memory
2
3  virtual_memory()
4
5  # svmem(
6      total=16683110400,
7      available=10202509312,
8      percent=38.8,
9      used=5504155648,
10     free=6524657664,
11     ...
12 # )
```

Memória



O PSUtil consegue recuperar dados de memória virtual e memória swap.

```
1  from psutil import virt
2
3  virtual_memory()
4
5  # svmem(
6  total=16683110400,
7  available=10202509312,
8  percent=38.8,
9  used=5504155648,
10 free=6524657664,
11 ...
12 # )
```

Valor em bytes

$\text{total} / 1024 = \text{KB}$

$\text{total} / 1024 / 1024 = \text{MB}$

$\text{total} / 1024 / 1024 / 1024 = \text{GB}$

Swap



A swap também pode ser recuperada da mesma forma.

- sin: Número de bytes que já foram escritos na swap
- sout: Número de bytes que foram tirados da swap

São valores cumulativos

```
1  from psutil import swap_memory
2
3  swap_memory( )
4
5  # sswap(
6      total=16683110400,
7      percent=38.8,
8      used=5504155648,
9      sin=1050411008,
10     sout=1906720768,
11     # )
```

pip install py-dashing



Bora começar nosso mini dashboard?



CPU [0]



Sobre CPU podemos extrair diversos dados "físicos" interessantes:

- Frequência do processador
- Número de cores

```
1  from psutil import cpu_freq, cpu_count
2
3  cpu_freq()
4  # scpufreq(
5      current=1924.21,
6      min=400.0, # Mhz / 1_000 = Ghz
7      max=4000.0
8  # )
9
10 cpu_count() # Núcleos físicos + lógicos
11 # 8
12
13 cpu_count(logical=False) # Só físicos
14 # 4
```

CPU [1] – Dados do sistema em uso



- Porcentagem de uso total e por núcleo
- O tempo gasto pelo processador

```
1  from psutil import cpu_times, cpu_percent
2
3  cpu_percent( )
4  # 2.0
5
6  cpu_percent(interval=1)
7  # 3.1 - Calculo pelo tempo de intervalo, + preciso
8
9  cpu_percent(percpu=True) # Todos o núcleos (F + L)
10 # [18.5, 12.4, 13.2, 12.5, 13.1, 13.1, 11.4, 12.1]
```


CPU [1] – Dados do sistema em uso



- Porcentagem de uso total e por núcleo
- O tempo gasto pelo processador

```
1  from psutil import cpu_times
2
3  cpu_times()
4  scputimes(
5      user=11763.37, # Tempo gasto pelo user
6      system=3265.64, # Tempo gasto com o sistema
7      idle=428961.05, # Tempo gasto sem fazer nada
8      ...
9  )
```

CPU [1] – Dados do sistema em uso



- Porcentagem de uso total e por núcleo
- O tempo gasto pelo processador

```
1  from psutil import c
2
3  cpu_times()
4  scputimes(
5      user=11763.37, # Tempo gasto pelo user
6      system=3265.64, # Tempo gasto com o sistema
7      idle=428961.05, # Tempo gasto sem fazer nada
8      ...
9  )
```

Tempo em segundos

```
mm, ss = divmod(secs, 60)
hh, mm = divmod(mm, 60)
```

Bora adicionar mais
coisas no dashboard

_



Pq não?



Sensores [0]



Com os sensores podemos obter diferentes dados do nosso hardware.

- Velocidade dos coolers (fans)
- Porcentagem de bateria (em notebooks)

```
1  from psutil import sensors_battery, sensors_fans
2
3  sensors_fans() # Velocidade em RPM
4  # {'asus': [sfan(label='cpu_fan', current=3200)]}
5
6  sensors_battery()
7  # sbattery(
8      percent=100.0,
9      secsleft=<BatteryTime.POWER_TIME_UNLIMITED: -2>,
10     power_plugged=True
11  # )
```

Temperaturas - sensors[1]

```
1  from psutil import sensors_temperatures
2
3  sensors_temperatures()
4  {
5      'acpitz': [shwtemp(label='', current=41.0, high=97.0, critical=97.0)],
6      'pch_skylake': [shwtemp(label='', current=38.5, high=None, critical=None)],
7      'iwlwifi_1': [shwtemp(label='', current=27.0, high=None, critical=None)],
8      'coretemp': [
9          shwtemp(label='Package id 0', current=42.0, high=100.0, critical=100.0),
10         shwtemp(label='Core 0', current=40.0, high=100.0, critical=100.0),
11         shwtemp(label='Core 1', current=41.0, high=100.0, critical=100.0),
12         shwtemp(label='Core 2', current=41.0, high=100.0, critical=100.0),
13         shwtemp(label='Core 3', current=39.0, high=100.0, critical=100.0)
14     ]
15 }
```

Temperaturas - sensors[1]

Zona termal, pode ser qualquer sensor

```
1  from psutil import sensors_temperatures
2
3  sensors_temperatures
4  {
5      'acpitz': [shwtemp(label='', current=41.0, high=97.0, critical=97.0)],
6      'pch_skylake': [shwtemp(label='', current=38.5, high=None, critical=None)],
7      'iwlwifi_1': [shwtemp(label='', current=45.0, high=None, critical=None)],
8      'coretemp': [
9          shwtemp(label='Core 0', current=45.0, high=100.0, critical=100.0),
10         shwtemp(label='Core 1', current=45.0, high=100.0, critical=100.0),
11         shwtemp(label='Core 2', current=45.0, high=100.0, critical=100.0),
12         shwtemp(label='Core 3', current=39.0, high=100.0, critical=100.0)
13     ]
14 }
15 }
```

Platform Controller Hub

Bora adicionar mais
coisas no dashboard

_



Pq não?





Para recuperar dados de HD, também temos uma grande gama de opções. Podemos obter:

- Partições do disco
- Uso do disco
- Contadores de I/O

```
1 from psutil import disk_usage
2
3 disk_usage('/')
4 sdiskusage(
5     total=21378641920,
6     used=4809781248,
7     free=15482871808,
8     percent=22.5
9 )
```

```
1 from psutil import disk_partitions
2
3 disk_partitions()
4 [
5     sdiskpart(
6         device='/dev/sda3', mountpoint='/',
7         fstype='ext4',
8         opts='rw,errors=remount-ro',
9         maxfile=255, maxpath=4096
10    ),
11    ...
12 ]
```


I/O e disco



```
1  from psutil import disk_io_counters
2
3  disk_io_counters()
4  sdiskio(
5      read_count=147118,          # Leituras
6      write_count=409151,         # Escritas
7      read_bytes=3979602944,      # Bytes Lidos
8      write_bytes=9933423616,     # Bytes Escritos
9      read_time=71422,            # Milisegundos Lendo
10     write_time=859492,           # Milisegundos Escrevendo
11     busy_time=905390             # Tempo gasto em I/O
12 )
```

Bora adicionar mais
coisas no dashboard

_



Pq não?





Na API de rede temos diversas funções mega interessante. Talvez valha até uma live somente sobre essa parte do PSUtil.

- Contadores de I/O
- Conexões abertas (assunto pra outra live)
- ADDRS (interfaces -> IPs)
- Interfaces e seus status

Interfaces [0]



```
1  from psutil import net_if_stats
2
3
4  net_if_stats()
5  {
6      'lo':snicstats(
7          isup=True, # Literalmente Is Up
8          # Tipo de conexão: Full/ Half/ Unknow
9          duplex=<NicDuplex.NIC_DUPLEX_UNKNOWN: 0>,
10         speed=0, # Velocidade
11         mtu=65536 # Quantia em bytes da quantidade máxima de transmissão
12     ),
13 }
```

Interfaces [1]



```
1  from psutil import net_if_addrs
2
3
4  net_if_addrs()
5  {
6      'wlp2s0': [
7          snicaddr(
8              family=<AddressFamily.AF_INET: 2>, # Família do protocolo v4 ou v6
9              address='192.168.15.56', # Endereço dessa interface
10             netmask='255.255.255.0', # Mascara de rede
11             broadcast='192.168.15.255',
12             ptp=None # Ponto a ponto (VPNs)
13         ),
14         ...
15     ]
16 }
```

Contadores de rede



```
1  from psutil import net_io_counters
2
3
4  net_io_counters()
5  # snetio(
6      bytes_sent=165549046,    # bytes enviados
7      bytes_recv=1275641590,   # bytes recebidos
8      packets_sent=443312,     # pacotes enviados
9      packets_recv=1259816,    # pacotes recebidos
10     errin=0, # erro para receber
11     errout=0, # erro para enviar
12     dropin=1015, # pacotes descartados
13     ...
14  # )
```

Bora adicionar mais
coisas no dashboard

_



Pq não?



Mais utilitários



O psutil também conta com informações sobre o tempo de boot e sobre os usuários logados

```
1 from psutil import boot_time, users
2 from datetime import datetime
3
4 datetime.fromtimestamp(
5     boot_time()
6 ).strftime("%Y-%m-%d %H:%M:%S")
7 # '2022-06-13 01:07:28'
8
9 users()
10 # [
11     suser(
12         name='dunossauro',
13         terminal=':0',
14         host='localhost',
15         started=1655093248.0,
16         pid=1073)
17     # ]
```


Bora adicionar mais
coisas no dashboard

_



Pq não?



P

Util

Agora sim, os
processos

Processos



Talvez o item mais complexo de toda a lista de operações aceitas pelo PSUtil seja sua relação com processos.

Temos algumas funções muito úteis:

- Listagem de PIDs (Process IDs)
- Listagem de todos os processos por atributos
- Checagem de PIDs em memória
- Espera para finalização de processos

São MUITOS atributos



Vamos pra documentação





picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3

