

```

import psutil
import datetime
import pandas as pd
import numpy as np

# Definir listas vazias
pid = []
nome = []
uso_cpu = []
uso_memoria = []
percentual_uso_memoria = []
data_criacao = []
status = []
threads = []
for process in psutil.process_iter():
    pid.append(process.pid)
    nome.append(process.name())

uso_cpu.append(process.cpu_percent(interval=1)/psutil.cpu_count())

uso_memoria.append(round(process.memory_info().rss/(1024*1024),2))
percentual_uso_memoria.append(round(process.memory_percent(),2))

data_criacao.append(datetime.datetime.fromtimestamp(process.create_time()).strftime('%Y-%m-%d %H:%M:%S'))
status.append(process.status())
threads.append(process.num_threads())

# Criar dicionário
data = {"PIId":pid,
        "Nome": nome,
        "CPU":uso_cpu,
        "Memoria_MB":uso_memoria,
        "Memoria_percentual": percentual_uso_memoria,
        "Data_criacao": data_criacao,
        "Estado": status,
        "Threads": threads
        }

# converter para Pandas DataFrame
processos = pd.DataFrame(data)
# pid é o índice
processos = processos.set_index("PIId")
# ordenar os processos pelo consumo de memória
processos =processos.sort_values(by='Memoria_MB', ascending=False)
# mostrar o DataFrame
print(processos)
# Exportar o DataFrame para um arquivo csv
processos.to_csv('Processos.csv')

```

```

# Cada coluna do Pandas é do tipo de dados serie
processos.Memoria_MB
# Comando describe fornece o resumo dos dados de uma coluna
processos.Memoria_MB.describe()

# informações sobre partições do disco
print(psutil.disk_partitions())

# informações sobre espaço em disco
print(psutil.disk_usage('/'))

# Tempo desde o último boot em segundos
boot_time = psutil.boot_time()
print("psutil.boot_time() = {0}".format(boot_time))
# Tempo de boot em formato legível
print(datetime.datetime.fromtimestamp(psutil.boot_time()).strftime("%Y
-%m-%d %H:%M"))
# Usuários conectados no sistema
users = psutil.users()
print("psutil.users() = {0}".format(users))


# Os dados de um data frame podem ser exportados para numpy
memoria = processos.Memoria_MB.to_numpy()
# Colocar o seu código aqui
media = np.mean(memoria)
dvpd = np.std(memoria)
print(memoria)
print('-----')
print("Média: ",media)
print("Desvio Padrão: ",dvpd)

```