

```
def scanner_no_ip(endereco_ip, exibir_portas_fechadas=0):
```

O “scanner_no_ip” é uma função que irá realizar aqui a varredura de portas de um endereço IP

```
portas = [20, 21, 22, 23, 42, 43, 43, 69, 80, 109, 110, 115, 118, 143,  
          156, 220, 389, 443, 465, 513, 514, 530, 547, 587, 636, 873,  
          989, 990, 992, 993, 995, 1433, 1521, 2049, 2081, 2083, 2086,  
          3306, 3389, 5432, 5500, 5800]
```

Portas que receberão uma tentativa de conexão e verificar quais estão abertas ou não

Com o comando TRY se da inicio do processo de Scanner,

podendo encerra-lo pressionando <ctrl>+<c> que está relacionado ao comando EXCEPT

Execução principal do script

```
try:
    for porta in portas:
        sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        sock.settimeout(3)
        result = sock.connect_ex((endereco_ip, porta))
        if result == 0:
            print(f' {porta} >>>>>>> Aberta')
        else:
            if exibir_portas_fechadas > 0:
                print(f' {porta}')
            sock.close()
except KeyboardInterrupt:
    print('Você pressionou <Ctrl>+<C>')
    sys.exit()
```

```
ip_do_servidor = socket.gethostbyname('casasbahia.com.br')
```

Aqui será onde o scanner irá buscar o IP por meio das urls selecionadas,

função gethostbyname retorna a partir de um nome passado o endereço IP associado ao nome. A função realiza o papel de um DNS.

```
t1 = datetime.now()  
  
scanner_no_ip(ip_do_servidor)
```

Onde acontecerá a hora inicial do scanneamento utilizando o módulo datetime

```
t2 = datetime.now()  
total = t2 - t1  
print(f'Scanner finalizado em: {total}')
```

O código que representa hora final do scaneamento e saber quanto tempo durou o scanner