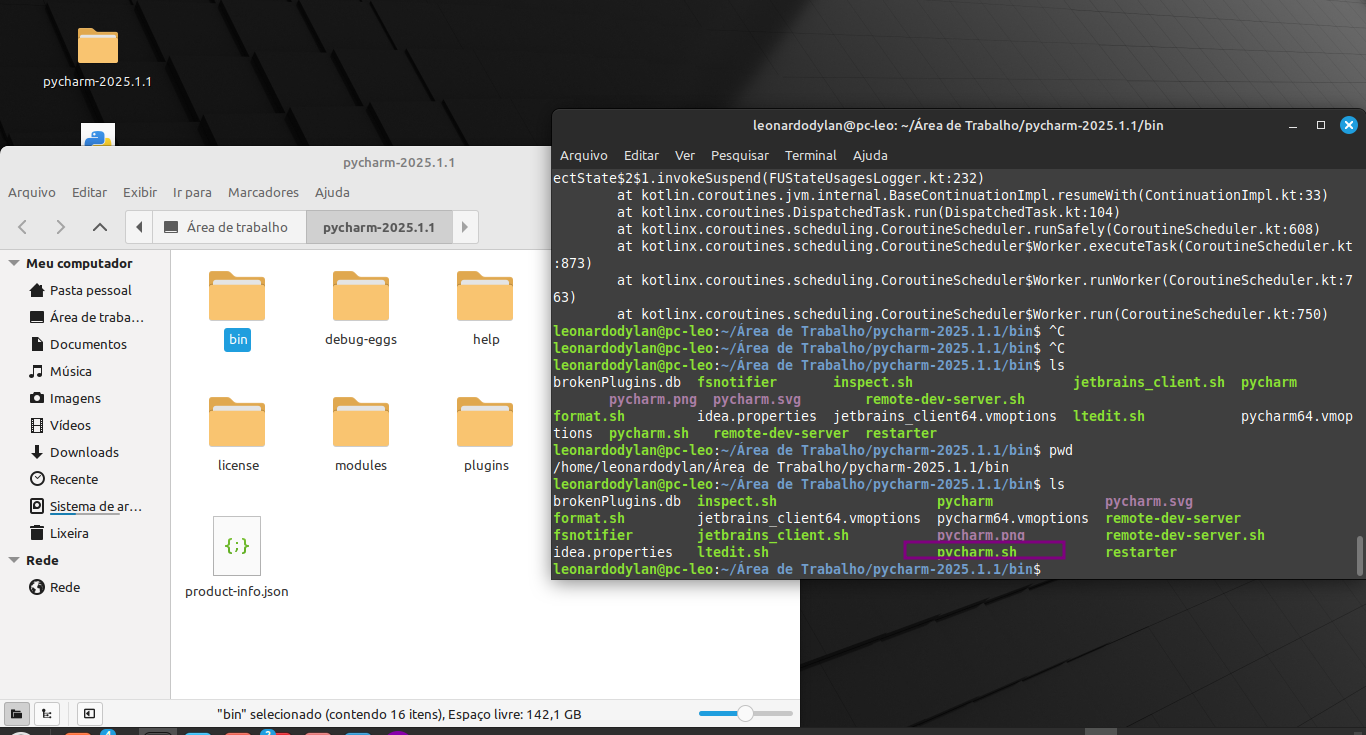
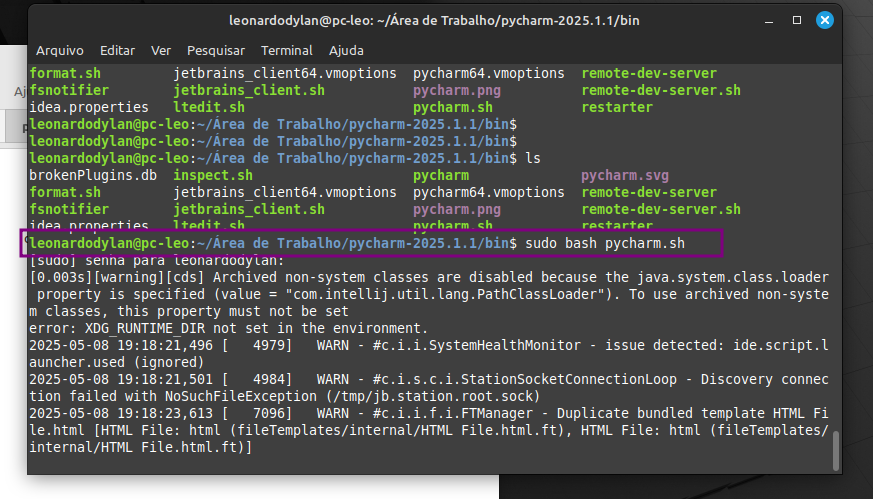
Documentação - Desafio Meli - NATIS

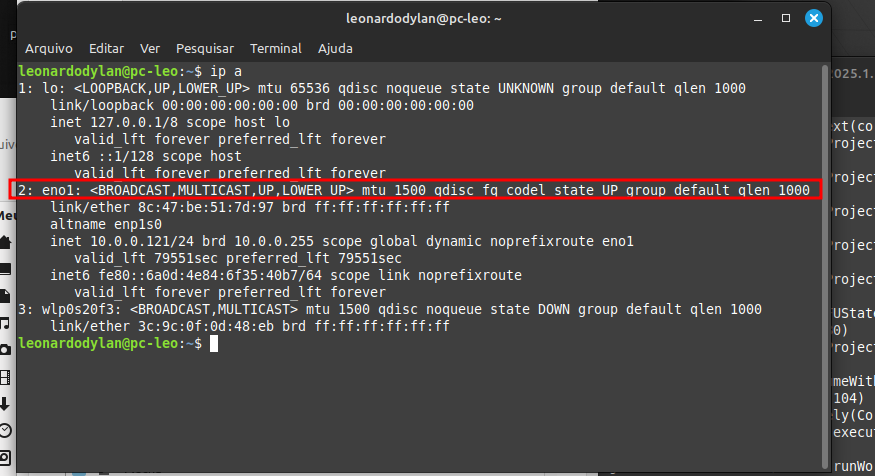
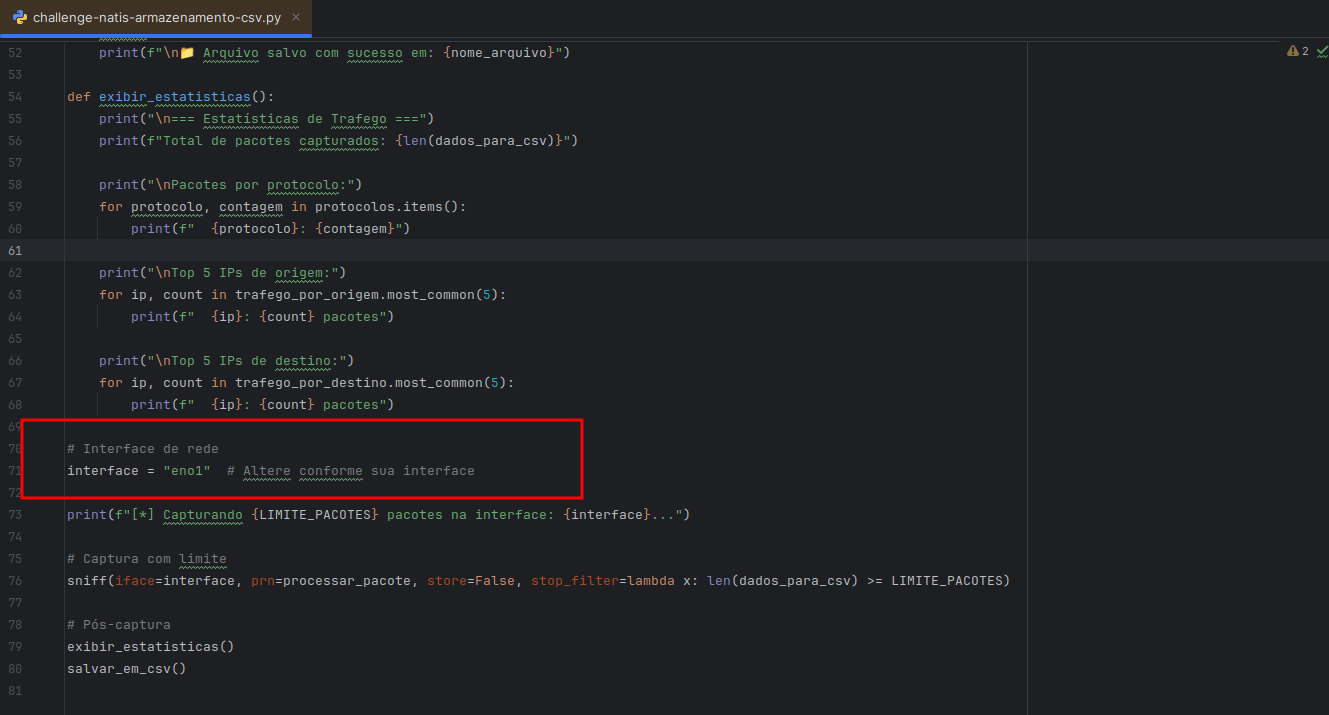
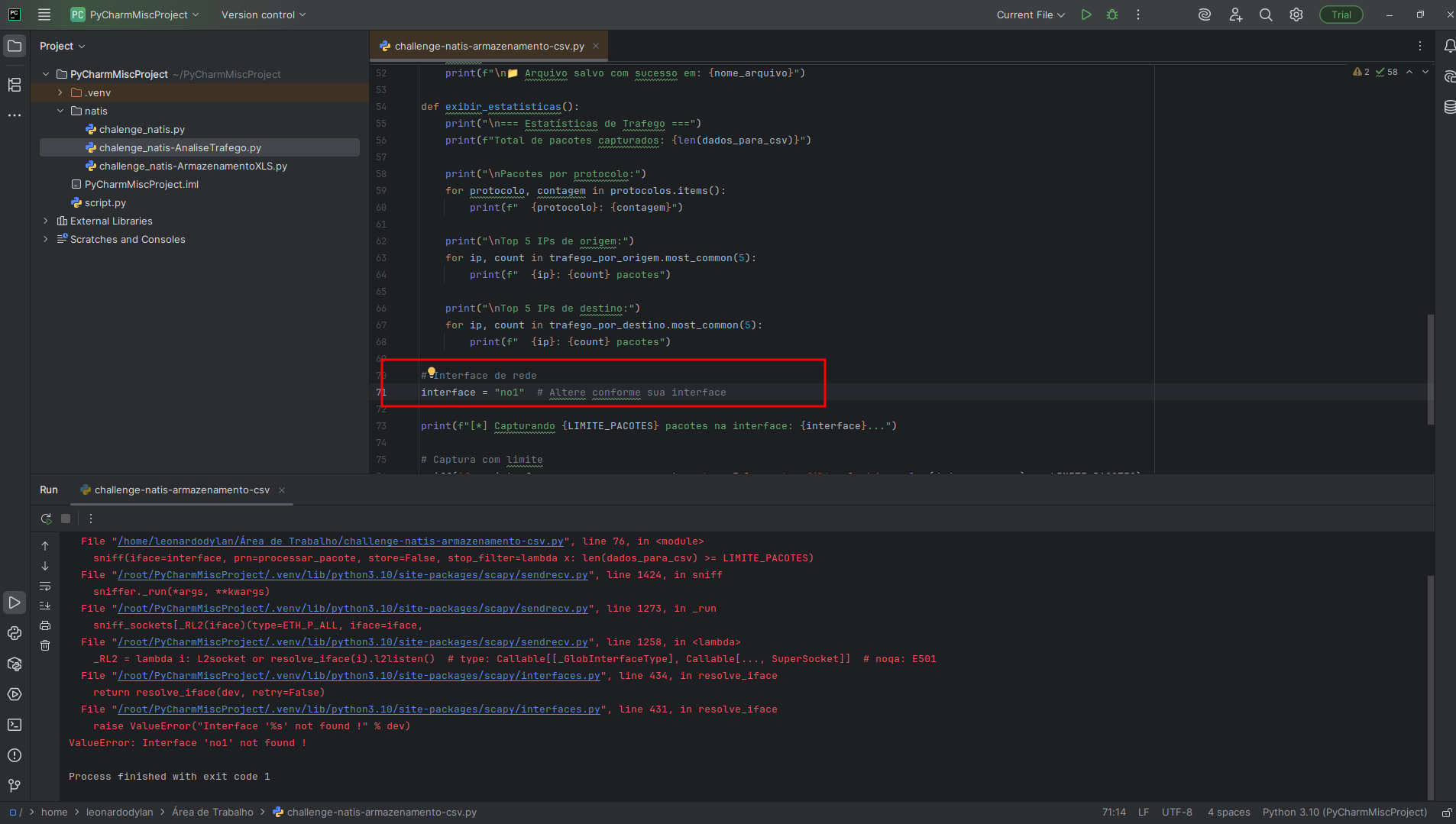
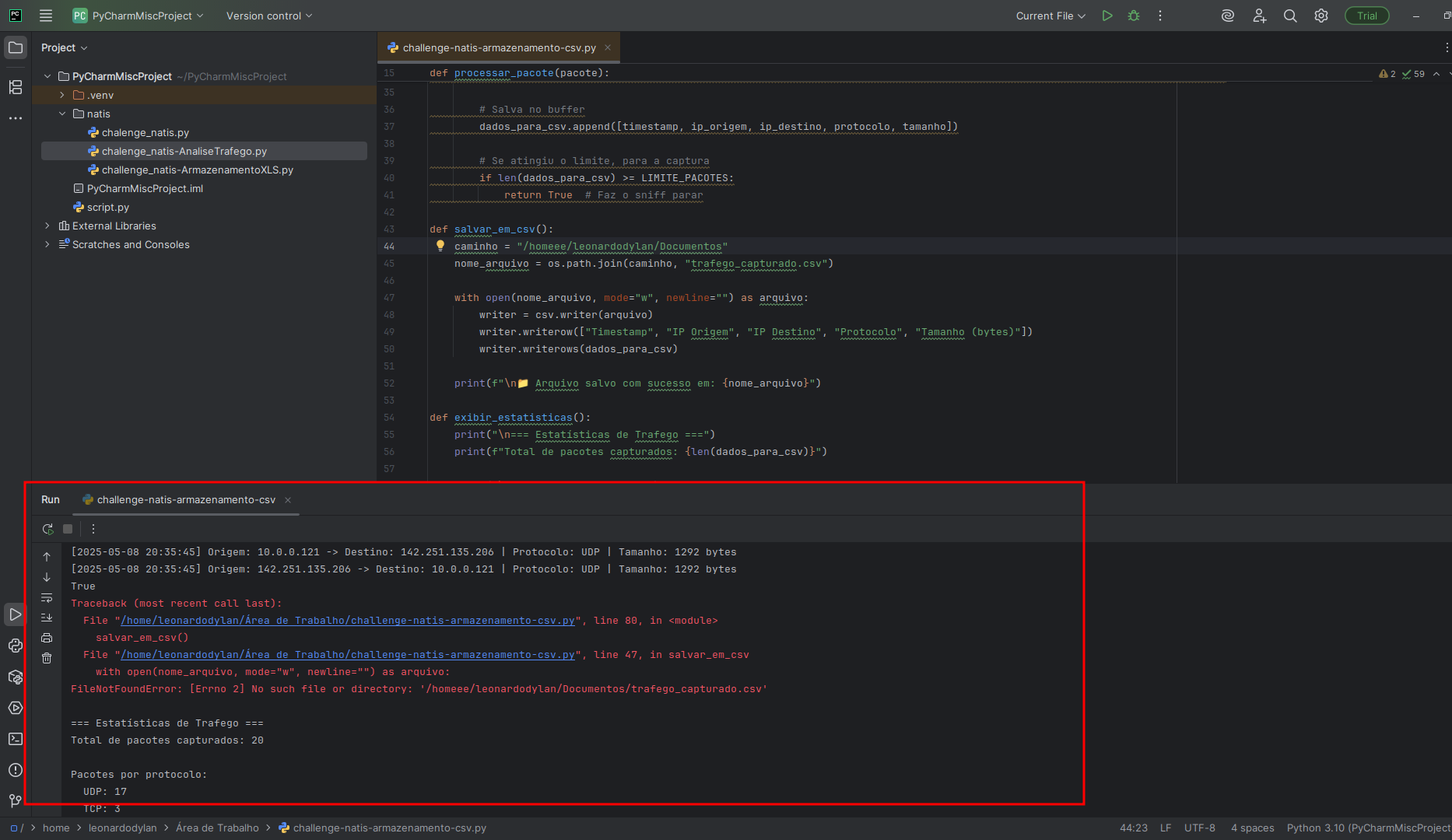
| Leonardo Dylan da Silva | 09/05/2025 |
| --- | --- |

Olá, abaixo compartilho detalhamento do ambiente de configuração para realização do challenge proposto.   
  
Github com scripts + Documentação: https://github.com/leoz1k/challenge-meli  
  
Utilizei um notebook próprio que já possuía uma distro Linux embarcada (Distro Linux Mint) sendo assim não elaborei os scripts em ambientes virtuais (Virtualbox)  
  
Versão OS - Utilizada: Linux Mint 22.1 (Distro baseada em Debian)  
  
Ferramentas instaladas para desenvolvimento do Challenge:  
Python3  
Pycharm Community - IDE de minha preferência.  
  
Linux MINT: <https://linuxmint.com/download.php>  
  
Python3 - Instalação através do terminal - sudo apt install python3 python3-pip -y  
  
Pycharm: <https://www.jetbrains.com/help/pycharm/installation-guide.html#toolbox_linux> (descompactar pacote e executar [pycharm.sh](http://pycharm.sh) dentro da pasta bin após extração do pycharm)  
  
  
  
  
  
Após a abertura da IDE necessitamos instalar os pacotes para desenvolvimento do challenge:  
Scapy - Recomendado na doc do challenge, realiza a captura de pacotes da placa de rede ou placas de redes do host executor do script.   
OS - Permite passar como parametro os diretórios do SO.  
CSV - Armazenar dados/variaveis e afins em arquivos/arquivos formato CSV.  
Datetime - Utilizei para coletar data/hora/dia dos pacotes coletados para melhor detalhamento dos pacotes capturados atráves do Scapy  
sniff: Função utilizada para capturar pacotes de rede de uma interface especificada.

IP: Permite acessar as informações dos pacotes IP (Internet Protocol). Ao capturar um pacote, você pode acessar detalhes como IP de origem e destino.

TCP: Refere-se ao protocolo TCP (Transmission Control Protocol), usado em pacotes de rede. O Scapy permite verificar se um pacote usa TCP.

UDP: Refere-se ao protocolo UDP (User Datagram Protocol), outro protocolo de comunicação de rede.

Para que o script seja executado com sucesso além da instalação das dependências necessitamos alterar a placa de rede no script para coleta das infos, para isso em ambientes UNIX (linux e mac) coletamos o nome da placa utilizando o comando “ip a”:  
  
No meu caso a conexão atual era a cabeada sendo referenciada por eno1 a qual está nos scripts, será necessário alterar este campo para que a coleta não se perca.   
  
  
  
  
Por fim, optei pela criação de 3 scripts separados de acordo com as exigências do challenge.   
  
1 - challenge\_natis\_captura\_pkt.py - Captura de Pacotes  
2- chalenge\_natis\_trafego\_analise.py - Analise de tráfego  
3 - Armazenamento - challenge-natis-armazenamento\_csv.py  
  
Possiveis erros ao executar o SCRIPT:  
Não alterar a interface para a interface ativa em sua máquina:  
  
  
Resolução: Através do comando “ip a” identificar placas de redes ativas e alterar o campo em questão do print acima:  
  
  
Erro ao gerar CSV com dados coletados: Path incorreto,  
  
Resolução: Apontar um diretório existente na máquina executora.