

Tarefa 4

```
#!/usr/bin/python

from re import compile
from ip2geotools.databases.noncommercial import DbIpCity
from geopy.distance import distance

def geolocIP(ip):
    dictGeoIp = dict()
    res = DbIpCity.get(ip, api_key="free")
    dictGeoIp.setdefault("city", res.city)
    dictGeoIp.setdefault("region", res.region)
    dictGeoIp.setdefault("pais", res.country)
    dictGeoIp.setdefault("latitud", res.latitude)
    dictGeoIp.setdefault("longitud", res.longitude)
    return(dictGeoIp)

with open('auth.log','r') as arquivo:
    axLineas = arquivo.readlines() #Queda o arquivo separado por comas
    arquivo.seek(0) #Voltamos ao inicio
    axCompleto = arquivo.read()

listaDiccionarios = list() #Lista Final
listaIps = list() #Lista Vacía

for linea in axLineas:
    if compile("sshd.* Failed .* (?:[0-9]{1,3}\.){3}[0-9]{1,3}").search(linea): #Se coincide a
#Expresión regular dentro da liña
        ip = compile("(?:[0-9]{1,3}\.){3}[0-9]{1,3}").findall(linea)[0] # Como o findall devolve unha
#lista, escollemos o elemento 0.
        listaIps.append(ip) #Engadir ip atopada na lista de IPs
    else:
        pass

listaIps = set(listaIps)

for ip in listaIps:

    listaUsuarios = compile(f"sshd.* Failed .* ({ip})").findall(axCompleto)
    listaNomesUsuario = list()
    for palabra in listaUsuarios:
        try:
            usuario = compile(f"\w+ from {ip}").findall(palabra)[0].split()[0] #Buscamos a palabra que
#vai antes de from.
        except:
            usuario = "NULL"
        listaNomesUsuario.append(usuario)
    listaNomesUsuario = set(listaNomesUsuario)

    print(listaNomesUsuario)

    dicInfo = {"ip":ip,"usuarios":listaNomesUsuario}
    dicInfo.update(geolocIP(ip)) #Para xuntar dous diccionarios temos que facer .update

    listaDiccionarios.append(dicInfo)
print(listaDiccionarios)
```