

# PIA Programacion

version

**Diego Osvaldo Avila, Mauricio Arriaga Islas, Jorge Pèrez García**

diciembre 08, 2020



# Contenido

<b>Documentacion PIA de programacion</b>	<b>1</b>
Instrucciones de uso	1
Webscraping (w)	1
Envío de Correos(c)	1
Encabezados(e)	1
Escaneo de Puertos(p)	1
Cifrado de un mensaje(m)	1
Tabla de Scripts:	1
Archivo Main	1
Codigo	1
Archivo Cifrado de Mensaje	2
Codigo	2
Archivo Envio de Correos	3
Codigo	3
Archivo Lectura de Encabezados	3
Codigo	3
Archivo Escaneo de Puertos	4
Codigo	4
Archivo WebScraping	5
Codigo	5
Archivo Powershell	5
Codigo	5



# Documentacion PIA de programacion

## Instrucciones de uso

Para todos los scripts se debe de utilizar el argumento -accion con la respectiva letra correspondiente del script. Los argumentos de nuestro script son:

### *Webscraping (w)*

-url: Es para agregar la dirección de la página para usar en el webscraping.

### *Envío de Correos(c)*

-destinatario: agregamos hacia quien está dirigido el correo. -tema: Agregamos el título del correo. -cuerpo: Escribe el contenido del correo. -archivo: Si quieres añadir un archivo con este argumento junto con el nombre y la extensión del archivo se puede adjuntar.

### *Encabezados(e)*

-búsqueda: tienes que ingresar que quieres buscar

### *Escaneo de Puertos(p)*

-ip: Agrega tu ip remota -ports: Indica que puertos quieres agregar

### *Cifrado de un mensaje(m)*

-mensaje: Agrega el mensaje que quieres cifrar -clave: agrega una clave para descifrar el mensaje

### *Tabla de Scripts:*

Scripts	Descripcion
Main	Menu del Proyecto
Cifrado=Codigos/Cifrado.rst	Script para cifrar mensajes
CorreosPIA	Script para el envio de correos
Encabezados	Script para escaneo de encabezados
Escaneo	Script para escanear puertos
Scraping	Script para hacer WebScraping

## Archivo Main

### Codigo

```
import argparse from scraping import scrapimage, scraplinks from CorreosPIA import envcorreo from Escaneo
import Escaneo from Encabezados import EscaneoHeaders from Cifrado import CifradoC
```

```
parser = argparse.ArgumentParser(«Webscraping») parser.add_argument(«-url», dest=»url», help=»Pagina a
scrapear») parser.add_argument(«-destinatario», dest=»dest», help=»Hacia quien esta dirigido» «el correo»)
parser.add_argument(«-tema», dest=»subject», help=»Escriba el titulo») parser.add_argument(«-cuerpo»,
dest=»body», help=»Escriba el cuerpo del correo») parser.add_argument(«-archivo», dest=»arch», help=»Nombre
y extencion del archivo») parser.add_argument(«-busqueda», dest=»buscar», help=»Indica que quiere buscar.» »
Ingresar solo un argumento»,type=str) parser.add_argument(«-ip», dest=»direc», help=»Indica tu ip remota»)
parser.add_argument(«-ports», dest=»ports», help=»Indica el puerto(s) que quieras» » consultar separandolos con
```

una coma Ejemplo:20,21,22») parser.add\_argument(«-mensaje», dest=»msg», help=»Mensaje a utilizar») parser.add\_argument(«-clave», dest=»clv», help=»Clave para cifrar») parser.add\_argument(«-accion», dest=»act», help=»Accion a realizar: Escaneo de puertos» «(p), Envio de correos(c), Webscraping(w), Busqueda de encabezados(e), Cifrar msg(m)»)

```
params = parser.parse_args() url = params.url receiver_email = params.dest subject = params.subject body = params.body filename = params.arch busqueda = params.buscar ip = params.direc puerto = params.ports message = params.msg clave = params.clv accion = params.act
```

**if (accion == «p»):**

```
puertos = params.ports.split(",") for i in range (len(puertos)):
```

```
    puertos[i] = int(puertos[i])
```

```
    Escaneo(ip, puertos)
```

**elif (accion == «c»):**

```
    envcorreo(receiver_email,subject,body,filename)
```

**elif (accion == «w»):**

```
    scrapimage(url) scraplinks(url)
```

**elif (accion == «e»):**

```
    EscaneoHeaders(busqueda)
```

**elif (accion == «m»):**

```
    CifradoC(message, clave)
```

**else:**

```
    print(«Tienes que elgir un accion valida, Ejemplo: -accion p») exit()
```

## Archivo Cifrado de Mensaje

### Codigo

**def CifradoC(message, clave):**

```
    espacios = 1 while espacios > 0:
```

```
        espacios = clave.count(" ") if clave.isalpha() == False:
```

```
            espacios += 1
```

```
    key = len(clave)
```

```
    SYMBOLS = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890 !?."
```

```
    translated = ""
```

**for symbol in message:**

**if symbol in SYMBOLS:**

```
    symbolIndex = SYMBOLS.find(symbol) translatedIndex = symbolIndex + key
```

**if translatedIndex >= len(SYMBOLS):**

```
    translatedIndex = translatedIndex - len(SYMBOLS)
```

**elif translatedIndex < 0:**

```
    translatedIndex = translatedIndex + len(SYMBOLS)
```

```
    translated = translated + SYMBOLS[translatedIndex]
```

**else:**

```
    translated = translated + symbol
```

```
print(translated)
```

## Archivo Envio de Correos

### Codigo

```
import os,time,random import email, smtplib, ssl from email import encoders from email.mime.base import
MIMEBase from email.mime.multipart import MIMEMultipart from email.mime.text import MIMEText import getpass
import logging

LOG_FORMAT = «%(asctime)s - %(message)s» logging.basicConfig(filename = «./RegistroExcepciones.log», level
= logging.DEBUG,
    format = LOG_FORMAT, filemode = "w")

logger = logging.getLogger()

try:
    import sys
except ImportError:
    logging.error(«No estan instalados los requerimientos») print(«Instalando Sys») os.system("pip install os-sys")
    exit()

def envcorreo(receiver_email,subject,body,filename):
    sender_email = «mai.pruebas03@gmail.com» password = («1926728»)

    message = MIMEMultipart() message[«From»] = sender_email message[«To»] = receiver_email
    message[«Subject»] = subject message[«Bcc»] = receiver_email

    message.attach(MIMEText(body, «plain»))

    try:
        with open(filename, «rb») as attachment:
            part = MIMEBase(«application», «octet-stream») part.set_payload(attachment.read())

            encoders.encode_base64(part)

            part.add_header(
                «Content-Disposition», f»attachment; filename= {filename}»,
            )

            message.attach(part)
    except TypeError:
        pass

    text = message.as_string() context = ssl.create_default_context() try:
        with smtplib.SMTP_SSL(«smtp.gmail.com», 465, context=context) as server:
            server.login(sender_email, password) server.sendmail(sender_email, receiver_email, text) print(«Listo
            revisa tu email ;)»)
    except Exception as e:
        logging.error(«No se ha podido iniciar sesion a su correo») print(«No se ha podido conectar a su correo»)
        pass
```

## Archivo Lectura de Encabezados

### Codigo

```
import os,time,random import sys import logging

LOG_FORMAT = «%(asctime)s - %(message)s» logging.basicConfig(filename = «./RegistroExcepciones.log», level
= logging.DEBUG,
    format = LOG_FORMAT, filemode = "w")
```

```

logger = logging.getLogger()
try:
    from googlesearch import search
except ImportError:
    logging.error(«No estan instalados los requerimientos») print(«Instalando google...») os.system("pip install google") exit()
try:
    import requests
except ModuleNotFoundError:
    logging.error(«No estan instalados los requerimientos») print(«Instalando requests») os.system("pip install requests") exit()
def EscaneoHeaders(busqueda):
    x=1 query = busqueda for enlace in search(query,tld=»com»,num=10,stop=11,pause=5):
        r = requests.get(enlace) headers=r.headers f= open(«Encabezados.txt», «a»)
        f.write((str(x))+»:»+(str(headers))+»n») x=x+1 f.close()
    print(«Se ha generado una lista de encabezados en el archivo Encabezados.txt»)

```

## Archivo Escaneo de Puertos

### Codigo

```

import socket import sys import logging import subprocess
LOG_FORMAT = «%(asctime)s - %(message)s» logging.basicConfig(filename = «./RegistroExcepciones.log», level
= logging.DEBUG,
    format = LOG_FORMAT, filemode = "w")
logger = logging.getLogger()
def Escaneo(ip,puertos):
    try:
        for port in puertos:
            sock=socket.socket(socket.AF_INET,socket.SOCK_STREAM) result = sock.connect_ex((ip,port))
            if result == 0:
                f= open(«Puertos.txt», «a») f.write(«nPuerto {}:t Abierto».format(port)) f.close()
            else:
                f= open(«Puertos.txt», «a») f.write(«nPuerto {}:t Cerrado».format(port)) f.close()
        sock.close() print(«Se ha generado un reporte de los puertos en Puertos.txt»)
        subprocess.call(r'C:WindowsSystem32WindowsPowerShellv1.0powershell.exe Set-ExecutionPolicy
-ExecutionPolicy Bypass", shell=True) p
        subprocess.Popen([r»C:WindowsSystem32WindowsPowerShellv1.0powershell.exe»,
r»C:UsersuserDownloadsLogs_PIA.ps1»), stdout=sys.stdout) p.communicate()
    except (OverflowError , UnboundLocalError) as error:
        logging.error(«El argumento ingresado es erroneo») print(«El argumento ingresado es erroneo») pass
    except socket.error as error:
        logging.error(«Error en la conexion») print(«Error en la conexion») sys.exit() pass

```



## Archivo WebScraping

### Codigo

```
import logging import re import sys import os,time,random from urllib.request import Request, urlopen import requests import json

LOG_FORMAT = «%(asctime)s - %(message)s» logging.basicConfig(filename = «./RegistroExcepciones.log», level = logging.DEBUG, format = LOG_FORMAT, filemode = "w") logger = logging.getLogger()

try:
    from bs4 import BeautifulSoup
except ImportError:
    logging.debug(«No estan instalados los requerimientos») print(«Instalando BeautifulSoup...») os.system("pip install bs4") exit()

def scrapimage(url):
    try:
        html = urlopen(url) bs = BeautifulSoup(html, "html.parser") images = bs.find_all("img", {"src":re.compile(".jpg")}) k= open(«Images.txt», «a») k.write(«Imágenes de la pagina:») for image in images:
            k.write(image["src"]+"n")

        print(«Se ha generado una lista de imagenes en Images.txt»)
    except Exception as e:
        logging.debug(«No se ha podido conectar a la url») print(«La url no es valida o no se ha podido conectar a ella») pass

def scraplinks(url):
    try:
        req = Request(url) html_page = urlopen(req) soup = BeautifulSoup(html_page, «lxml») links = [] for link in soup.findAll("a"):
            links.append(link.get("href"))

        f= open(«Links.txt», «a») f.write(«Links de la pagina:n») for item in links:
            f.write(«%sn» % item)

        print(«Se ha generado una lista de links en Links.txt») indicators = links
        api_key="5075cc54d875f160201bd274d461ad8f6defca9081aeb2859d098e93e8cc9467" url =
        "https://www.virustotal.com/vtapi/v2/url/report" for site in indicators:
            params = {"apikey": api_key, "resource":site} response = requests.get(url, params=params) h = open
            («Apireport.txt», «a») h.write(str(response.json()))

        print(«Se han escaneado todas las url, se ha generado un reporte Apireport.txt»)
    except Exception as e:
        logging.debug(«No se ha podido conectar a la url») print(«La url no es valida o no se ha podido conectar a ella») pass
```

## Archivo Powershell

### Codigo

```
Start-Transcript -Path Computer.log
Write-Host «everything will end up in Computer.log»
Stop-Transcript
```