# Casos prácticos

## Hacking con Python

Telefónica EDUCACIÓN DIGITAL

# Casos prácticos

## 1 | Caso práctico 1

Practicar libremente con los tipos de datos básicos, los operadores, asignaciones, sentencias, bucles, etc.

## 1 | Solución

La solución es totalmente abierta. Algún ejemplo en la lección de teoría.

## 2 | Caso práctico 2

Averiguar si el pais informado al registrar el dominio “wikipedia.org” (donde está la sede) es realmente donde se encuentra localizado el servidor (Pista: utilizar la técnica de Whois y la librería pygeoip).

## 2 | Solución

Una posible solución:

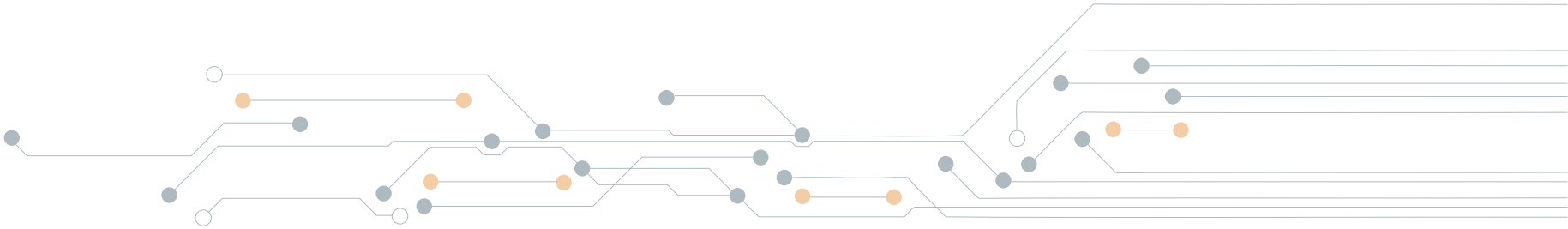
```
import pythonwhois
import pygeoip

whois_result = pythonwhois.get_whois('wikipedia.org')
whois_country = whois_result['contacts']['admin']['country']

geo_result = pygeoip.GeoIP('GeoIP.dat')
geo_country = gip.country_code_by_name('wikipedia.org')

if whois_country == geo_country:
 print("El servidor está hubicado en el mismo pais")

else:
 print("El servidor está en otro pais: Whois=",whois_country, " geo=", geo_country)
```



## 3 | Caso práctico 3

Buscar mediante Shodan 5 servidores que tengan el servicio ProFTPD en la versión vulnerable 1.3.3a y mostrar las IPs por consola.

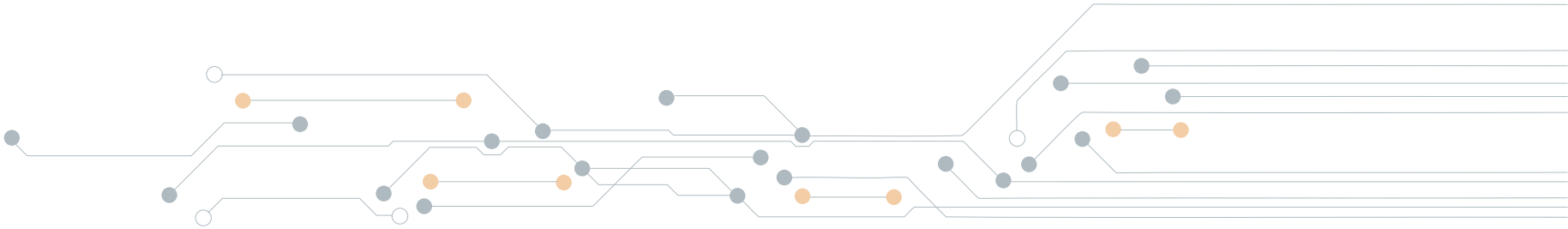
## 3 | Solución

Una posible solución:

```
import shodan
SHODAN_API_KEY = "XXXXX"

api = shodan.Shodan(SHODAN_API_KEY)
r = api.search('ProFTPD 1.3.3a')

for i in range(0,5):
        print(r['matches'][i]['ip_str'])
```



## 4 | Caso práctico 4

Utilizar la librería de Tweepy de Twitter para obtener los últimos Tweet geolocalizados de la cuenta "'stevewoz'".

## 4 | Solución

Una posible solución:

```
import tweepy

CONSUMER_KEY = "XXXXX"
CONSUMER_SECRET = "XXXXX"

OAUTH_TOKEN = "XXXXXX"
OAUTH_TOKEN_SECRET = "XXXX"

auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(OAUTH_TOKEN, OAUTH_TOKEN_SECRET)

api = tweepy.API(auth)

timeline = api.user_timeline('stevewoz')
for tweet in timeline:
        print("Aplicación de envío: ", tweet.source)
        print("Coordenadas: ",tweet.coordinates['coordinates'])
        print("Nombre del lugar: ", tweet.place.name)
        print("Tipo del lugar: ", tweet.place.place_type)
        print("Nombre completo: ", tweet.place.full_name)
        print("Pais: ", tweet.place.country)
        print(". . . . . .")
```

## 5 | Caso práctico 5

Utilizar el método de Streams de la librería Tweepy de Twitter para quedarse a la escucha de los Tweets localizados en la puerta del Sol de Madrid.

## 5 | Solución

Una posible solución:

```
import tweepy

CONSUMER_KEY = "XXX"
CONSUMER_SECRET = "XXXX"

OAUTH_TOKEN = "XXX"
OAUTH_TOKEN_SECRET = "XXXXX"

auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(OAUTH_TOKEN, OAUTH_TOKEN_SECRET)

api = tweepy.API(auth)

class CustomStreamListener(tweepy.StreamListener):
        def on_status(self, status):
                print("Nombre de la cuenta: ", status.user.screen_name)
                print("Fecha de creación: ", status.created_at)
                print("Texto del Tweet: ",status.text)
                print(". . . . . .")
        def on_error(self, status_code):
                print ('Se ha encontrado un error con el código:', status_code)
                return True
        def on_timeout(self):
                print('Timeout...')
                return True

stream = tweepy.streaming.Stream(auth, CustomStreamListener())

#puerta del sol  ----  [40.415274, -3.705557] [40.418309, -3.701496]

p1lon =- 3.705557
p1lat = 40.415274
p2lon = -3.701496
p2lat = 40.418309

stream.filter(locations=[p1lon, p1lat, p2lon, p2lat])
```

## 6 | Caso práctico 6

Indicar si el número de resultados (solo el total) obtenidos de la página haveibeenpwned y hesidohackeado son iguales para la cuenta de correo “freeman@hotmail.com”. (Pista: para haveibeenpwned sumar los DataLeaks y los Pastes).

## 6 | Solución

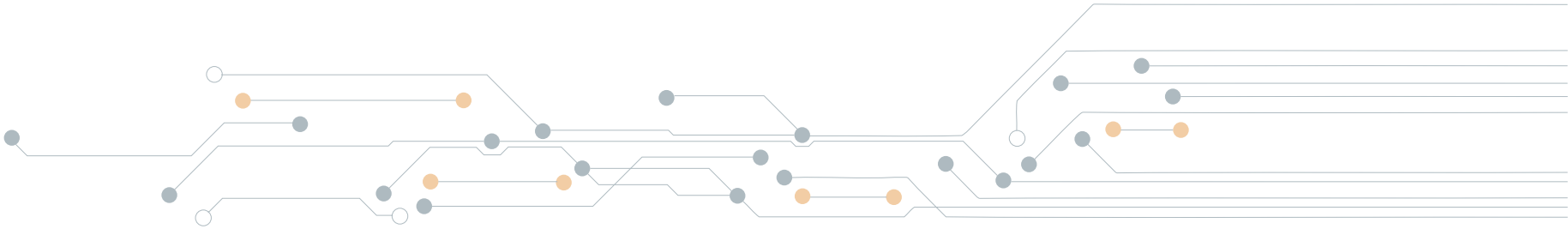
Una posible solución:

```
import requests
import json
import pypwned

h1_dataleaks = pypwned.getAllBreachesForAccount(email="freeman@hotmail.com")
h1_pastes = pypwned.getAllPastesForAccount(account="freeman@hotmail.com")
h1_total = len(h1_dataleaks) + len(h1_pastes)

h1_dataleaks = requests.get("https://hesidohackeado.com/api?q=freeman@hotmail.com")

if str(h1_total) == h1_dataleaks.json()['results']:
        print("El número de filtraciones son similares")
else:
        print("No coinciden el número de filtraciones: haveibeenpwned=",str(h1_total), " hesidohackeado=", h1_dataleaks.json()['results'])
```



## 7 | Caso práctico 7

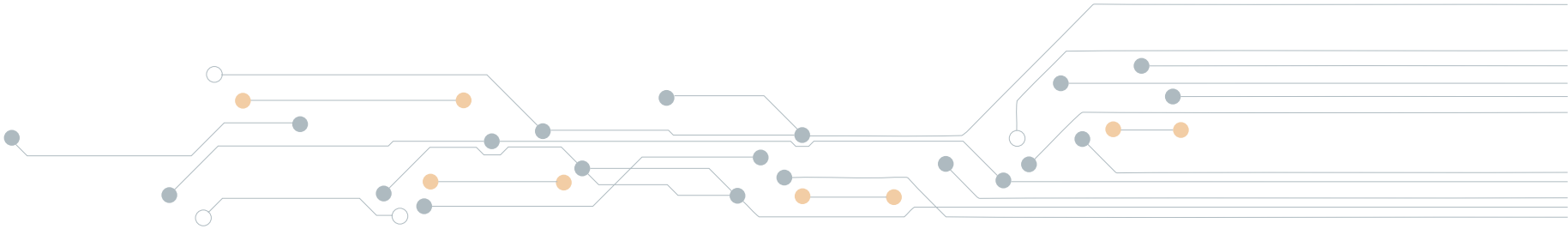
Utilizar la librería PyPDF2 para obtener todos los metadatos de un fichero PDF cualquiera.

## 7 | Solución

Una posible solución:

```
from PyPDF2 import PdfFileReader

pdf = PdfFileReader(open('file_path.pdf','rb'))
metadata = pdf.getDocumentInfo()
for item in metadata:
        print ("--" + item + "=" + metadata[item])
```



## 8 | Caso práctico 8

Elegir un tipo de escaneo de red de los mostrados en la lección e implementarlo.

## 8 | Solución

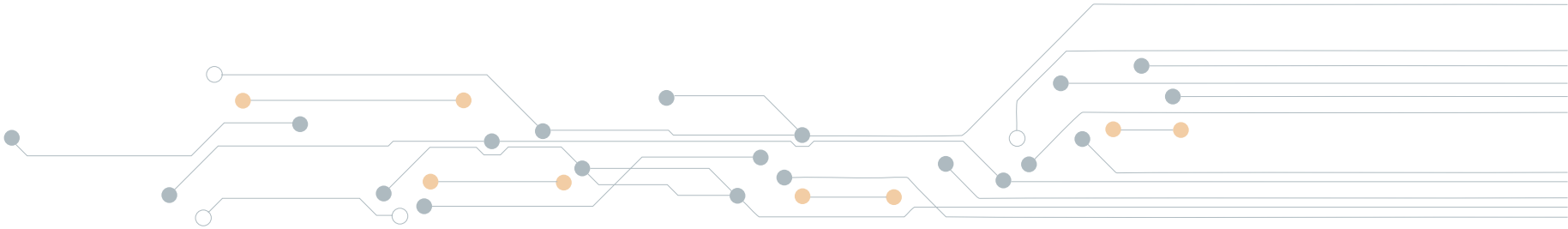
Respuesta abierta, una posible solución (TCP Stealth Scan):

```
from scapy.all import *

dst_ip = "xxx.xxx.xxx.xxx"
ports = "21:22:80:110:135:139:455:8080"

scanPorts = ports.replace(" ","").strip().split(":")

for port in scanPorts:
        r = sr1(IP(dst=dst_ip)/TCP(dport=int(port),flags="S"))
        if r is None:
                print("El puerto ", str(port), " está filtrado.")
        elif(r.haslayer(TCP)):
                if(r.getlayer(TCP).flags == 0x12):
                        r2 = sr1(IP(dst=dst_ip)/TCP(sport=src_port,dport=int(port),flags="R"))
                        print("El puerto ", str(port), " está abierto.")
                elif(r.getlayer(TCP).flags == 0x14):
                        print("El puerto ", str(port), " está cerrado.")
        elif(r.haslayer(ICMP)):
                if(int(r.getlayer(ICMP).type)==3 and int(r.getlayer(ICMP).code) in [1,2,3,9,10,13]):
                        print("El puerto ", str(port), " está filtrado.")
```



## 9 | Caso práctico 9

Utilizar la herramienta Nmap desde un script y mostrar por pantalla todos los puertos abiertos bajo el protocolo "TCP", los servicios, y sus versiones. Utilizar los siguientes parámetros para la herramienta: -sV -O -p -p 21,22,80,110,135,139,455,8080'.

## 9 | Solución

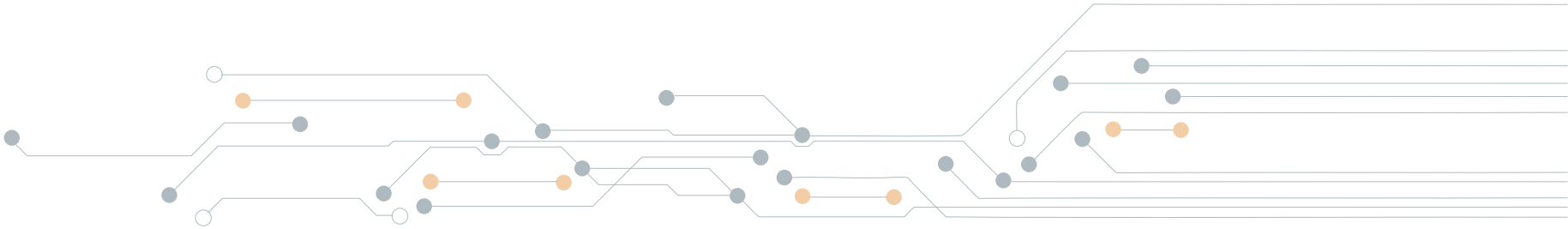
Una posible solución:

```
import nmap

host='10.0.2.2'
nm = nmap.PortScanner()

r = nm.scan(host, arguments='-sV -O -p 21,22,80,110,135,139,455,8080')

print(nm[host]['tcp'])
```



## 10 | Caso práctico 10

Mediante un script comprobar las funcionalidades de pareado y comprobar cerrojo de la Herramienta Latch.

Para la realización de esta práctica hay que registrarse en la Web de Latch (**https://latch.elevenpaths.com/www/**) y descargar la aplicación desde los Markets oficiales:

## 10 | Solución

Una posible solución:

**Pareado de un dispositivo:**

```
import latch

AAP_ID="XXXXX"
SECRET_TOKEN="XXXXX"

latch = latch.Latch(AAP_ID, SECRET_TOKEN)
PAIR_TOKEN="4sbvbx"
r = latch.pair(PAIR_TOKEN)

accountID = r.get_data()['accountId']
print("El accountID pareado es:", accountID)
```

**Comprobar cerrojo:**

```
import latch

AAP_ID="XXXXXX"
SECRET_TOKEN="XXXXXX"

latch = latch.Latch(AAP_ID, SECRET_TOKEN)

r2 = latch.status("accountID")

r2.get_data()
```

Telefónica EDUCACIÓN DIGITAL