

# ExamenFinalSimulacion

February 8, 2021

**0.1 Nombre: Xavier Jarro**

**0.2 Materia: Simulacion**

**Objetivo:** • Consolidar los conocimientos adquiridos en clase para desarrollar simulaciones automática de procesos RPA.

**Enunciado:** • En base a ello, diseñe y desarrolle un modelo y/o script que permita realizar marketing digital automatizado del proceso de campaña política para un partido:

Generar un documento de Excel Online o Google Drive Calc en donde se tenga un listado de las persona para enviar la notificación por email el mismo que constara de: Nombre, Apellido, Email, Estado (Activo/Desactivo).

Se debe generar un flyer (canvas.com) o imagen (PaintOnline), etc. que permitan identificar la fecha de publicación, hora, candidato, partido y cualquier dato de interés adicional.

Ademas, deben de publicar dentro de cualquier red social(facebook, instagram, twitter, etc.) el flyer o imagen para ello pueden utilizar sus cuentas personales o crear una cuenta para esta tarea.

Después, enviar por email el flyer o imagen generada (los correos se encuentran en el primer archivo).

Finalmente, generar un informe de las personas que visitaron la red social, comentaron o realizaron algún tipo de acción o evento y guardarlo en el GIT o en un documento en Google Drive.

**0.2.1 Importacion de librerias.**

```
[45]: import pandas as pd
import smtplib
import email.message
from datetime import date
from datetime import datetime
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.common.exceptions import TimeoutException
from time import sleep
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support.ui import Select
from selenium.webdriver.common.by import By
from selenium.webdriver.support import expected_conditions
```

```

from selenium.common.exceptions import NoSuchElementException
import time
import os
import pyautogui
import cv2

```

## 0.2.2 Metodo para enviar correos con archivo de drive

```

[46]: def correo():
    sheet_url = 'https://docs.google.com/spreadsheets/d/
    ↪1AfFbYSACM89S77jGx5sjcDyDOB-zCvT_i8op3hcuyo/edit#gid=0'
    csv_export_url = sheet_url.replace('/edit#gid=', '/export?format=csv&gid=')
    today = date.today()
    df=pd.read_csv(csv_export_url)
    dest=[i[0]+" "+i[1]+" <"+i[2]+">" for i in df.values]
    server = smtplib.SMTP('smtp.gmail.com:587')
    email_content = ""

    <html>

    <head>
        <title>Elecciones 2021</title>
    </head>

    <body>

        <table id="main" width="600" align="center" cellpadding="0"
    ↪cellspacing="15" bgcolor="ffffff">
            <tr>
                <td>
                    <table id="header" cellpadding="10" cellspacing="0"
    ↪align="center" bgcolor="8fb3e9">
                        <tr>
                            <td width="570" align="center" bgcolor="#0a14d8">
                                <h1>""+str(today)+"</h1>

                            </td>
                        </tr>
                        <tr>
                            <td width="570" align="center" bgcolor="#0a14d8">
                                <h1>Elecciones</h1>
                            </td>
                        </tr>
                    </table>
                </td>
            </tr>

```

```

        <tr>
            <td>
                <table id="content-3" cellpadding="0" cellspacing="0"
→align="center">
                    <tr>
                        <td valign="top" bgcolor="d0d0d0" style="padding:5px;
→">
                            
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
</td>
</tr>
</table><!-- wrapper -->
</body>
</html>

"""

msg = email.message.Message()
msg['Subject'] = 'Campaña Electoral - Lista 18'
msg['From'] = 'Star Bank <starbankoficial@gmail.com>'
password = "Covid1991X"
msg.add_header('Content-Type', 'text/html')
msg.set_payload(email_content)
s = smtplib.SMTP('smtp.gmail.com:587')
s.starttls()
s.login('starbankoficial@gmail.com', password)
s.sendmail(msg['From'], dest, msg.as_string())

```

### 0.2.3 Metodo para crear imagen

```

[47]: def crear_imagen():
    imagen = cv2.imread(r"C:\Users\xa_cx\Downloads\yaku.jpg")
    imagenFlayer = imagen.copy()
    today = date.today()
    now = datetime.now()
    hora = str(today)+' '+str(now.hour)+":"+str(now.minute)
    cv2.putText(imagenFlayer, hora, (500, 750), cv2.FONT_HERSHEY_COMPLEX, 1.
→0, (255, 255, 255), 2)
    cv2.imwrite(r"C:\Users\xa_cx\Downloads\yakuPost.jpg", imagenFlayer)

```

```
cv2.waitKey(0)
```

## 0.2.4 Metodo para postear en facebook

```
[48]: def post():
    usr = "xavierjarro@gmail.com"
    pwd = "Covid1991X"
    PATH = 'msedgedriver.exe'
    browser = webdriver.Edge(executable_path=PATH)
    browser.get("http://www.facebook.com")
    sleep(2)
    text_urr = browser.find_element_by_id("email")
    text_urr.send_keys(usr)
    text_pwd = browser.find_element_by_id("pass")
    text_pwd.send_keys(pwd)
    text_pwd.send_keys(Keys.RETURN)
    sleep(20)
    loguin = browser.find_element_by_xpath("/html/body/div[1]/div/div[1]/div/
    ↪div[3]/div/div/div[1]/div[1]/div/div[2]/div/div/div[3]/div/div[2]/div/div/div/
    ↪div[2]/div[2]/div[1]/span[2]/span").click()
    sleep(5)
    crear_imagen()
    pyautogui.write(r"C:\Users\xa_cx\Downloads\yakuPost.jpg")
    pyautogui.press("enter")
    sleep(5)
    loguin = browser.find_element_by_xpath("/html/body/div[1]/div/div[1]/div/
    ↪div[4]/div/div/div[1]/div/div[2]/div/div/div/form/div/div[1]/div/div/div[3]/
    ↪div[2]/div").click()

main()
```

## 0.2.5 Ejecucion de programa.

```
[38]: if __name__ == '__main__':
    correo()
    post()
```

**Referencias:** <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6291769/>

<https://www.mdirector.com/marketing-automation/estrategias-de-marketing-automation-para-fidelizar.html>

<https://blog.inconcertcc.com/automatiza-el-marketing-digital-en-tus-redes-sociales/>