

Programación con Python

POR VÍCTOR FUENTES



Análisis de texto

```
13 texto = ('POR ÚLTIMO, LA FORMA MÁS RECOMENDABLE Y PYTHÓNICA, PERO MÁS COMPLEJA, SERÍA USAR MATPLOTLIB MEDIANTE LA INTERFAZ O
14
15 t = texto.replace('.', ' ').replace(',', ' ')
16 print(t)
17 print()
18
19 t = texto.lower()
20 print(t)
21 print()
22
23 for vocal in ('AEIOU'):
24     print(vocal, ':', texto.count(vocal))
25 print()
26
27 max_veces = 0
28 max_letra = ""
29 for l in ('mnlrs') :
30     print(l, ":", t.count(l))
31     if t.count(l) > max_veces :
32         max_veces = t.count(l)
33         max_letra = l
34 print('consonante mas usada', max_letra, max_veces)
35 print()
36
37 palabras = t.split(" ")
38 print()
39
40
41 for palabra in palabras :
42     if len(palabra)>6 and palabra[0] not in ('aeiou') :
43         print(palabra)
44 print()
45
46 text = input("Entre texto: ")
```

POR ÚLTIMO LA FORMA MÁS RECOMENDABLE Y PYTHÓNICA, PERO MÁS COMPLEJA, SERÍA USAR MATPLOTLIB MEDIANTE LA INTERFAZ ORIENTADA A OBJETOS CUANDO SE PROGRAMA CON MATPLOTLIB, NO MIENTRAS SE TRABAJA INTERACTIVAMENTE. ESTA ES LA FORMA QUE PERMITE TENER MÁS CONTROL SOBRE EL CÓDIGO. QUIZÁ VEAMOS ESTO EN EL FUTURO SI OS ANIMÁIS A ESCRIBIR SOBRE ELLO.

por último, la forma más recomendable y pythónica, pero más compleja, sería usar matplotlib mediante la interfaz orientada a objetos. cuando se programa con matplotlib, no mientras se trabaja interactivamente, esta es la forma que permite tener más control sobre el código. quizá veamos esto en el futuro si os animáis a escribir sobre ello.

A : 30
E : 33
I : 18
O : 26
U : 6

m : 17
n : 15
l : 15
s : 19
r : 22
consonante mas usada r 22

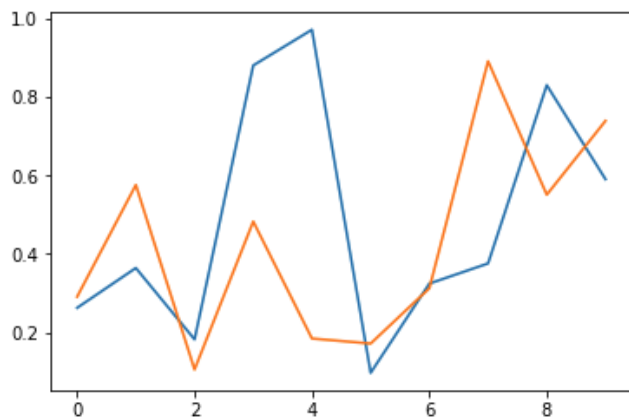
último,
recomendable
pythónica,
compleja,
matplotlib
mediante
programa
matplotlib,
mientras
trabaja
permite
control
código.

Entre texto: por ultimo

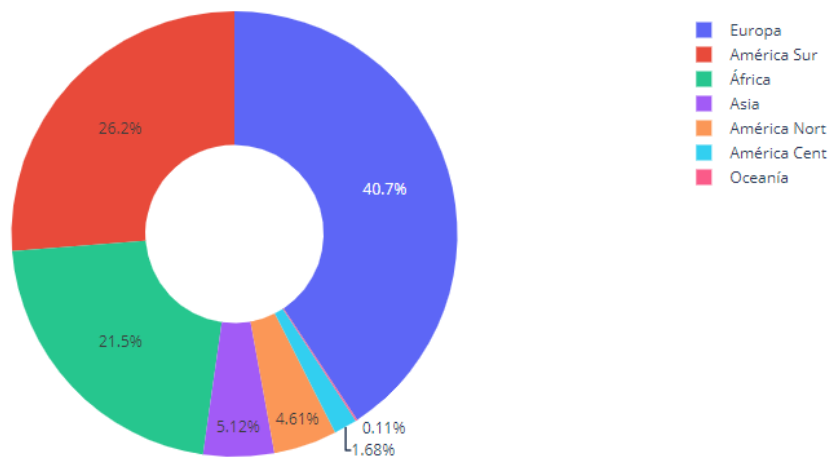
MATPLOTLIB, NUMPY, PLOTLY

LIBRERIA MATPLOTLIB amb NUMPY

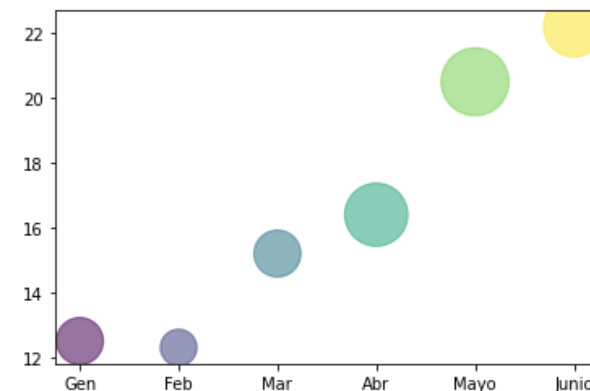
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 plt.plot(np.random.rand(10))
4 plt.plot(np.random.rand(10))
5 plt.show()
```



```
1 import plotly.graph_objects as go
2
3 labels = ['Europa', 'África', 'América Nort', 'América Cent', 'América Sur', 'Asia', 'Oceania']
4 values = [round((x + y)/2, 2) for x, y in zip(hombres, mujeres)]
5
6 fig = go.Figure(data=[go.Pie(labels=labels, values=values, hole=.4)])
7 fig.show()
8
```



```
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 meses = ['Gen', 'Feb', 'Mar', 'Abr', 'Mayo', 'Junio']
6 temperaturas = [12.5, 12.3, 15.2, 16.4, 20.5, 22.2]
7 plujiame = [335, 203, 334, 604, 700, 555]
8
9 s = [x*2.5 for x in plujiame]
10
11 colors = [1,2,3,4,5,6]
12 plt.scatter(meses, temperaturas, c=colors, s=s, alpha = 0.5)
13 plt.xlabel = 'meses'
14 plt.ylabel = 'temperaturas'
15 plt.show()
16
17
```



Clases y herencia

```
6 class Carrito:
7
8     def __init__(self, lista = []):
9         self.lista = lista
10
11     def agregar_al_carro(self, producto):
12         self.lista.append(producto)
13
14
15
16     def eliminar_del_carro(self, referencia):
17         for p in self.lista:
18             if p.referencia == referencia:
19                 self.lista.remove(p)
20                 break
21
22     def mostrar_carro(self):
23         print('-----carrito-----')
24         for p in self.lista:
25             print(p)
26
27
28     def total_compra(self):
29         total = 0
30         for p in self.lista:
31             total += p.pvp
32         print('Total carrito', total)
33
34
35
36 carrito = Carrito([])
37 carrito.mostrar_carro()
38
39 carrito.agregar_al_carro(t)
40 carrito.mostrar_carro()
41 carrito.agregar_al_carro(b)
42 carrito.mostrar_carro()
43
44 carrito.eliminar_del_carro(t)
45 # carrito.mostrar_carro()
46
47 carrito.total_compra()
```

```
-----carrito-----
-----carrito-----
REFERENCIA      45WD
NOMBRE          Bajera
PVP             12.5
DESCRIPCIÓN     1unit
color           blanco
material                algodón
```

```
-----carrito-----
REFERENCIA      45WD
NOMBRE          Bajera
PVP             12.5
DESCRIPCIÓN     1unit
color           blanco
material                algodón
```

```
REFERENCIA      56J
NOMBRE          Taladro
PVP             35.5
DESCRIPCIÓN     percutor
medida          30
peso            1.5
```

Total carrito 48.0

Librería Pandas

```
5 import pandas as pd
6
7 dades = {'Nom': ['Sònia', 'Laura', 'David', 'Rosa', 'Sam'],
8         'Dept': ['PROD', 'ADMIN', 'MANT', 'ADMIN', 'PROD'],
9         'DiesV': [32, 55, 20, 43, 30]}
10
11 df = pd.DataFrame(dades)
12 df = df.append({'Nom': 'Sara', 'Dept': 'ADMIN', 'DiesV': 3}, ignore_index = True)
13 df = df.append({'Nom': 'Joan', 'Dept': 'MANT', 'DiesV': 5}, ignore_index = True)
14 df = df.append({'Nom': 'Paco', 'Dept': 'PROD', 'DiesV': 7}, ignore_index = True)
15 df = df.append({'Nom': 'Susan', 'Dept': 'VENT', 'DiesV': 9}, ignore_index = True)
16 df = df.sort_values(['Dept', 'DiesV'])
17 print(df)
18
19
```

	Nom	Dept	DiesV
5	Sara	ADMIN	3
3	Rosa	ADMIN	43
1	Laura	ADMIN	55
6	Joan	MANT	5
2	David	MANT	20
7	Paco	PROD	7
4	Sam	PROD	30
0	Sònia	PROD	32
8	Susan	VENT	9

```
1 import pandas as pd
2 ventas = pd.DataFrame({'A': [41, 32, 56, 18],
3                          'B': [17, 54, 6, 78],
4                          'C': [12, 13, 16, 18] },
5                       index = ["Gen", "Feb", "Mar", "Abr"])
6 print (ventas)
7 print("SUMA")
8 print(ventas.sum(axis=1))
9 print(ventas.describe())
```

	A	B	C
Gen	41	17	12
Feb	32	54	13
Mar	56	6	16
Abr	18	78	18
SUMA			
Gen	70		
Feb	99		
Mar	78		
Abr	114		
dtype:	int64		
	A	B	C
count	4.000000	4.000000	4.000000
mean	36.750000	38.750000	14.750000
std	15.945219	33.260337	2.753785
min	18.000000	6.000000	12.000000
25%	28.500000	14.250000	12.750000
50%	36.500000	35.500000	14.500000
75%	44.750000	60.000000	16.500000
max	56.000000	78.000000	18.000000

```
1 import pandas as pd
2 import numpy as np
3
4 df = pd.read_csv('https://raw.githubusercontent.com/asalber/manual-python/master/datos/colesterol.csv')
5 df.info()
6 print(df)
7
8 df.dropna()
9 df.sort_values(by='colesterol', ascending=False).dropna()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14 entries, 0 to 13
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  -
0   nombre      14 non-null      object
1   edad        14 non-null      int64
2   sexo        14 non-null      object
3   peso        13 non-null      float64
4   altura      14 non-null      float64
5   colesterol  13 non-null      float64
dtypes: float64(3), int64(1), object(2)
memory usage: 800.0+ bytes
```

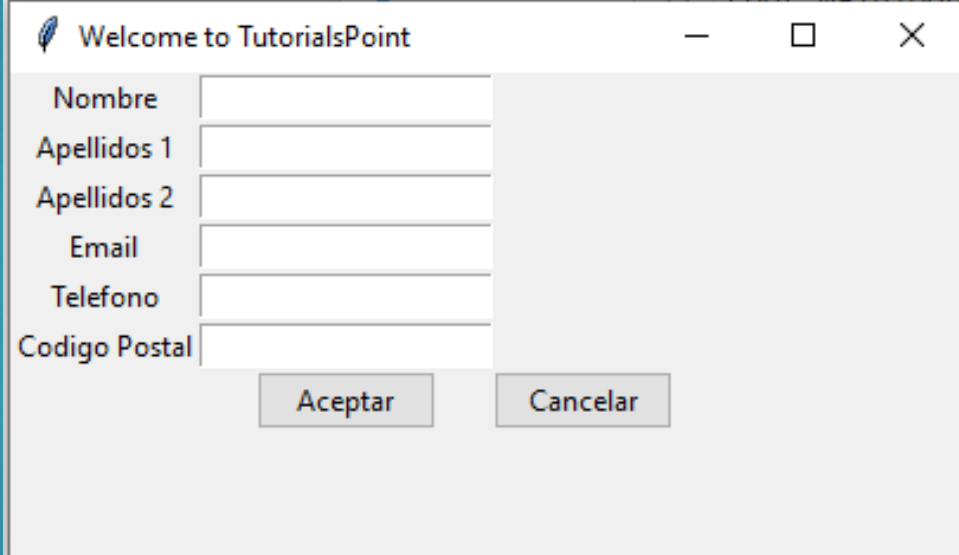
	nombre	edad	sexo	peso	altura	colesterol
0	José Luis Martínez Izquierdo	18	H	85.0	1.79	182.0
1	Rosa Díaz Díaz	32	M	65.0	1.73	232.0
2	Javier García Sánchez	24	H	NaI	1.81	191.0
3	Carmen López Pinzón	35	M	65.0	1.70	200.0
4	Marisa López Collado	46	M	51.0	1.58	148.0
5	Antonio Ruiz Cruz	68	H	66.0	1.74	249.0
6	Antonio Fernández Ocaña	51	H	62.0	1.72	276.0
7	Pilar Martín González	22	M	60.0	1.66	NaI
8	Pedro Gálvez Tenorio	35	H	90.0	1.94	241.0
9	Santiago Reillo Manzano	46	H	75.0	1.85	280.0
10	Macarena Álvarez Luna	53	M	55.0	1.62	262.0
11	José María de la Guía Sanz	58	H	78.0	1.87	198.0
12	Miguel Angel Cuadrado Gutiérrez	27	H	109.0	1.98	210.0
13	Carolina Rubio Moreno	20	M	61.0	1.77	194.0

Out[1]:

	nombre	edad	sexo	peso	altura	colesterol
9	Santiago Reillo Manzano	46	H	75.0	1.85	280.0
6	Antonio Fernández Ocaña	51	H	62.0	1.72	276.0
10	Macarena Álvarez Luna	53	M	55.0	1.62	262.0
5	Antonio Ruiz Cruz	68	H	66.0	1.74	249.0
8	Pedro Gálvez Tenorio	35	H	90.0	1.94	241.0
1	Rosa Díaz Díaz	32	M	65.0	1.73	232.0
12	Miguel Angel Cuadrado Gutiérrez	27	H	109.0	1.98	210.0
3	Carmen López Pinzón	35	M	65.0	1.70	200.0
11	José María de la Guía Sanz	58	H	78.0	1.87	198.0
13	Carolina Rubio Moreno	20	M	61.0	1.77	194.0
0	José Luis Martínez Izquierdo	18	H	85.0	1.79	182.0
4	Marisa López Collado	46	M	51.0	1.58	148.0

Tkinter

```
8 from tkinter import *
9 from tkinter import ttk
10
11 window = Tk()
12
13 window.title("Welcome to Tutorialspoint")
14 window.geometry('400x200')
15 window.minsize(width=400, height=200)
16
17 frame = Frame(window)
18
19 Label(frame ,text = "Nombre").grid(row = 0,column = 0)
20 Label(frame ,text = "Apellidos 1").grid(row = 1,column = 0)
21 Label(frame ,text = "Apellidos 2").grid(row = 2,column = 0)
22 Label(frame ,text = "Email").grid(row = 3,column = 0)
23 Label(frame ,text = "Telefono").grid(row = 4,column = 0)
24 Label(frame ,text = "Codigo Postal").grid(row = 5,column = 0)
25
26 Entry(frame).grid(row = 0,column = 1)
27 Entry(frame).grid(row = 1,column = 1)
28 Entry(frame).grid(row = 2,column = 1)
29 Entry(frame).grid(row = 3,column = 1)
30 Entry(frame).grid(row = 4,column = 1)
31 Entry(frame).grid(row = 5,column = 1)
32
33 btn = ttk.Button(frame ,text="Aceptar").grid(row=6,column=1)
34 frame.pack(anchor=NW, expand=1)
35
36 btn = ttk.Button(frame ,text="Cancelar").grid(row=6,column=2)
37 frame.pack(anchor=NW, expand=1)
38
39 window.mainloop()
```



The screenshot shows a Tkinter window titled "Welcome to Tutorialspoint". The window contains a form with the following elements:

- Labels: "Nombre", "Apellidos 1", "Apellidos 2", "Email", "Telefono", and "Codigo Postal".
- Entry fields: Six text input fields corresponding to the labels above.
- Buttons: Two buttons labeled "Aceptar" and "Cancelar" at the bottom right.

JSON,XML

```
1 import requests
2 import json
3
4 response = requests.get('https://gorest.co.in/public/v2/users')
5
6 print(response.json())
7
8 for data in response.json():
9     print(data['email'], data['name'])
10
```

```
[{'id': 2568, 'name': 'Rajan Joshi', 'email': 'rajan_joshi@wisoky.io', 'gender': 'male', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.wisoky.io', 'social_media': {'facebook': 'https://www.facebook.com/wisoky.io', 'twitter': 'https://www.twitter.com/wisoky.io', 'instagram': 'https://www.instagram.com/wisoky.io', 'youtube': 'https://www.youtube.com/wisoky.io'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2569, 'name': 'Prathamesh Patel IV', 'email': 'patel_iv_prathamesh@haley.com', 'gender': 'female', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.haley.com', 'social_media': {'facebook': 'https://www.facebook.com/haley.com', 'twitter': 'https://www.twitter.com/haley.com', 'instagram': 'https://www.instagram.com/haley.com', 'youtube': 'https://www.youtube.com/haley.com'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2570, 'name': 'Shwet Mehra', 'email': 'shwet_mehra@mcclure-marvin.biz', 'gender': 'male', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.mcclure-marvin.biz', 'social_media': {'facebook': 'https://www.facebook.com/mcclure-marvin.biz', 'twitter': 'https://www.twitter.com/mcclure-marvin.biz', 'instagram': 'https://www.instagram.com/mcclure-marvin.biz', 'youtube': 'https://www.youtube.com/mcclure-marvin.biz'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2571, 'name': 'Msgr. Shwet Kaul', 'email': 'shwet_msgr_kaul@damore.biz', 'gender': 'female', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.damore.biz', 'social_media': {'facebook': 'https://www.facebook.com/damore.biz', 'twitter': 'https://www.twitter.com/damore.biz', 'instagram': 'https://www.instagram.com/damore.biz', 'youtube': 'https://www.youtube.com/damore.biz'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2572, 'name': 'Akula Bhattacharya', 'email': 'akula_bhattacharya@wiegand-wilkinson.name', 'gender': 'female', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.wiegand-wilkinson.name', 'social_media': {'facebook': 'https://www.facebook.com/wiegand-wilkinson.name', 'twitter': 'https://www.twitter.com/wiegand-wilkinson.name', 'instagram': 'https://www.instagram.com/wiegand-wilkinson.name', 'youtube': 'https://www.youtube.com/wiegand-wilkinson.name'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2573, 'name': 'Bhaaswar Mahajan', 'email': 'bhaaswar_mahajan@stracke.info', 'gender': 'female', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.stracke.info', 'social_media': {'facebook': 'https://www.facebook.com/stracke.info', 'twitter': 'https://www.twitter.com/stracke.info', 'instagram': 'https://www.instagram.com/stracke.info', 'youtube': 'https://www.youtube.com/stracke.info'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2574, 'name': 'Smriti Gill', 'email': 'gill_smriti@mosciski.biz', 'gender': 'male', 'status': 'inactive', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.mosciski.biz', 'social_media': {'facebook': 'https://www.facebook.com/mosciski.biz', 'twitter': 'https://www.twitter.com/mosciski.biz', 'instagram': 'https://www.instagram.com/mosciski.biz', 'youtube': 'https://www.youtube.com/mosciski.biz'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2575, 'name': 'Rajan Joshi', 'email': 'rajan_joshi@wisoky.io', 'gender': 'male', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.wisoky.io', 'social_media': {'facebook': 'https://www.facebook.com/wisoky.io', 'twitter': 'https://www.twitter.com/wisoky.io', 'instagram': 'https://www.instagram.com/wisoky.io', 'youtube': 'https://www.youtube.com/wisoky.io'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2576, 'name': 'Prathamesh Patel IV', 'email': 'patel_iv_prathamesh@haley.com', 'gender': 'female', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.haley.com', 'social_media': {'facebook': 'https://www.facebook.com/haley.com', 'twitter': 'https://www.twitter.com/haley.com', 'instagram': 'https://www.instagram.com/haley.com', 'youtube': 'https://www.youtube.com/haley.com'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2577, 'name': 'Shwet Mehra', 'email': 'shwet_mehra@mcclure-marvin.biz', 'gender': 'male', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.mcclure-marvin.biz', 'social_media': {'facebook': 'https://www.facebook.com/mcclure-marvin.biz', 'twitter': 'https://www.twitter.com/mcclure-marvin.biz', 'instagram': 'https://www.instagram.com/mcclure-marvin.biz', 'youtube': 'https://www.youtube.com/mcclure-marvin.biz'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2578, 'name': 'Msgr. Shwet Kaul', 'email': 'shwet_msgr_kaul@damore.biz', 'gender': 'female', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.damore.biz', 'social_media': {'facebook': 'https://www.facebook.com/damore.biz', 'twitter': 'https://www.twitter.com/damore.biz', 'instagram': 'https://www.instagram.com/damore.biz', 'youtube': 'https://www.youtube.com/damore.biz'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2579, 'name': 'Akula Bhattacharya', 'email': 'akula_bhattacharya@wiegand-wilkinson.name', 'gender': 'female', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.wiegand-wilkinson.name', 'social_media': {'facebook': 'https://www.facebook.com/wiegand-wilkinson.name', 'twitter': 'https://www.twitter.com/wiegand-wilkinson.name', 'instagram': 'https://www.instagram.com/wiegand-wilkinson.name', 'youtube': 'https://www.youtube.com/wiegand-wilkinson.name'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2580, 'name': 'Bhaaswar Mahajan', 'email': 'bhaaswar_mahajan@stracke.info', 'gender': 'female', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.stracke.info', 'social_media': {'facebook': 'https://www.facebook.com/stracke.info', 'twitter': 'https://www.twitter.com/stracke.info', 'instagram': 'https://www.instagram.com/stracke.info', 'youtube': 'https://www.youtube.com/stracke.info'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2581, 'name': 'Smriti Gill', 'email': 'gill_smriti@mosciski.biz', 'gender': 'male', 'status': 'inactive', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.mosciski.biz', 'social_media': {'facebook': 'https://www.facebook.com/mosciski.biz', 'twitter': 'https://www.twitter.com/mosciski.biz', 'instagram': 'https://www.instagram.com/mosciski.biz', 'youtube': 'https://www.youtube.com/mosciski.biz'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2582, 'name': 'Darshwana Nayar', 'email': 'nayar_darshwana@connelly.co', 'gender': 'female', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.connelly.co', 'social_media': {'facebook': 'https://www.facebook.com/connelly.co', 'twitter': 'https://www.twitter.com/connelly.co', 'instagram': 'https://www.instagram.com/connelly.co', 'youtube': 'https://www.youtube.com/connelly.co'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2583, 'name': 'Sitara Kaniyar IV', 'email': 'iv_sitara_kaniyar@zemlak-roob.biz', 'gender': 'female', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.zemlak-roob.biz', 'social_media': {'facebook': 'https://www.facebook.com/zemlak-roob.biz', 'twitter': 'https://www.twitter.com/zemlak-roob.biz', 'instagram': 'https://www.instagram.com/zemlak-roob.biz', 'youtube': 'https://www.youtube.com/zemlak-roob.biz'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2584, 'name': 'Eekalabya Johar', 'email': 'johar_eekalabya@bernhard.net', 'gender': 'male', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.bernhard.net', 'social_media': {'facebook': 'https://www.facebook.com/bernhard.net', 'twitter': 'https://www.twitter.com/bernhard.net', 'instagram': 'https://www.instagram.com/bernhard.net', 'youtube': 'https://www.youtube.com/bernhard.net'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}, {'id': 2585, 'name': 'Krishnadasa Panicker', 'email': 'krishnadasa_panicker@koepp.co', 'gender': 'male', 'status': 'active', 'email_verified': True, 'phone_number': '9876543210', 'password': '1234567890', 'profile_picture': 'https://i.pinimg.com/564x/00/00/00/00000000000000000000000000000000.jpg', 'bio': 'I am a software engineer and I love to learn new things.', 'website': 'https://www.koepp.co', 'social_media': {'facebook': 'https://www.facebook.com/koepp.co', 'twitter': 'https://www.twitter.com/koepp.co', 'instagram': 'https://www.instagram.com/koepp.co', 'youtube': 'https://www.youtube.com/koepp.co'}, 'created_at': '2023-01-01T00:00:00Z', 'updated_at': '2023-01-01T00:00:00Z'}]
```

```
1 import xml.etree.ElementTree
2
3 response = requests.get("https://www.washingtonpost.com/arcio/news-sitemap/")
4
5 string_xml = response.content
6 tree = xml.etree.ElementTree.fromstring(string_xml)
7 xml.etree.ElementTree.dump(tree)
```

```
1 import requests
2 import json
3
4 response = requests.get('https://gorest.co.in/public/v2/users')
5 # print(response.json())
6
7 for data in response.json():
8     if data['gender'] == 'female' and data['status'] == 'active':
9         print(data['name'], data['email'])
10
11
```

Tanushri Pilla tanushri_pilla@effertz.net
Nirbhay Achari nirbhay_achari@kerluke.co
Chandraketu Reddy chandraketu_reddy@haley.net

SQL

```
1 import sqlite3
2
3 def conectar_usuarios():
4     """ abre la conexion con la base de datos """
5
6     db_name = "../dat/practica2.db"
7     conexion = None
8
9     try :
10         conexion = sqlite3.connect(db_name)
11     except Exception as e:
12         print("Error en la base de datos: ", e)
13     return (conexion)
14
15 def crear_tabla_usuarios():
16     """ si no existe, crea la tabla de usuarios """
17
18     conexion = conectar_usuarios()
19     cursor = conexion.cursor()
20
21     # Ahora crearemos una tabla de usuarios con nombres, edades y emails
22     sql = "CREATE TABLE IF NOT EXISTS usuarios " \
23           "(codigo CHAR(6) PRIMARY KEY, nombre VARCHAR(100), activo CHAR(1), email VARCHAR(100), depto CHAR(3))"
24     cursor.execute(sql)
25     conexion.commit() # Guardar cambios
26     conexion.close()
27
28 def crear_usuarios(lista_usuarios):
29     """ inserta un array de tuplas de usuarios en la base de datos """
30
31     conexion = conectar_usuarios()
32     cursor = conexion.cursor()
33
34     # Ahora utilizamos el método executemany() para insertar varios
35     cursor.executemany("INSERT INTO usuarios VALUES (?, ?, ?, ?, ?)", lista_usuarios)
36     conexion.commit() # Guardar cambios
37     conexion.close()
```

```
72 def consultar_usuarios() :
73     """ retorna una lista de tuplas de usuarios """
74
75     conexion = conectar_usuarios()
76     cursor = conexion.cursor()
77
78     # Recuperamos Los registros de la tabla de usuarios
79     cursor.execute("SELECT * FROM usuarios")
80
81     # Recorremos todos Los registros con fetchall
82     # y Los volcamos en una lista de usuarios
83     personal = cursor.fetchall()
84     conexion.close()
85     return (personal)
86
87 def consultar_usuario(codigo) :
88     """ retorna una lista de 1 tupla con el usuario encontrado """
89
90     conexion = conectar_usuarios()
91     cursor = conexion.cursor()
92
93     # Recuperamos Los registros de la tabla de usuarios
94     cursor.execute(f"SELECT * FROM usuarios WHERE codigo = '{codigo}'")
95
96     # Recuperamos todos Los registros y Los volcamos en una lista de usuarios
97     personal = cursor.fetchall()
98     conexion.close()
99     return (personal)
100
101 if __name__ == '__main__':
102     crear_tabla_usuarios()
103     eliminar_todo()
104     # Crea datos de prueba
105
106     crear_usuario (('0', 'Pruebas', 'S/N', 'pruebas@ejemplo.com', 'DEPTO'))
107     # Creamos una lista con varios usuarios (codigo, nombre, edad, mail)
108     # =====
109     usuarios = [('1946', 'Mario Benet', 'S', 'mario@ejemplo.com', 'MANT'), \
110                ('4622', 'Marta Artigas', 'N', '5589@telefonica.es', 'ADMIN'), \
111                ('3717', 'Joan Piqué', 'S', 'joan@dosnoms.cat', 'ADMIN')]
```

```
39 def crear_usuario(usuario):
40     """ inserta una tupla de usuario en la base de datos """
41
42     conexion = conectar_usuarios()
43     cursor = conexion.cursor()
44
45     sql = "INSERT INTO usuarios(codigo, nombre, activo, email, depto) VALUES (?, ?, ?, ?, ?);"
46     print(sql, usuario)
47     cursor.execute (sql, usuario)
48     conexion.commit() # Guardar cambios
49     conexion.close()
50
51 def eliminar_usuario(codigo):
52     """ elimina un usuario """
53
54     conexion = conectar_usuarios()
55     cursor = conexion.cursor()
56
57     sql = f"DELETE FROM usuarios WHERE codigo = '{codigo}';"
58     cursor.execute (sql)
59     conexion.commit() # Guardar cambios
60     conexion.close()
61
62 def eliminar_todo():
63     """ elimina todos los usuarios """
64
65     conexion = conectar_usuarios()
66     cursor = conexion.cursor()
67
68     cursor.execute ("DELETE FROM usuarios")
69     conexion.commit() # Guardar cambios
70     conexion.close()
```

```
113 # # =====
114 crear_usuarios (usuarios)
115 eliminar_usuario('0')
116
117 for usuario in consultar_usuarios():
118     print(usuario)
```

```
INSERT INTO usuarios(codigo, nombre, activo, email, depto) VALUES (?, ?, ?, ?, ?); ('0', 'Pruebas', 'S/N', 'pruebas@ejemplo.com',
'DEPTO')
('1946', 'Mario Benet', 'S', 'mario@ejemplo.com', 'MANT')
('4622', 'Marta Artigas', 'N', '5589@telefonica.es', 'ADMIN')
('3717', 'Joan Piqué', 'S', 'joan@dosnoms.cat', 'ADMIN')
```