

## Casos de Estudio

Analice, diseñe y codifique los siguientes enunciados en Python

- Una institución bancaria tiene una aplicación que recibe una lista con los movimientos de las cuentas bancarias de sus clientes mediante un servicio externo. Para cada cuenta se recibe el número de cuenta, nombre y apellido del cliente, importe, fecha del movimiento y el tipo de movimiento (E=Extracción, D=Depósito). El formato de cada cuenta está dado por una cadena de caracteres donde cada atributo de una cuenta se encuentra separado por comas.

Por ejemplo:

```
27200123456,MARIA FERNANDEZ,0000500000,30-05-2021,E
27200321654,CARLOS TORRES,0000400045,31-05-2021,D
27200987125,LAURA AQUINO,0000230000,30-05-2021,D
27200852369,MARTIN ESTRADA,0000700000,25-05-2021,E
27200123456,MARIA FERNANDEZ,0000250099,31-05-2021,E
27200795169,FLORENCIA ROBLES,0000350080,20-05-2021,D
```

Tenga en cuenta que:

- Los 2 últimos dígitos del importe corresponden a los centavos, o sea que el valor 0000500000 es equivalente a 5000 pesos.
- La fecha solo tiene carácter informativo por lo que debe guardarse como un string.

Se solicita lo siguiente:

1. Procesar la lista recibida y guardar los datos recibidos en una lista.
2. Mostrar la lista con las cuentas obtenidas.
3. Mostrar los movimientos correspondientes a depósitos y el total acumulado.
4. Mostrar los movimientos de una cuenta.

*Nota: utilice los datos del ejemplo para validar el programa*

## Ejercicios

1. Analizar, realizar la prueba de escritorio y mostrar la salida de los siguientes programas:

<p>Texto: '192.168.0.1' separador = '.'</p> <pre>def s_sep(cadena, car):     lpalabras = list()     if car in cadena:         lpalabras = cadena.split(car)     else:         print('Ingreso inválido')     return lpalabras  mensaje = "" Ingrese un texto y luego un caracter separador por ejm: Texto: 1-2-3-4 Separador: - """ print(mensaje) txt = input('Texto:') separador = input('Separador:') lista = s_sep(txt, separador) print(lista)</pre>	<p>edad = 'uno' , -10, 0</p> <pre>def lee_edad():     """Solicita una edad"""     i = 0     while i &lt; 3:         i += 1         valor = input(f'Intento {i}, edad:')         try:             edad = float(valor)             if edad &lt;= 0.0:                 print('La edad debe ser &gt; 0')             else:                 return edad         except ValueError:             print('Dato inválido')     print(f'Incorrecto! {i} intentos')  edad = lee_edad() print(edad)</pre>
--	--

2. Analizar y ejecutar el siguiente código que tiene tres funciones que realizan la misma tarea: contar la cantidad de vocales que tiene una frase. Investigar en internet las instrucciones que no conozca.

```
def contar_vocales1(cadena):
    contador = 0
    for letra in cadena:
        if letra.lower() in 'aeiouáéíóúü':
            contador += 1
    return contador

def contar_vocales2(texto):
    return sum(map(texto.lower().count, 'aeiouáéíóúü'))

import re
def contar_vocales3(texto):
    return len(re.findall('[aeiouáéíóúü]', texto, re.IGNORECASE))

cad = input('Ingrese una frase:') #Pájaro de mal agüero'
print(contar_vocales1(cad))
print(contar_vocales2(cad))
print(contar_vocales3(cad))
```

3. Analizar, ejecutar y mejorar el siguiente código.

```
import random
def bienvenida(mensaje):
    bienv_inp = ['hola', 'buenas', 'que tal', 'hi', 'buenos', 'buen', 'todo']
    bienv_out = ['Hola, ¿cómo estás?', 'Hola, ¿qué tal?', 'Hola, ¿te puedo ayudar?', 'Hola, encantado de comunicarme contigo', 'Hola, espero que estés bien!']

    for palabra in mensaje.split():
        if palabra.lower() in bienv_inp:
            return random.choice(bienv_out)
        else:
            return 'Tu mensaje no está en mi base de datos'

cad = input('Soy un robot, salúdame por favor: ') # Hola robot
print(bienvenida(cad))
```

4. Ingresar una dirección de email por teclado. Verificar que la cadena ingresada contiene un solo caracter "@"
5. Ingresar una frase por teclado, se desea modificar todos los espacios en blanco dobles y dejar solo un espacio en blanco.
6. Solicitar el ingreso de una palabra clave por teclado. Controlar que la cadena ingresada tenga más de 6 caracteres y al menos un número y un caracter en mayúsculas para que sea válido, en caso contrario mostrar un mensaje de error.
7. Escribir un módulo que tiene un parámetro que es una cadena de caracteres devuelve: La primera letra de cada palabra de la cadena en mayúsculas. Por ejemplo, si recibe "universal serial bus" debe devolver "USB".

8. Escribir un módulo que reciba una cadena de unos y ceros (es decir, un número en representación binaria) y devuelva el valor decimal correspondiente.
9. Escribir un módulo denominado `ins_car_xd(cadena, car, x, d)`, devuelve una cadena que inserta un caracter en la cadena cada `x` cantidad de caracteres de la cadena, siendo el parámetro `d` la dirección desde dónde se comienza la inserción, `d` puede ser "i" o "f", si `d` es "i" es desde el inicio y si `d` es "f" es desde el fin, por defecto `d` = "i" si no se especifica este parámetro. Ejemplo: `cadena` = "2552552550" `car` = "." y `x` = 3 el módulo debería devolver "255.255.255.0" otro ejemplo `cadena` = "1234567890" `car` = "," `x` = 3 y `d` = "f" el módulo debería devolver "1,234,567,890"
10. Hacer un módulo que acepte una cadena original y devuelve otra cadena que es el gerigoso de la cadena original. Utilizar una iteración sobre la cadena para agregar la sílaba 'pa', 'pe', 'pi', 'po', o 'pu' según corresponda luego de cada vocal de la cadena original.
11. Diseñar un programa que permita realizar el registro a una aplicación de estudiantes de una universidad, cada estudiante cuenta con los siguientes atributos: **dni**, **clave**, **nombre**, **eMail** y **número de celular**. Se solicita desarrollar las siguientes funcionalidades:
  - a. **Registrarse en la aplicación:** Se solicitan los datos del estudiante con las siguientes validaciones:
    - **dni:** es de tipo string debe tener 8 dígitos numéricos, luego de ingresar este valor se debe verificar que no exista actualmente.
    - **clave:** debe tener como mínimo una longitud de 6 caracteres, al menos una mayúscula y un número
    - **nombre:** no puede tener números ni caracteres especiales y al menos 3 caracteres
    - **eMail:** debe verificar que tenga el @ y que no se encuentre al inicio ni al final de la cadena ingresada.
    - **Número de Celular:** debe incluir el código de área más el número sin el 15 (por ejemplo 388-4800123), es decir:
      - Los primeros dígitos corresponden al código de área y los valores solo pueden ser 388, 3886, 3887 y 3888
      - El guión es obligatorio
      - Los restantes números corresponden al número de celular y su longitud debe ser de 7 dígitos.
  - b. **Buscar un estudiante:** el programa solicitará el nombre y en función de ello mostrará lo siguiente:
    - Si se ingresa un vacío (enter), el programa mostrará la lista completa de estudiantes registrados
    - Si se ingresa un nombre, el programa mostrará los estudiantes cuyos nombres contengan al valor ingresado, por ejemplo si ingresa ana, mostrará Mariana, Analía, etc.
  - c. **Ingresar (Login):** debe solicitar el dni y la clave y verificar que coincida con algún estudiante registrado, si no existe emitirá el mensaje "Usuario o Clave incorrectos". Si los datos son correctos mostrará el mensaje "Ingreso exitoso" y también mostrará el nombre y correo electrónico del estudiante.