

# Ejercicio

Estructuras de control y tipos  
de datos



Fundamentos de Python



## 1B. Ejercicio. Estructuras de control y tipos de datos

En esta actividad pondrás en práctica tus habilidades para la solución de problemas básicos utilizando algoritmos, lo que te permitirá familiarizarte con la herramienta y su lógica. Para ello, planteamos situaciones ordinarias que encontrarás cuando escribas código.

Primero, deberás **analizar los ejercicios**. Luego, será necesario diseñar un plan de trabajo, considerando algoritmos que resuelvan cada una de las secciones, tomando en cuenta las condiciones indicadas. Finalmente, ejecuta un algoritmo a la vez y realiza pruebas que validen su funcionamiento. Para esta actividad será oportuno utilizar diferentes **estructuras de control** y **algoritmos**.

### Ejercicio 1. Cajero automático

Has sido contratado por un banco que está en proceso de actualizar el software de sus cajeros automáticos. Tu tarea es desarrollar un algoritmo básico que simule el funcionamiento de un cajero automático, tomando como referencia las siguientes condiciones:

- Entregar siempre la menor cantidad de billetes posible.
- Mantener un inventario que indique cuál es la cantidad de billetes disponible por cada denominación.
- Las denominaciones disponibles son: 50, 100, 200, 500 y 1000.
- Si el inventario no cuenta con una combinación de billetes suficientes para satisfacer el importe solicitado, no dispensará ningún billete.



- Siempre que inicie el algoritmo, deberá haber en inventario 10 billetes de cada denominación.

## Desarrollo inicial

Para facilitar tu proceso de resolución, te proporcionamos el inicio del desarrollo del ejercicio. A partir de ese punto, deberás continuar y completar el resto del algoritmo aplicando lo aprendido.

1. **Creación de las variables que representan el inventario por cada denominación de billete.** En este paso, se definen variables para indicar cuántos billetes hay disponibles de cada tipo. Cada variable almacenará un número entero (*int*), y en este caso, se asigna el **valor** inicial de 10 a cada una.

```
# Variables para contar billetes
billetes_1000 = 10
billetes_500 = 10
billetes_200 = 10
billetes_100 = 10
billetes_50 = 10
billetes_20 = 10
```

2. **Creación de las variables para el número de billetes por cada denominación.** Con esto, le decimos a Python que cada una de las variables será declarada con 0, debido a que aún no se ha determinado cuántos billetes serán necesarios.



```
# Variables para billetes a entregar
entregar_1000 = 0
entregar_500 = 0
entregar_200 = 0
entregar_100 = 0
entregar_50 = 0
entregar_20 = 0
monto_restante = monto
```

3. **Mensaje de inicio al sistema y solicitud del monto a retirar del cajero.** En este paso, se muestra un mensaje en la consola indicando que el sistema del cajero ha iniciado. Después, se solicita al usuario que ingrese la cantidad que desea retirar en número entero (*int*) y es almacenado en una variable.

```
# iniciamos el sistema
print("\n--- Dispensadora de Billetes ---")
# Solicitar monto
print("\nIngrese el monto a retirar (0 para salir): ")
entrada = input()
#transformamos el valor dato a un tipo de dato entero
monto = int(entrada)
```

Ahora, deberás continuar con la lógica del programa, para que funcione de acuerdo a las condiciones mencionadas.



## Ejercicio 2. Inicio de sesión

En una empresa se quiere crear un sistema de inicio de sesión con un número limitado de intentos. El sistema debe pedir al usuario su nombre de usuario y contraseña. Para crear dicho sistema, ten en cuenta las siguientes condiciones:

- Se permite un máximo de 3 intentos.
- Mostrar un error al usuario cuando no exista uno de los datos (usuario y contraseña).
- Si uno de los campos (usuario y contraseña) está vacío deberá mostrar un error de autenticación.
- Únicamente usar estructuras de control.

### Desarrollo inicial

Para facilitar tu proceso de resolución, te proporcionamos el inicio del desarrollo del ejercicio. A partir de ese punto, deberás continuar y completar el resto del algoritmo aplicando lo aprendido.

- 1. Variables que almacenan al usuario y contraseña a las que se les dará el acceso.** En este paso, se crean las variables que guardarán los datos de acceso autorizados. Estas variables contendrán los valores que el sistema usará para verificar si el usuario y la contraseña ingresados son correctos.

```
# Credenciales válidas (como variables individuales)
usuario_correcto = "admin"
clave_correcta = "1234"
```



- 2. Mensaje de inicio al sistema, intentos que tienes, solicitud de usuario y contraseña.** El sistema muestra un mensaje en la consola indicando que ha iniciado, informa al usuario cuántos intentos tiene para iniciar sesión y solicita el ingreso del valor del usuario y contraseña.

```
print("\n--- Sistema de Inicio de Sesión ---")
print(f"Intentos restantes: {intentos}")

# Solicitar credenciales
print("Usuario: ")
usuario = input()
print("Contraseña: ")
clave = input()
```

Ahora, deberás continuar con la lógica del programa, para que funcione de acuerdo a las condiciones mencionadas.