

Ejercicio

Funciones y estructuras de
datos



Fundamentos de Python



2A. Ejercicio. Funciones y estructuras de datos

En esta actividad pondrás en práctica tus habilidades para la solución de problemas básicos utilizando algoritmos, lo que te permitirá identificar las principales estructuras de datos y funciones en Python. Para ello, planteamos situaciones ordinarias que encontrarás cuando escribas código.

Primero, deberás **analizar** los ejercicios. Luego, **implementa** 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 datos** y **funciones**, hasta llegar a la solución total.

Este ejercicio de ejemplo será tu punto de partida para resolver los problemas posteriores. El código incluye comentarios (#) que explican cada paso del proceso, lo que facilitará su comprensión y te permitirá adaptarlo a las diferentes situaciones planteadas.

Ejercicio de ejemplo

Eres parte del equipo de desarrollo de un hotel que necesita un sistema básico para gestionar reservas y la liberación de habitaciones. El hotel cuenta con dos tipos de habitaciones: individual y doble, cada una con disponibilidad limitada. Tu trabajo es desarrollar el código con los siguientes requisitos del programa:



- Habitaciones individuales disponibles: 3
- Habitaciones dobles disponibles: 2
- Reservar: Asignar una habitación a un cliente (si hay disponibilidad).
- Liberar: Devolver una habitación al inventario.

Solución usando estructuras de datos y funciones

```
#Se crea un función llamada hotel, aquí vamos a hacer el sistema de las reservaciones del mismo
def hotel():

    #Usamos la estructura diccionario, donde se almacenan las habitaciones de camas individuales y dobles disponibles
    habitaciones = {"individual": 3, "doble": 2}

    #Usamos un while para dejar ciclo infinito, que nos permitirá usar las veces que queramos el sistema
    while True:

        #Declaramos una variable como entrada de valor, dónde se guardará el mismo
        accion = input("¿Reservar o liberar? ")

        #Con la variante anterior se ocupará un Match que nos permita escoger una acción entre reservar o liberar
        match accion:
            case "reservar":

                #Para reservar pedimos qué tipo de habitación quiere el cliente y se almacena en una variable
                print("qué tipo de habitación quiere")
                tipo = input("Tipo: ")
```



```
#Se ocupa un condicional if para preguntar si el tipo de habitación se encuentra en el diccionario "habitaciones" y si la cantidad es mayor a 0
    if tipo in habitaciones and habitaciones[tipo] > 0:

#Cuando se cumpla la condición se le restará -1 a la cantidad del tipo de habitación que se escogió y se manda mensaje que su habitación fue reservada.
        habitaciones[tipo] -= 1
        print(f"Habitación {tipo} reservada")
    else:

#Si la condición no se cumple manda este mensaje
        print("No disponible")

    case "liberar":

#Para liberar pedimos qué tipo de habitación quiere liberar y se almacena en una variable
        print ("que tipo de habitación quiere")
        tipo = input("Tipo: ")

#Cuando se cumpla la condición se le suma 1 a la cantidad del tipo de habitación que se escogió y se manda mensaje que su habitación fue liberada
        if tipo in habitaciones:
            habitaciones[tipo] += 1
            print(f"Habitación {tipo} liberada")

#Este código sirve para cerrar el sistema
    case "salir":
        break
```



```
#Este código sirve para que, cuando se coloca un dato que no  
corresponda a los anteriores casos, mande un mensaje  
  
    case _:  
        print("Acción no reconocida")  
  
#Imprime las habitaciones disponibles  
    print("Estado:", habitaciones)  
  
#Se escribe el nombre de la función que se va a llamar a  
ejecutar  
hotel()
```

Ejecución del código:

Para probar el funcionamiento del programa, cuando se ejecute, escribe en la consola una de las opciones disponibles: reservar, liberar o salir, y presiona *Enter*. Si escribes *reservar* o *liberar*, el programa seguirá preguntando el tipo de habitación y actualizando el inventario.

Para reservar:

```
¿Reservar o liberar? reservar  
que tipo de habitacion quiere  
Tipo: individual  
Habitación individual reservada  
Estado: {'individual': 2, 'doble': 2}
```

Para liberar:



```
¿Reservar o liberar? liberar  
que tipo de habitacion quiere  
Tipo: doble  
Habitación doble liberada  
Estado: {'individual': 2, 'doble': 3}  
¿Reservar o liberar? █
```

Si escribes *salir*, el programa termina.

Con base en este ejemplo, resuelve los ejercicios restantes, adaptando la estructura y el flujo del programa a las siguientes situaciones:

Ejercicio 1. Calculadora de edad

Trabajas en una compañía que ofrece seguros de vida y de gastos médicos. La empresa está desarrollando una herramienta tecnológica, para que los agentes puedan calcular el monto de una póliza. Para dicho cálculo, se consideran varios factores de riesgo, entre los que se encuentra la edad. Por ello, el contratante debe llenar un formulario, incluyendo su fecha de nacimiento. Como primer reto, deberás calcular la edad del cliente al momento de la cotización.

- La captura de la fecha de nacimiento debe ser en una sola cadena de texto con formato DD-MM-AAAA.
- Validar que sea una fecha existente, si supera los 31 días o menor a 1 día de un mes, se deberá marcar como inválido.
- El año de nacimiento debe ser mayor de 1900.
- Validar si el cliente ya cumplió años en la fecha de captura o aún no.



- Validar que no se puede dejar información en blanco o campos vacíos.

Ejercicio 2. Sistema de evaluación

En una escuela se quiere crear un sistema de evaluación para profesores que permita la obtención de los alumnos aprobados y reprobados de acuerdo a su historial académico, dicho sistema ayudará a dar de forma rápida todos los datos del alumno, siguiendo las siguientes condiciones:

- Deberá pedir el número de alumnos y materias, pedir el nombre del alumno, y su matrícula.
- El sistema debe evaluar, como condición de evaluación, que si es una calificación mayor a 6, es aprobado y de lo contrario, reprobado.
- Validar que no se puede dejar información en blanco o campos vacíos.