ACTIVITAT AVALUABLE AC6

Mòdul: MP14-Big Data
UF: UF1 – Big Data
Professor: Albert Guardiola
Data d'entrega: 27/11/2022
Mètode d'entrega: Clickedu.
Instruccions:

El ejercicio debe realizarse sobre el código adjunto. Sólo se puede añadir código en aquellas secciones marcadas para ese propósito ("# CÓDIGO AQUÍ"). No se puede modificar el prototipo de las funciones (ni sus parámetros ni su valor de retorno).

Resultats de l'aprenentatge:
RA2. Programa pel Big data a partir de data sources.

Enlace del programa:
https://github.com/JCARL0/Big-Data/tree/main/MP14-UF1-AC6

TAREA 1. En este ejercicio vamos a realizar un sencillo programa de gestión de las reservas de un hotel.
El hotel consta de 4 pisos de habitaciones, con cinco habitaciones en cada piso.

a)Crea una variable *hotel* que tenga el siguiente tipo complejo. En todos los pisos, la habitación X1 es de tipo *"suite"*, y el resto del tipo *"normal"*. Inicializa todas las habitaciones como desocupadas, con fecha de ingreso vacía (cadena de texto vacía), con 0 noches reservadas y con precio de reserva a 0.
-La información relacionada con la reserva de cada habitación debe estar almacenada en un diccionario:

habitación = { numero: 23, (tipo entero)
ocupada: True, (tipo booleano)
fechaIngreso: "20221121", (tipo string)
numNoches: 4, (tipo entero)
clase: "normal", (tipo string)
precio: 200.0 } (tipo decimal)

-Las distintas habitaciones de cada piso deben ir almacenadas en una lista: *piso = [habitacion1, habitacion2 ...]*, donde cada habitacion es del tipo diccionario que se ha descrito.
-Los distintos pisos deben ir almacenados, a su vez, en una nueva lista: *reservas = [piso1, piso2 ...]*, donde cada piso es la lista descrita en el punto anterior.
Es decir: la variable *hotel* con la que trabajará nuestro programa será una lista de listas, siendo esta segunda lista una lista de diccionarios.

```
for piso, habitaciones in enumerate(hotel):                    #
iteramos pisos
    for numHab, habitacion in enumerate(habitaciones):         #
iteramos habitaciones
        habitacion["numero"] = (piso+1)*10 + (numHab+1)        #
inicializamos cada habitación
```

```
            habitacion["ocupada"] = False
            habitacion["fechaIngreso"] = ""
            habitacion["numNoches"] = 0
            if numHab + 1 == 1:
                habitacion["clase"] = "suite"
            else:
                habitacion["clase"] = "normal"
            habitacion["precio"] =0.0
```

Implementa las siguientes funciones:

b)La función *registrarReserva*(), a la que se le pase el número de habitación, la fecha de ingreso y las noches de estadía, rellene los datos en la variable *reservas*. La función debe calcular y almacenar automáticamente el precio de la reserva.
(Las habitaciones normales tienen un precio de 50 euros la noche; las suites, 100 euros la noche)

```
def hacerReserva(piso,hab,fechaIngreso,numNoches):
  habitacion = hotel[piso-1][hab-1]
  habitacion["ocupada"] = True
  habitacion["fechaIngreso"] = fechaIngreso
  habitacion["numNoches"] = numNoches
  if hab == 1:
    habitacion["precio"] = habitacion["numNoches"] * 100
  else:
    habitacion["precio"] = habitacion["numNoches"] * 50
  return None
```

c)La función *consultarReserva()*, a la que se pase el número de habitación y muestre por pantalla los datos de la reserva, o un mensaje informativo si no hay reserva en esa habitación. *(2 punts)*

```
def consultarReserva(numHab):
  habitacion = None
  for piso in hotel:
    for hab in piso:
      if hab["numero"] == numHab and hab["ocupada"]:
        habitacion = hab
        break

  if habitacion:
    print("Datos de la reserva:")
    print(f"Número de habitación: {habitacion['numero']}")
    print(f"Fecha de ingreso: {habitacion['fechaIngreso']}")
    print(f"Noches reservadas: {habitacion['numNoches']}")
    print(f"Tipo de habitación: {habitacion['clase']}")
    print(f"Precio de reserva: {habitacion['precio']}")
  else:
    print("Habitación sin reserva")
```

d)La función *anularReserva()*, a la que se pase el número de habitación, y vuuelva a dejar a su valor de inicialización los datos de la habitación. *(2 punts)*

```
def anularReserva(numHab):
  for piso in hotel:
    for hab in piso:
      if hab["numero"] == numHab:
        hab["ocupada"] = False
```

```
            hab["fechaIngreso"] = ""
            hab["numNoches"] = 0
            hab["precio"] = 0.0
            break
```

e)Las funciones modificarReserva(), a la que se le pase el parámetro de reserva que se desea modificar y su nuevo valor *(2 punts)*. Por ejemplo:

modificarReserva("fechaIngreso", "20221203")

-La función debe recalcular el precio de la reserva en cada caso.

-La función sólo debe poder modificar los parámetros *fechaIngreso* o *nochesReserva*.

```
def modificarReserva(clave, valorNuevo):
  if clave in ["fechaIngreso", "numNoches"]:
    for piso in hotel:
      for hab in piso:
        if hab["ocupada"]:
          hab[clave] = valorNuevo
          if hab["clase"] == "suite":
            hab["precio"] = hab["numNoches"] * 100
          else:
            hab["precio"] = hab["numNoches"] * 50
```

f)La función *listarOcupadas()*, que devuelva una lista de los números de las habitaciones no ocupadas. *(2 punts)*

```
def listarOcupadas():
  habsOcupadas = []
  for piso in hotel:
    for hab in piso:
      if hab["ocupada"]:
        habsOcupadas.append(hab["numero"])
  return habsOcupadas
```

g)La función estaLibre(), que devuelva True si la habitación está ocupada y False es caso contrario. *(2 punts)*

```
def estaLibre(numHab):
  for piso in hotel:
    for hab in piso:
      if hab["numero"] == numHab:
        return not hab["ocupada"]
```