



>> Actividad 2.3 – Pilas y Colas en Python <<

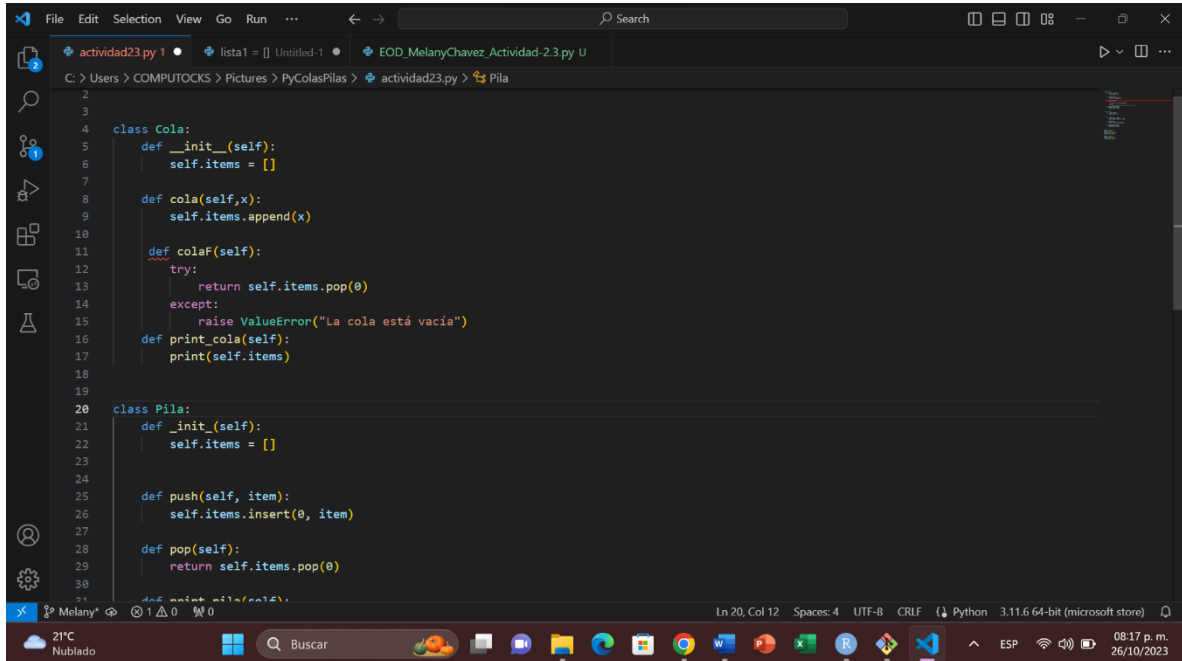
Alumno: Melany Marlen Chavez Ortiz

Maestro: Eduardo Flores.

Grupo: ITIC5

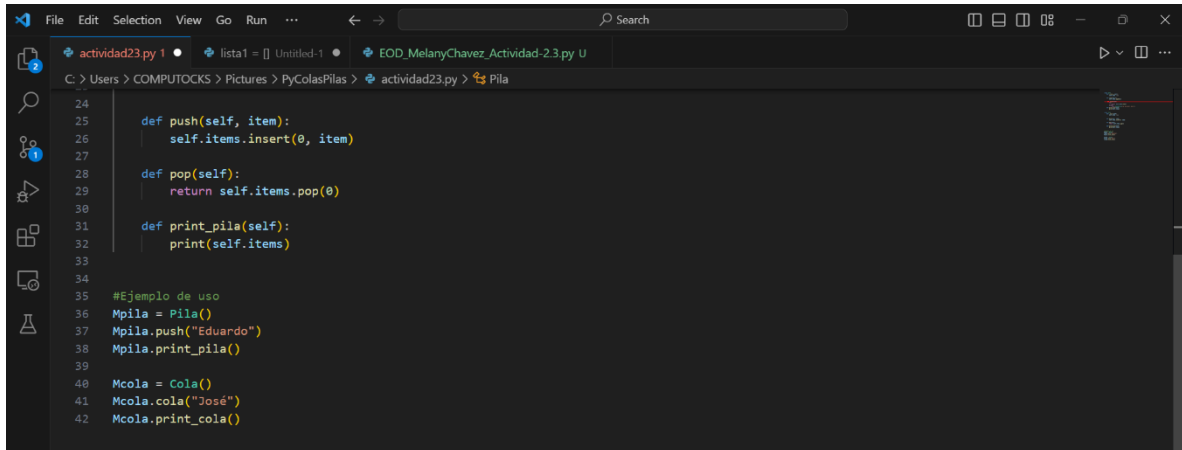
Fecha: 26/10/23

Paso 1: Pegar el programa, tiene problemas con la indentacion solo en la linea 11.



The screenshot shows a code editor with a Python file named 'actividad23.py'. The code defines two classes: 'Cola' and 'Pila'. The 'Cola' class has methods: '__init__', 'cola', 'colaF', and 'printCola'. The 'Pila' class has methods: '__init__', 'push', and 'pop'. There is a syntax error on line 11, which is 'def colaF(self):'. The error message at the bottom of the editor says 'Ln 20, Col 12 Spaces: 4 UTF-8 CRLF Python 3.11.6 64-bit (microsoft store)'. The status bar at the bottom shows the temperature as 21°C, the time as 08:17 p.m., and the date as 26/10/2023.

```
1 class Cola:
2
3
4     def __init__(self):
5         self.items = []
6
7
8     def cola(self,x):
9         self.items.append(x)
10
11     def colaF(self):
12         try:
13             return self.items.pop(0)
14         except:
15             raise ValueError("La cola está vacía")
16
17     def printCola(self):
18         print(self.items)
19
20 class Pila:
21     def __init__(self):
22         self.items = []
23
24
25     def push(self, item):
26         self.items.insert(0, item)
27
28     def pop(self):
29         return self.items.pop(0)
30
31     def print_pila(self):
32         print(self.items)
```



The screenshot shows the continuation of the Python code in the same editor. It includes the 'push' and 'pop' methods for the 'Pila' class, and a section for example usage. The example usage creates a 'Pila' object 'Mpila' and a 'Cola' object 'Mcola', and demonstrates their methods. The status bar at the bottom shows the temperature as 21°C, the time as 08:17 p.m., and the date as 26/10/2023.

```
24
25     def push(self, item):
26         self.items.insert(0, item)
27
28     def pop(self):
29         return self.items.pop(0)
30
31     def print_pila(self):
32         print(self.items)
33
34
35 #Ejemplo de uso
36 Mpila = Pila()
37 Mpila.push("Eduardo")
38 Mpila.print_pila()
39
40 Mcola = Cola()
41 Mcola.cola("José")
42 Mcola.printCola()
```

Paso 2: Editar el programa para ejecutarlo correctamente.

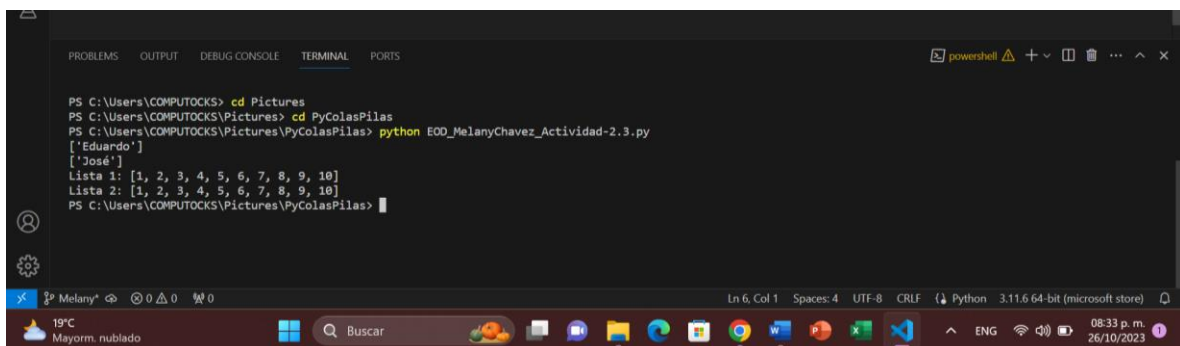
```
1  ##Programa solucionado
2
3  class Cola:
4      def __init__(self):
5          self.items = []
6
7      def cola(self,x):
8          self.items.append(x)
9
10     def colaF(self):
11         try:
12             return self.items.pop(0)
13         except:
14             raise ValueError("La cola está vacía")
15     def printCola(self):
16         print(self.items)
17
18
19     class Pila:
20         def __init__(self):
21             self.items = []
22
23
24         def push(self, item):
25             self.items.insert(0, item)
26
27         def pop(self):
28             return self.items.pop(0)
```

Paso 3: Hacer dos listas y llenar lista con números de 1 al 10.

```
#Hacer dos listas y mandar llamar las funciones llenando la lista con números de 1 al 10.
lista1 = []
lista2 = []

for numero in range(1, 11):
    lista1.append(numero)
    lista2.append(numero)
print("Lista 1:", lista1)
print("Lista 2:", lista2)
```

Programa ejecutado:



```
PS C:\Users\COMPUTOCKS> cd Pictures
PS C:\Users\COMPUTOCKS\Pictures> cd PyColasPilas
PS C:\Users\COMPUTOCKS\Pictures\PyColasPilas> python EOD_MelanyChavez_Actividad-2.3.py
['Eduardo']
['José']
Lista 1: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Lista 2: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
PS C:\Users\COMPUTOCKS\Pictures\PyColasPilas>
```

>> ¿Cual es la linea de codigo que hace que se borre el registro de la lista? def pop(self):

```
return self.items.pop(0)
```