

# PROGRAMA DESCUENTOS EN CAFETERÍA JAVERIANA

Juan Fernando Vergara y Juan Camilo Ariza, Universidad Javeriana Cali

1/12/2020

## Descuentos en cafeteria para estudiantes y profesores

Para recuperarse un poco del tiempo en cuarentena, las cafeterías de la universidad se encuentran dando descuentos a la comunidad estudiantil y dependiendo si es estudiante o profesor, tienen descuentos diferentes. Se desea saber entonces por cada compra cuánto debe pagar el usuario en caja. Para ello:

- 1) Pida por teclado la siguiente información para el cliente: cédula y rol: profesor o estudiante
- 2) Registrar el producto a comprar: código producto, cantidad de unidades y precio del producto.
- 3) Los descuentos estan dados de la siguiente forma: los estudiantes tienen un 50% de descuento mientras que los profesores tienen un 20% de descuento.

Listing 1: Codigo de python para manejo de descuentos en cafeteria

```
1 def main():
2     global productos,roles
3     inicializarRoles()
4     while(True):
5         try:
6             cedula=int(input("Ingrese su cedula "))
7             if(cedula<1):
8                 raise ValueError
9             break
10        except ValueError:
11            print("Eso no es un numero valido")
12    while(True):
13        rol=str(input("Ingrese su rol en la universidad "))
14        rol= rol.capitalize()
15        if rol in roles:
16            break
17        else:
18            print("Este rol no existe")
19    productos= dict()
20    inicializarDiccionario()
21    articulos=comprarProducto()
22    obtenerPrecio(articulos, rol, cedula)
23
```

```

24 def inicializarDiccionario():
25     global productos
26     productos["Papas"]=[1, 2500]
27     productos["Gaseosa"]=[2, 2000]
28     productos["Perro"]=[3, 5000]
29     productos["Pizza"]=[4, 6000]
30     productos["Galletas"]=[5, 1200]
31     productos["Chicle"]=[6, 200]
32     productos["Paleta"]=[7, 1500]
33 def inicializarRoles():
34     global roles
35     roles=["Profesor", "Estudiante"]
36
37 def comprarProducto():
38     global productos, roles
39     for i in productos:
40         print(i,".....",productos[i])
41
42     while(True):
43         try:
44             cantidadArticulos=int(input("Cuantos articulos desea comprar? ↵
45                                     "))
46             if(cantidadArticulos<1):
47                 raise ValueError
48             break
49         except ValueError:
50             print("Eso no es un numero valido")
51     articulosAComprar=[]
52     for i in range(cantidadArticulos):
53         articulo=str(input("Ingrese el nombre del articulo que desea ↵
54                             comprar: "))
55         articulo= articulo.capitalize()
56         while(True):
57             if articulo in productos:
58                 try:
59                     cantidadDeArticulo=int(input("Ingrese la cantidad de "↵
60                                                     + articulo + " desea comprar "))
61                     if(cantidadDeArticulo<1):
62                         raise ValueError
63                     articulosAComprar.append([cantidadDeArticulo, articulo↵
64                                             ])
65                     break
66                 except ValueError:
67                     print("Eso no es un numero valido")
68             else:
69                 print("Este articulo no existe")
70                 articulo=str(input("Ingrese el nombre del articulo que ↵

```

```

        desea comprar: "))
67     return articulosAComprar
68 def obtenerPrecio(articulos, rol, cedula):
69     global productos
70     if(rol=="Estudiante"):
71         descuento=0.5
72     elif(rol=="Profesor"):
73         descuento=0.8
74     total=0
75     for i in articulos:
76         codigo=productos[i[1]][0]
77         precioArticulo=productos[i[1]][1] * descuento
78         cantidadDeArticulo=i[0]
79         precio= precioArticulo*cantidadDeArticulo
80         print("El", rol, "con cedula", cedula, ", debe pagar", ←
            precioArticulo, "por el producto", codigo, "y son", ←
            cantidadDeArticulo, "unidades, para un total de", precio)
81     total+=precio
82     print("El precio total es:", total)
83 main()

```

---

Revisando Listing 1... Tenemos el código completo del programa creado para solución al problema principal. Se utilizó varias funciones para poder trabajar los requisitos que se pedían del programa y una función main que se encarga de ejecutar el programa de una manera correcta utilizando las funciones definidas previamente. También se trabajó el manejo de errores en el código para los errores que puedan surgir durante la ejecución.

## Funciones

Listing 2: inicializarDiccionario()

```

1 def inicializarDiccionario():
2     global productos
3     productos["Papas"]=[1, 2500]
4     productos["Gaseosa"]=[2, 2000]
5     productos["Perro"]=[3, 5000]
6     productos["Pizza"]=[4, 6000]
7     productos["Galletas"]=[5, 1200]
8     productos["Chicle"]=[6, 200]
9     productos["Paleta"]=[7, 1500]

```

---

Esta función define el arreglo que va a contener todos los artículos que se van a poder comprar en la cafetería

---

Listing 3: inicializarRoles()

---

```
1 def inicializarRoles():
2     global roles
3     roles=["Profesor", "Estudiante"]
```

---

Esta función define el arreglo que va a contener todos los roles de la universidad.

---

Listing 4: comprarProducto()

---

```
1 def comprarProducto():
2     global productos, roles
3     for i in productos:
4         print(i,".....",productos[i])
5
6     while(True):
7         try:
8             cantidadArticulos=int(input("Cuantos articulos desea comprar? ↵
9                                     "))
10            if(cantidadArticulos<1):
11                raise ValueError
12            break
13        except ValueError:
14            print("Eso no es un numero valido")
15    articulosAComprar=[]
16    for i in range(cantidadArticulos):
17        articulo=str(input("Ingresa el nombre del articulo que desea ↵
18                        comprar: "))
19        articulo= articulo.capitalize()
20        while(True):
21            if articulo in productos:
22                try:
23                    cantidadDeArticulo=int(input("Ingresa la cantidad de "↵
24                                                + articulo + " desea comprar "))
25                    if(cantidadDeArticulo<1):
26                        raise ValueError
27                    articulosAComprar.append([cantidadDeArticulo, articulo↵
28                                            ])
29                    break
30                except ValueError:
31                    print("Eso no es un numero valido")
32            else:
33                print("Este articulo no existe")
34                articulo=str(input("Ingresa el nombre del articulo que ↵
35                                desea comprar: "))
36    return articulosAComprar
```

---

Esta función define cual es el procedimiento y análisis que debe hacer el programa para realizar la compra de un producto, verificando los productos que existen, la cantidad de productos que se quieren comprar, el nombre del producto que se quiere comprar y el manejo de errores en caso de una entrada errónea de parte del usuario.

Listing 5: obtenerPrecio()

---

```
1 def obtenerPrecio(articulos, rol, cedula):
2     global productos
3     if(rol=="Estudiante"):
4         descuento=0.5
5     elif(rol=="Profesor"):
6         descuento=0.8
7     total=0
8     for i in articulos:
9         codigo=productos[i[1]][0]
10        precioArticulo=productos[i[1]][1] * descuento
11        cantidadDeArticulo=i[0]
12        precio= precioArticulo*cantidadDeArticulo
13        print("El", rol, "con cedula", cedula,", debe pagar", ←
            precioArticulo, "por el producto", codigo, "y son", ←
            cantidadDeArticulo, "unidades, para un total de", precio)
14        total+=precio
15    print("El precio total es:", total)
```

---

Esta función permite obtener el precio total de la compra realizada por el usuario. La función recibe como entrada los articulos, el rol del usuario y su cédula para poder ejecutarse y hacer su respectivo cálculo. Al final retorna un texto indicando el rol del usuario, su cédula, la cantidad de artículos que va a comprar y el precio de cada artículo, y el total de toda la compra.

Listing 6: main()

---

```
1
2 def main():
3     global productos,roles
4     inicializarRoles()
5     while(True):
6         try:
7             cedula=int(input("Ingrese su cedula "))
8             if(cedula<1):
9                 raise ValueError
10            break
11        except ValueError:
12            print("Eso no es un numero valido")
```

---

```

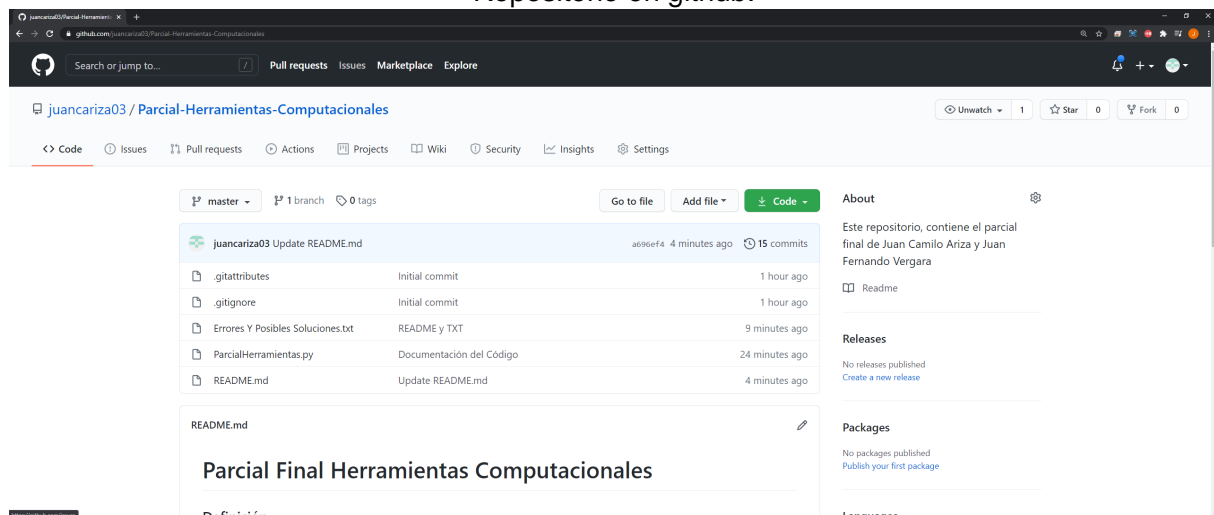
13 while(True):
14     rol=str(input("Ingrese su rol en la universidad "))
15     rol= rol.capitalize()
16     if rol in roles:
17         break
18     else:
19         print("Este rol no existe")
20 productos= dict()
21 inicializarDiccionario()
22 articulos=comprarProducto()
23 obtenerPrecio(articulos, rol, cedula)

```

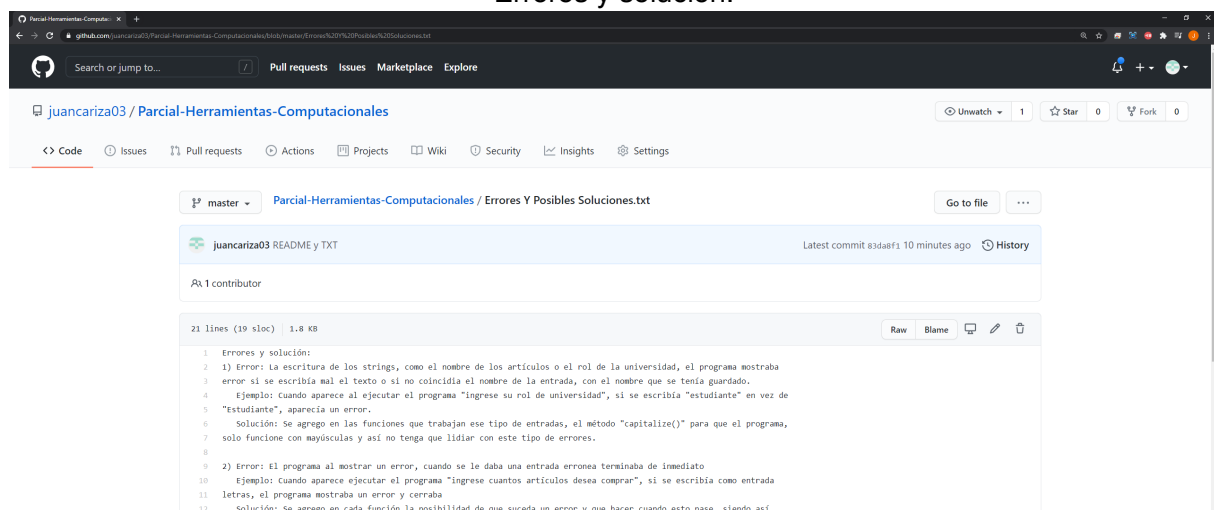
Esta es la función base de nuestro programa que se encargar de hacer llamado a todas las demás funciones para hacer que el programa funcione de manera correcta.

## Imágenes del repositorio

Repositorio en github:



Errores y solución:



## README:

The screenshot shows the GitHub interface for the repository 'juancariza03 / Parcial-Herramientas-Computacionales'. The README file is displayed, showing the title 'Parcial Final Herramientas Computacionales' and a section titled 'Definición'. The text describes an algorithm that allows a university user to get a discount in the cafeteria by purchasing items, depending on their role. It mentions that functions were implemented containing methods, conditions, variables, arrays, and text impressions. It also notes that recursion was used.

Parcial Final Herramientas Computacionales

**Definición**

El algoritmo permite al usuario de la universidad adquirir un descuento en cafetería al comprar artículos dependiendo de su rol en esta.

Para la solución del problema trabajado en el algoritmo, se implementaron funciones que contenían métodos, condiciones, variables, arreglos e impresiones de texto. Aparte también se dio uso a la recursividad.

## Documentación:

The screenshot shows the GitHub interface for the repository 'juancariza03 / Parcial-Herramientas-Computacionales'. The documentation file 'ParcialHerramientas.py' is displayed. The code is a Python script with a main function that contains comments describing the input and output of the program. The input is a client's ID, which must be greater than 0 and can be up to 8 digits. The role of the client is also specified, with options for 'Profesor' and 'Estudiante'. The output is the price of the products being purchased.

```
1 def main():
2     """
3     Entrada:
4     -cedula del cliente: Esta tiene que ser un número mayor que 0, recuerde que la cedula puede ser 000000001
5     -Rol del cliente: Este es el rol que tiene asignado el usuario, debe pertenecer al grupo de los roles definidos previamente ["Profesor", "Estudiante"]
6     -Roles del sistema (Global)
7     -Productos del sistema (Global)
8     Salida:
9     -Precio de los productos que está adquiriendo
10    """
11    global productos, roles
12    inicializarRoles()
```