Leonardo de los Rios

Leonardo de los Rios

Leonardo de los Rios
DNI: 44.018.085
LSI - 21198

Parcial II POO

Práctica

Aprobado

ClaseAparatoTec.py

```
from abc import ABC
import abc
class AparatoTec(ABC):
    --marca=''
    --modelo=''
    --color=''
    --paisFab=''
    --precioBase=0.0
    --importeVenta=0.0
def __init__(self, marca='', modelo='', color='', paisFab='', precioBase=0.0):
    assert isinstance(marca, str)
    assert isinstance(modelo, str)
    assert isinstance(color, str)
    assert isinstance(paisFab, str)
    assert isinstance(precioBase, float)
    assert precioBase > 0
    self.__marca = marca
    self.__modelo = modelo
    self.__color = color
    self.__paisFab = paisFab
    self.__precioBase = precioBase
    self.__importeVenta = 0.0
@abc.abstractmethod
def calculaImporte():
    pass
def getPrecioBase(self):
    return self.__precioBase
def setImporte(self, importe):
    assert isinstance(importe, float) and importe > 0
    self.__importeVenta = importe
```

NOTA

```
def calculaImporte(self):
    self.getPrecioBase()
```



Leonardo de los Ríos

```
def getMarca(self):
    return self.__marca
def getImporte(self):
    return self.__importeLlanta
def getPais(self):
    return self.__paisFab
```

ClaseTelevisor.py

```
from claseAparatoTec import AparatoTec

class Televisor(AparatoTec):
    --tipoPantalla = ''
    --pulgadas = 0.0
    --tipoDef = ''
    --conexInternet = False


def __init__(self, marca='', modelo,='', color='', paisFab='', precioBase=0.0,
             tipoPantalla='', pulgadas=0.0, tipoDef='', conexInternet=False):
    assert isinstance(tipoPantalla, str)
    assert isinstance(pulgadas, float)
    assert pulgadas > 0
    assert isinstance(tipoDef, str)
    assert isinstance(conexInternet, bool)
    super().__init__(marca, modelo, color, paisFab, precioBase)
    self.__tipoPantalla = tipoPantalla
    self.__pulgadas = pulgadas
    self.__tipoDef = tipoDef
    self.__conexInternet = conexInternet

def calculaImporte(self):
    precioB
    importe = self.getPrecioBase()
    if self.__tipoDef == 'SD':
        importe += importe * 0,01
    elif self.__tipoDef == 'HD':
        importe += importe * 0,02
    elif self.__tipoDef == 'FULL HD':
        importe += importe * 0,03
```

NOTA

Leonardo de los Rios



if self.__canexInternet == True:
importe += self.getPrecioBase() * 0,8
return importe

ClaseHeladera.py
from claseAparatoTec import AparatoTec

class Heladera(AparatoTec):
__capacidad = 0.0
__freezer = False

def __init__(self, Marca="", modelo="", color="", paisFab="", precioBase=0.0,
capacidad=0.0, freezer=False):
assert isinstance(capacidad, float)
assert capacidad > 0
assert isinstance(freezer, bool)
super().__init__(Marca, modelo, color, PaisFab, PrecioBase)
self.__capacidad = capacidad
self.__Freezer = freezer

Leonardo de los Ríos

PÁG (5)

FECHA 3/06/21

```
def calculoImporte(self):
    importe = self.getPrecioBase()
    if self.__freezer == False:
        importe += importe * 0,01
    else:
        importe += importe * 0,05
    return importe
```

NOTA

else:



ClaseLavarropas.py

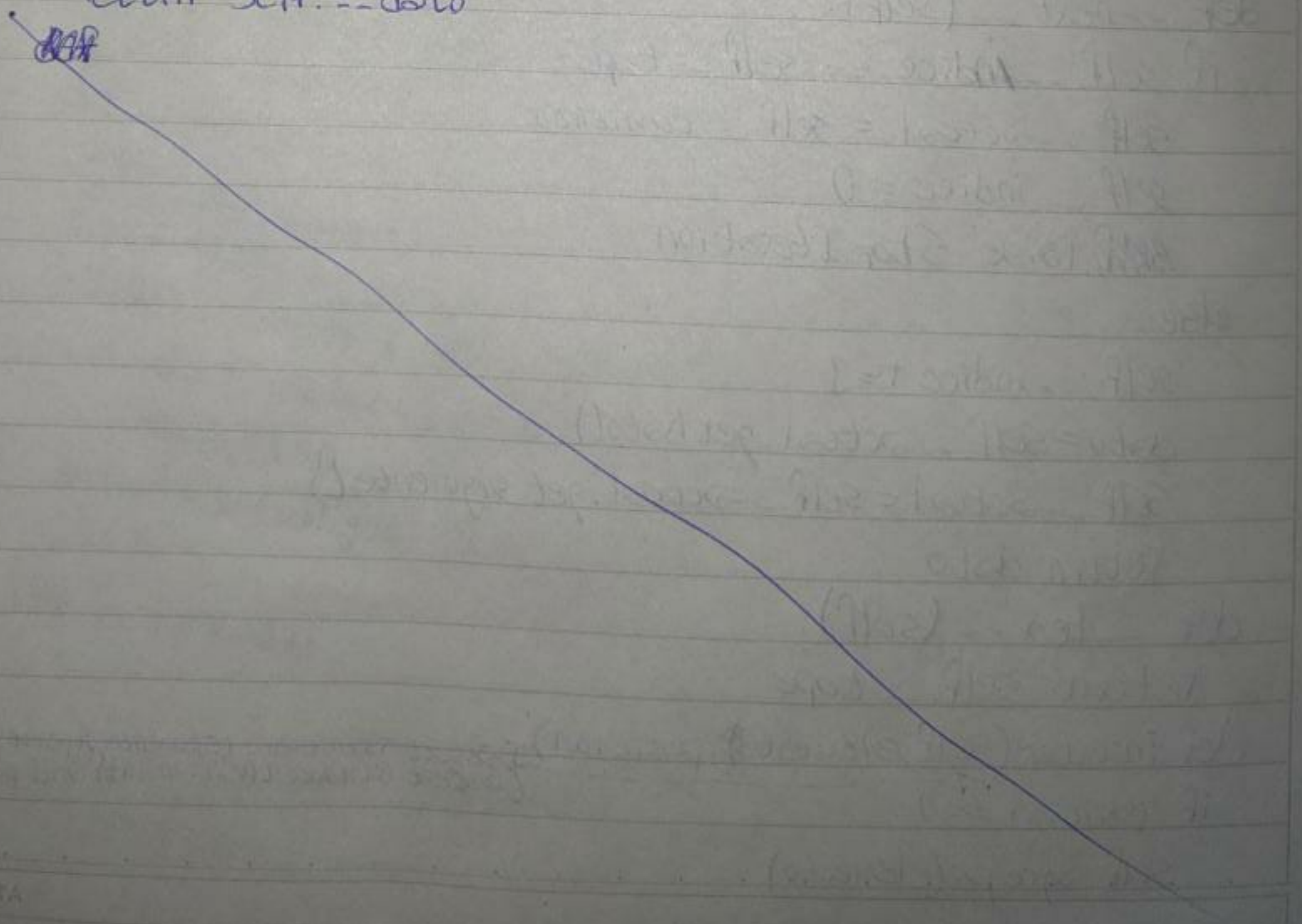
```
From claseAparatoTec import AparatoTec

Class Lavarropas(AparatoTec):
    --capacidad = 0
    --centrifugado = 0
    --cantidadProg = 0
    --tipoCarga = ''
def __init__(self, marca='', modelo='', color='', paisFab='', precioBase=0.0,
             capacidad=0, centrifugado=0, cantidadProg=0, tipoCarga=0):
    assert isinstance(capacidad, int) and capacidad > 0
    assert isinstance(centrifugado, int) and centrifugado > 0
    assert isinstance(cantidadProg, int) and cantidadProg > 0
    assert isinstance(tipoCarga, str) and (tipoCarga == 'Superior' or tipoCarga ==
                                           'Inferior')
    super().__init__(marca, modelo, color, paisFab, precioBase)
    self.__capacidad = capacidad
    self.__centrifugado = centrifugado
    self.__cantidadProg = cantidadProg
    self.__tipoCarga = tipoCarga
def calculaImporte(self):
    importe = self.getPrecioBase()
    if self.__capacidad <= 5:
        importe += importe * 0,05
    else:
        importe += importe * 0,03
def getCarga(self):
    return self.__tipoCarga
```

ClaseNodo.py

```
from claseAparatoTec import AparatoTec

class Nodo:
    __dato = None
    __siguiente = None
    def __init__(self, dato):
        assert isinstance(dato, AparatoTec)
        self.__dato = dato
        self.__siguiente = None
    def setSiguiente(self, siguiente):
        self.__siguiente = siguiente
    def getSiguiente(self):
        return self.__siguiente
    def getDato(self):
        return self.__dato
```



NOTA

else:
aux = ...
Leonardo de los Bios

CloseLista.py

```
From closeTelevisor , import Televisor
From closeHeladera import Heladera
From closeLavarropa import Lavarropa
From closeNodo import Nodo

Class Lista:
  -- comienzo=None
  -- actual=None
  -- indice=0
  -- tope=0
  def__ init__(self):
    self.__comienzo=None
    Self.__actual=None
    self.__indice=0
    self.__tope=0
  def __iter__(self)
    return self
  def __next__(self):
    if self.__indice == self.__tope:
      self.__actual = self.__comienzo
      self.__indice=0
      raise StopIteration
  else:
    self.__indice+=1
    dato=self.__actual.getDato()
    self.__actual=self.__actual.getSiguiente()
    return dato
  def __len__(self):
    return self.__tope
  def insertar(self,elemento,posicion):
    assert isinstance(elemento,Aparat.Tec)
    assert isintance(posicion,int) and posicion>=0
    if posicion ==0:
      self.agregar(elemento)
```

Leonardo de los Ríos

```
else:
    aux = self.__comienzo
    i=0
    elemento=Nodo(elemento)
    while ~~posicion~~ i<posicion and aux != None:
        anterior = aux
        aux =aux.getSiguiente()
        i+=1
    if i==posicion:
        raise IndexError
    else:
        elemento.setSiguiente(aux)
        anterior.setSiguiente(elemento)
        self.__tope +=1
def agregar(self, elemento):
    assert isinstance(elemento, ApDratoTec)
    nodo = Nodo(elemento)
    nodo.setSiguiente(self.__comienzo)
    self.__comienzo = nodo
    self.__actual = nodo
    self.__tope+=1
def opc3(self):
    ~~i~~ i=0
    actual=self.__comienzo
    Cantidad=0
    while i<len(self) :
        if actual.getDato().getMarca() == 'Philips':
            Cantidad+=1
        actual=actual.getSiguiente()
        i+=1
    return Cantidad
```

NOTA

```
def opc4(self):
    Marcas=[]
    i=0
    actual=self.__comienzo
    while i<(self):
        if isinstance(actual,CoustTropa):
            if actual.getDato().getCarga()=='superior' and actual.getDato().getMarca() not in Marcas:
                Marcas.append(actual.getDato().getMarca())
        actual=actual.getSiguiente()
        i+=1
    for Marca in Marcas:
        Print(Marca)
def opc5(self):
    for dato in self:
        print(dato.getMarca()+'\n'+dato.getPais()+'\n')
        dato.calculaImporte()
        Print(dato.getImporte())
```

Pag 18

FECHA

NOTA

Leonardo de los Bios

Pag (52)

ClasMenuOpciones.py

```
from claseLista import Lista
from claseHeladera import Heladera
from claseTelevisor import Televisor
from claseLavarropa import Lavarropa

class Menu:
    __opc=0
    def __init__(self):
        self.__opc=0
    def menu():
        lista=Lista()
        continuar=True
        while continuar:
            Print(F"""    Menú opciones:
1) Insertar.
2) Agregar.
3) Mostrar Marca Philips. 4) Lavarropa carga superior. 5) Mostrar.
0) Salir
""")
            self.__opc=int(input('Ingrese una opción: '))
            if sel.__opc==1:
                aparatos=['Televisor','Heladera','Lavarropa'] ⟹ va en solicitaDatos()
                {posicion=int(input('Ingrese pos: '))
                self.solicitaDatos(lista,aparato)
                aparato=input('Ingrese aparato: ')
                if aparato==aparatos[0]:
                    tipoPant=input('Tipo Pant: ')
                    pulgadas=float(input('Pulgadas: '))
                    tipoDef=input('Tipo def: ')
                    conexInternet=bool(input('Conex Internet: '))
                    unTelevisor=Televisor(marca,modelo,color,paisFab,precioBase
                    tipoPan,pulgadas,tipoDef,conexInternet)
                if self.__opc==1:
                    lista.insertar(unTelevisor,posicion)
                else:
                    lista.agregar(unTelevisor)
```

si el usuario eligió la opc de insertar, se solicita la posición

Todo esto va en def solicitaDatos

Leonardo J.

Pág 13

FECHA

```
elif aparato == aparatos[1]:
    capacidad = float(input('Capacidad: '))
    freezer = bool(input('Freezer: '))
    unaHeladera = Heladera(marca, modelo, color, paisFab, precioBase,
    capacidad, freezer)
    if self.__opc == 1:
        lista.insertar(unaHeladera, posicion)
    else: lista.agregar(unaHeladera)
elif aparato == aparatos[2]:
    capacidad = int(input('Capacidad: '))
    centrifugado = int(input('Centrifugado: '))
    cantidadProg = int(input('Cantidad Prog: '))
    tipoCarga = input('Tipo carga: ')
    unLavarropa = Lavarropa(marca, modelo, color, paisFab, precioBase,
    capacidad, centrifugado, cantidadProg, tipoCarga)
    if self.__opc == 1:
        lista.insertar(unLavarropa, posicion)
    else:
        lista.agregar(unLavarropa)
elif self.__opc == 2:
    [illegible]
    self.solicitaDatos(lista, aparato)
    aparato = input('Ingrese aparato: ')
elif self.__opc == 3:
    cantidad = lista.opc3()
    print(cantidad)
elif self.__opc == 4:
    lista.opc4()
elif self.__opc == 5:
    lista.opc5()
elif self.__opc == 0:
    continuar = False
else:
    print('Opción no válida.')
```

def solicitaDatos()
se corrigió en pág 85 y 86

NOTA

Pág (15)

Leonardo de los Ríos

def solicitaDatos(self, lista, aparato):
Ya se escribió el contenido en las págs. (12 y 13) Se corrigió en pág (16)

Pág (14)

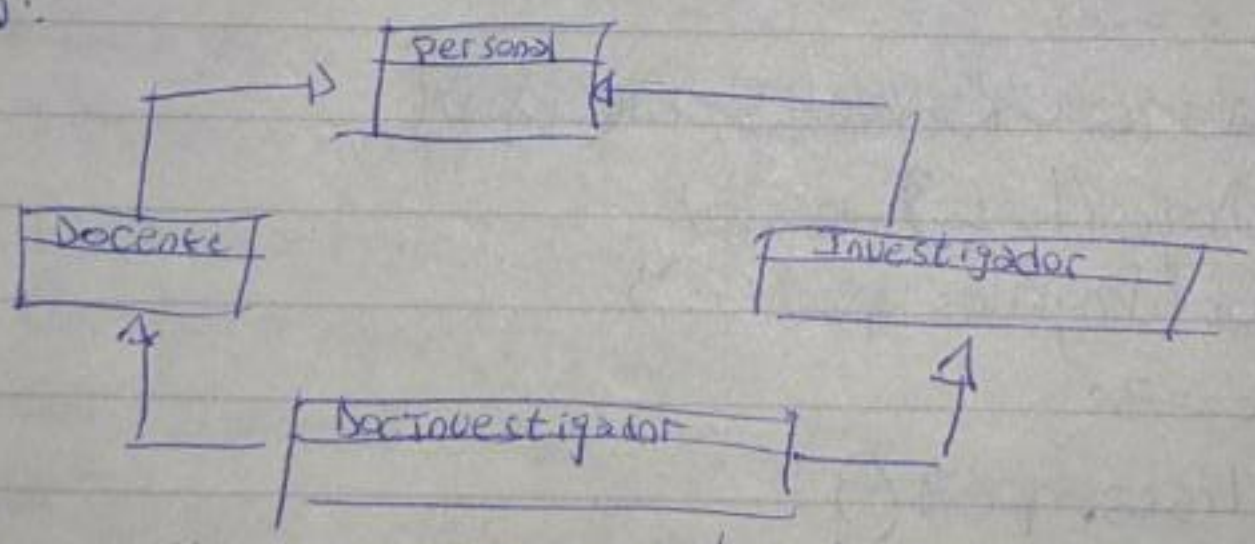
Main.py
```
from claseMenuOpciones import menuOpciones
if __name__ == '__main__':
    unMenu = MenuOpciones()
    unMenu.Menu
```

---

Teoría.

1) La diferencia es que las clases que implementen los métodos en una interfaz no deben estar relacionadas por la herencia. En cambio, las clases hijas que se derivan de una clase Padres heredan los métodos de la clase Padre.

2) La regla del diamante funciona de derecha a izquierda.
Ejemplo:




Personal: recibe sus atributos + los de docente + los de investigador.
Docente: recibe los de personal + los de investigador + sus atributos.
Investigador: recibe los de personal + los suyos + los de docente.
DocenteInves: recibe los de personal + los de investigador + los de docente.
En la clase DocenteInv. se puede apreciar su funcionamiento.

3) Los bloques mencionados se utilizan para no frenar la ejecución del programa y evitar de esa forma tener que volver a ejecutarlo.

Corrección en ClaseMenuOpciones.py :

```
if self.__opc == 1:
    aparatos = ['Televisor', 'Heladera', 'Lavarropa']
    aparato = input('Aparato:')
    if aparato in aparatos:
        self.solicitaDatos(lista, aparato)
elif self.__opc == 2:
    aparatos = ['Televisor', 'Heladera', 'Lavarropa']
    aparato = input('Aparato: ')
    if aparato in aparatos:
        self.solicitaDatos(lista, aparato)
```

```
def solicitaDatos(self, lista, aparato):
    marca = input('Marca: ')
    modelo = input('Modelo: ')
    color = input('Color: ')
    precioBase = float(input('PrecioBase: '))
    if aparato == 'Televisor':
        tipoPan = input('Tipo Pantalla: ')
        pulgadas = float(input('Pulgadas: '))
        tipoDef = input('Tipo def:')
        conex Internet = bool(input('Conex Internet: '))
        unTelevisor = Televisor(marca, modelo, color, precioBase, tipoPan, pulgadas,
                                tipoDef, conexInternet)
        if self.__opc == 1:
            posicion = int(input('posicion'))
            lista.insertar(unTelevisor, posicion)
        elif self.__opc == 2:
            lista.agregar(unTelevisor)
    elif aparato == 'Heladera':
        capacidad = float(input('Capacidad: '))
```

NOTA

```
freezer = bool(input('Freezer: '))
unaHeladera = Heladera(marca, modelo, color, paisFab, precioBase,
                       capacidad, freezer)
if self.__opc == 1:
    posicion = int(input('posicion: '))
    lista.insertar(unaHeladera, posicion)
elif self.__opc == 2:
    lista.agregar(unaHeladera)
else:
    capacidad = int(input('Capacidad: '))
    centrifugado = int(input('Centrifugado: '))
    cantidadProg = int(input('Cantidad Prog: '))
    tipoCarga = input('Tipo carga: ')
    unLavarropa = Lavarropa(marca, modelo, color, paisFab, precioBase,
                            capacidad, centrifugado, cantidadProg, tipoCarga)
    if self.__opc == 1:
        posicion = int(input('Posicion: '))
        lista.insertar(unLavarropa, posicion)
    elif self.__opc == 2:
        lista.agregar(unLavarropa)
```

NOTA: