

```
In [14]: 1 class Person :
2         def __init__ ( self, n, sN, s):
3             self.name = n
4             self.surname = sN
5             self.school = s
6         def print_name ( self ):
7             print (self.name,self.surname)
8         def print_school ( self ):
9             print (self.school)
10
11 p2= Person ('Laura','Exposito', 'UB')
12 p2.print_name()
```

Laura Exposito

```
In [33]: 1 class Complejo:
2         def __init__(self, realPart,imagPart):
3             self.r = realPart
4             self.i = imagPart
5         def re(self):
6             print ("parte real: ", self.r)
7         def im(self):
8             print ("parte imaginaria: ", self.i)
9         def __str__(self):
10            return (f"{self.r}+{self.i}i")
11
12
13 n = Complejo (2,3)
14 n.re()
15 n.im()
16 n2 = Complejo (3,7)
17 print(n2)
18
```

parte real: 2
parte imaginaria: 3
3+7i

In [50]:

```

1 class Pelicula:
2
3     # Constructor de clase
4     def __init__(self, titulo, duracion, lanzamiento):
5         self.titulo = titulo
6         self.duracion = duracion
7         self.lanzamiento = lanzamiento
8         print('Se ha creado la película:', self.titulo, "Duracion:", se
9
10    def __str__(self):
11        cad= f'{self.titulo} ({self.lanzamiento}) ({self.duracion} min)
12        return(cad)
13
14
15 class Catalogo:
16
17     peliculas = [] # Esta lista contendrá objetos de la clase Pelicula
18
19     def __init__(self, peliculas=[]):
20         print("\n CATALOGO-----")
21         self.peliculas = peliculas
22
23     def agregar(self, p): # p será un objeto Pelicula
24         self.peliculas.append(p)
25
26     def mostrar(self):
27         for p in self.peliculas:
28             print(p) # Print toma por defecto str(p)
29
30
31 p = Pelicula("El Padrino", 175, 1972)
32 c = Catalogo([p]) # Añado una lista con una película desde el principi
33 c.mostrar()
34 c.agregar(Pelicula("El Padrino: Parte 2", 202, 1974)) # Añadimos otra
35 c.mostrar()
36 c.agregar(Pelicula("El Padrino: Parte 3", 150, 1991))

```

Se ha creado la película: El Padrino Duracion: 175 min

CATALOGO-----

El Padrino (1972) (175 min)

Se ha creado la película: El Padrino: Parte 2 Duracion: 202 min

El Padrino (1972) (175 min)

El Padrino: Parte 2 (1974) (202 min)

Se ha creado la película: El Padrino: Parte 3 Duracion: 150 min

El Padrino (1972) (175 min)

El Padrino: Parte 2 (1974) (202 min)

El Padrino: Parte 3 (1991) (150 min)

In [90]:

```

1 class Car:
2     def __init__(self, marca, modelo, color, matricula):
3         self.marca = marca
4         self.modelo = modelo
5         self.color = color
6         self.matricula = matricula
7     def __str__(self):
8         cad= f'{self.marca} ({self.modelo})({self.color}) ({self.matricula})'
9         return(cad)
10
11 class Cars:
12     lista = []
13
14     def __init__(self, lista=[]):
15         self.lista = lista
16
17
18     def llista_tots(self):
19         for c in self.lista:
20             print(c)
21     def cerca_matricula(self, matricula):
22         encontrado = False
23         for coche in self.lista:
24             if matricula == coche.matricula:
25                 print("Encontrado:", coche)
26                 encontrado = True
27                 break
28         if not encontrado:
29             print ("No encontrado")
30
31
32 c1 = Car ("TOYOTA", "YARIS", "ROJO", "6789-CYR")
33 c2 = Car ("TOYOTA", "YARIS", "VERDE", "333-CYR")
34 c = Cars([c1,c2])
35 c.llista_tots()
36

```

TOYOTA (YARIS)(ROJO) (6789-CYR)

TOYOTA (YARIS)(VERDE) (333-CYR)

Encontrado: TOYOTA (YARIS)(VERDE) (333-CYR)

In [93]:

```
1 class Perro:
2     def sonido(self):
3         print('Guauuuuu!!!')
4 class Gato:
5     def sonido(self):
6         print('Miaaauuuu!!!')
7 class Vaca:
8     def sonido(self):
9         print('Múuuuuuuu!!!')
10
11 def a_cantar(animales):
12     for animal in animales:
13         animal.sonido()
14
15 perro1 = Perro()
16 perro2 = Perro()
17
18 gato1 = Gato()
19 gato2 = Gato()
20
21 vaca = Vaca()
22
23 granja = [perro1, gato1, vaca, gato2, perro2]
24 a_cantar(granja)
```

Guauuuuu!!!

Miaaauuuu!!!

Múuuuuuuu!!!

Miaaauuuu!!!

Guauuuuu!!!

In [110]:

```
1 class Producto:
2     def __init__(self, referencia, nombre, pvp, descripcion):
3         self.referencia = referencia
4         self.nombre = nombre
5         self.pvp = pvp
6         self.descripcion = descripcion
7
8     def __str__(self):
9         return f"REFERENCIA\t {self.referencia}\n" \
10              f"NOMBRE\t\t {self.nombre}\n" \
11              f"PVP\t\t {self.pvp}\n" \
12              f"DESCRIPCIÓN\t {self.descripcion}\n"
13
14 class Alimento(Producto):
15     productor = ""
16     distribuidor = ""
17
18     def __str__(self):
19         return f"REFERENCIA\t {self.referencia}\n" \
20              f"NOMBRE\t\t {self.nombre}\n" \
21              f"PVP\t\t {self.pvp}\n" \
22              f"DESCRIPCIÓN\t {self.descripcion}\n" \
23              f"PRODUCTOR\t\t {self.productor}\n" \
24              f"DISTRIBUIDOR\t\t {self.distribuidor}\n"
25
26
27 class Libro(Producto):
28     isbn = ""
29     autor = ""
30
31     def __str__(self):
32         return f"REFERENCIA\t {self.referencia}\n" \
33              f"NOMBRE\t\t {self.nombre}\n" \
34              f"PVP\t\t {self.pvp}\n" \
35              f"DESCRIPCIÓN\t {self.descripcion}\n" \
36              f"ISBN\t\t {self.isbn}\n" \
37              f"AUTOR\t\t {self.autor}\n"
38
39 class Textil(Producto):
40     color = "Negro"
41     material = "Cuero"
42
43     def __str__(self):
44         return f"REFERENCIA\t {self.referencia}\n" \
45              f"NOMBRE\t\t {self.nombre}\n" \
46              f"PVP\t\t {self.pvp}\n" \
47              f"DESCRIPCIÓN\t {self.descripcion}\n" \
48              f"COLOR\t\t {self.color}\n" \
49              f"MATERIAL\t\t {self.material}\n"
50
51 class Brico (Producto):
52     medida = "1 metro"
53     peso = "0.5 kh"
54
55     def __str__(self):
56         return f"REFERENCIA\t {self.referencia}\n" \
57              f"NOMBRE\t\t {self.nombre}\n" \
58              f"PVP\t\t {self.pvp}\n" \
59              f"DESCRIPCIÓN\t {self.descripcion}\n" \
```

```
60         f"MEDIDA\t\t {self.medida}\n" \
61         f"PESO\t\t {self.peso}\n"
62 p=Brico("J72w","Tornillo y tuerca 4mm","2.5","blister de 20 unidades")
REFERENCIA ` J72w
NOMBRE      Tornillo y tuerca 4mm
PVP         2.5
DESCRIPCIÓN blister de 20 unidades
MEDIDA      1 metro
PEÑO        0.5 kh
```