

```
1 import random  
2  
3 def buscarElemento(lista, elemento):  
4     for i in range(0,len(lista)):  
5         if(lista[i] == elemento):  
6             return i  
7  
8 def imprimirLista(lista,nombre):  
9     for i in range(0,len(lista)):  
10        print nombre + "[" + str(i) + ":" + lista[i] + "]"  
11  
12 def leerLista():  
13     lista=[]  
14  
15     i=0  
16     while True:  
17         linea = input()  
18         if linea == "":  
19             return lista  
20         else:  
21             lista.append(linea)  
22
```



Python

Anyela Jhohana Herrera Lobo - 1152256
Leydi Alejandra Durán Rozo- 1152258
Michelle Alexandra Rojas Durán- 1152261

HISTORIA

Comenzó el desarrollo de un nuevo lenguaje de programación casi por hobby, tomando su nombre de el grupo humorístico británico Monty Python



1989

FINALES DE LOS 80

Guido van Rossum ideó el lenguaje Python



1994 Y 2020

1991



Versiones 1.0 y 2.0



2001

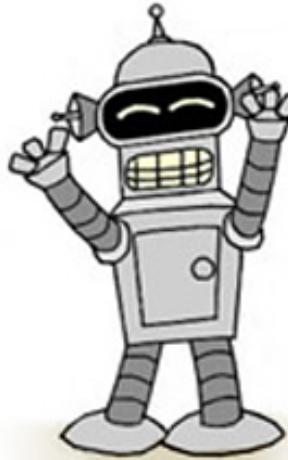
Se funda la Python Software Foundation (PSF), organización cuyo objetivo es el de fomentar el desarrollo de la comunidad Python, así como el de ser responsable de los procesos que se llevan a cabo dentro de dicha comunidad.



Version 3.0



python

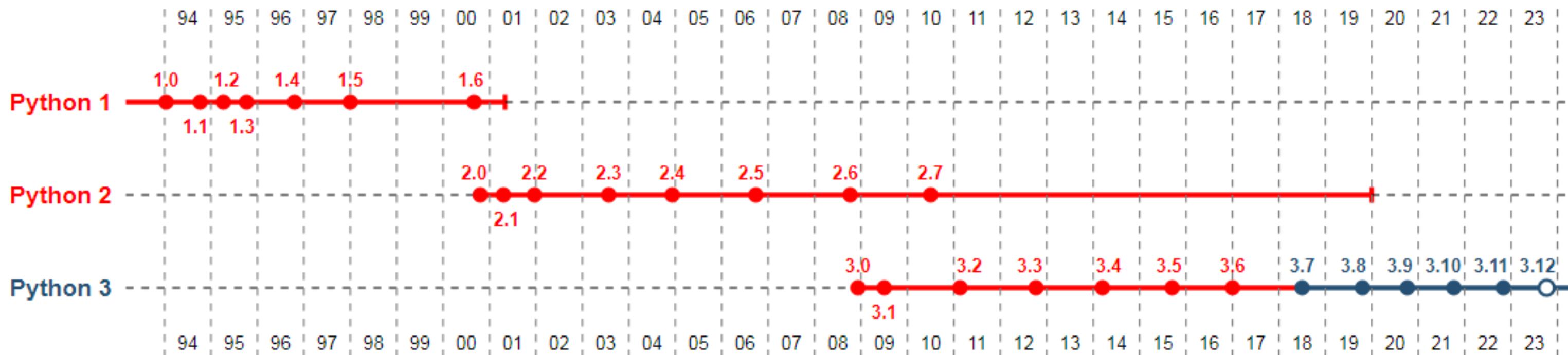


Hello world

Principales versiones

La primera versión ya incluía clases con herencias, manejo de excepciones, funciones y una de las características principales de Python: funcionamiento modular.

La tercera gran versión venía solucionando los principales fallos en el diseño de este lenguaje de programación.



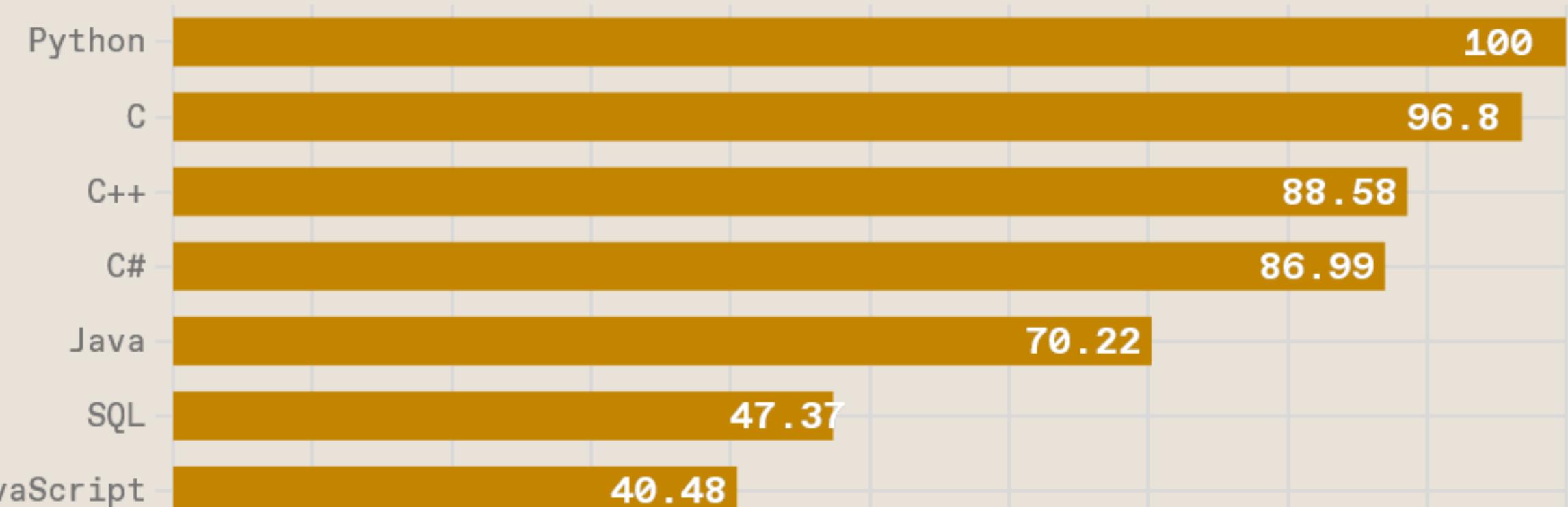
En la segunda versión en la que se incluyó la generación de listas

-Detecta el desbordamiento de números enteros cuando la nueva longitud asignada está cerca del tamaño máximo.
-En Linux, el multiprocessing módulo vuelve a usar sockets de dominio Unix

Ranking

Top Programming Languages 2022

Click a button to see a differently-weighted ranking



Utilidades del lenguaje

Juegos y gráficos 3D

Inteligencia artificial.

Tareas de ciencia de datos y machine learning.

Desarrollo web

Conceptos de Clases y Objetos

clases

```
2  
3 class Aerolinea:  
4  
1 class Persona:
```

Plantillas o modelos

objetos

```
pedro= Persona(2456543, "Pedro")  
maria= Persona(5890093, "María")  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15
```

entidades de la vida real

métodos

```
8  
9     def buscarUsuario(self, persona):  
10        resultado= None  
11        for p in self._usuarios:  
12            if p.__eq__(persona):  
13                resultado= persona  
14        return resultado  
15
```

acción o comportamiento

encapsulamiento

```
***  nombresYApellidos***  
|  |  _nombresYApellidos= nombresYApellidos  
***  _nombresYApellidos***
```

que tanto acceso se deja

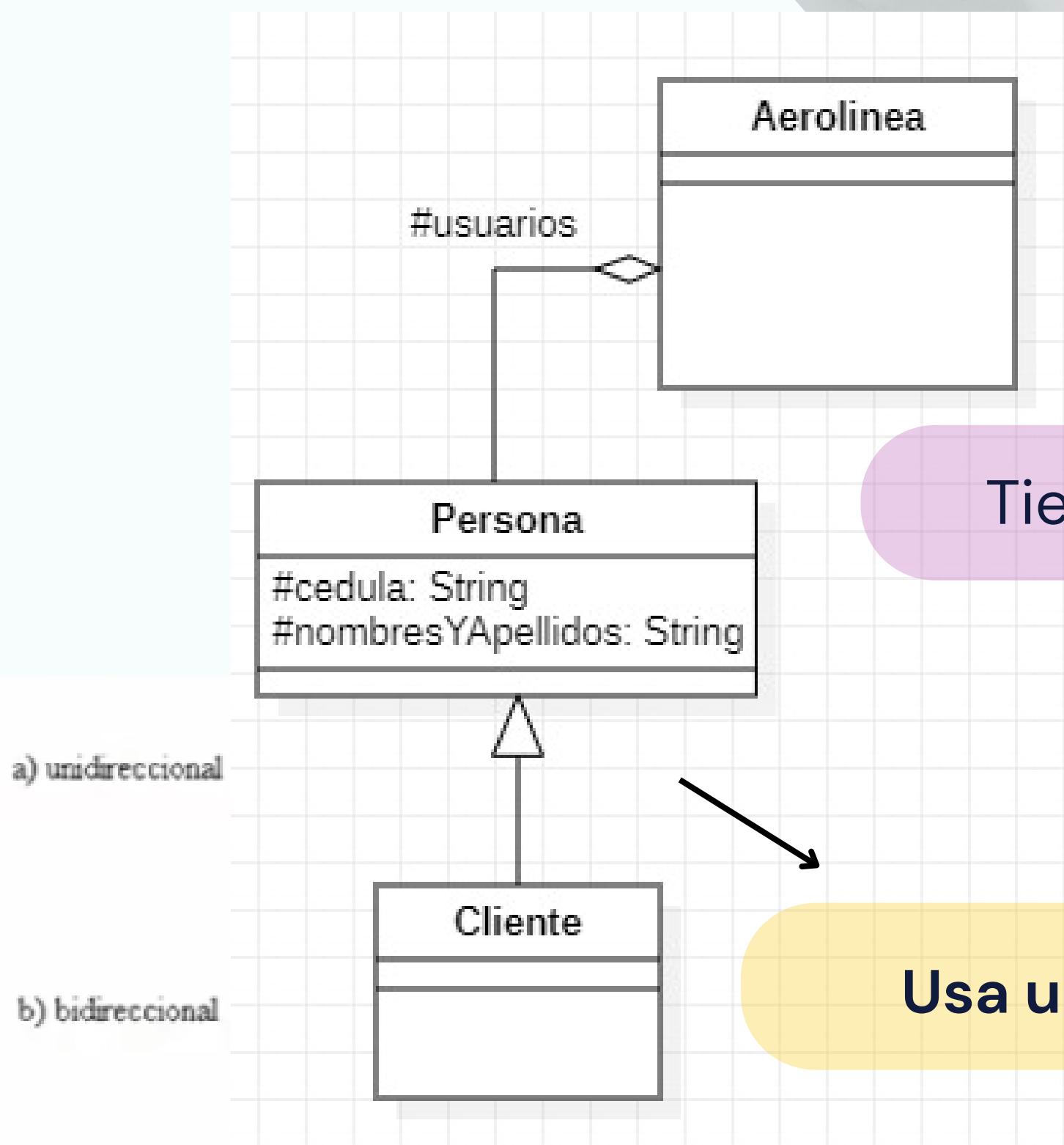
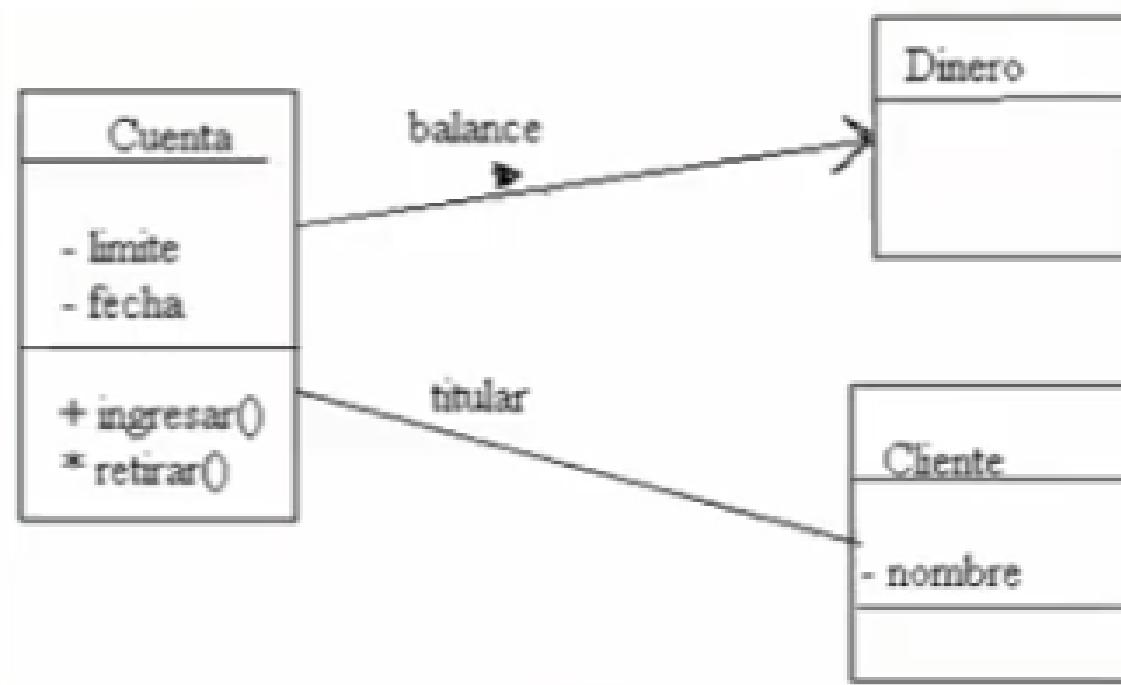


```
dict_name = {"bob":12, "john":11}
list_name = [2, 3, 4, 5, 1]
```

CONCEPTOS Contenedores

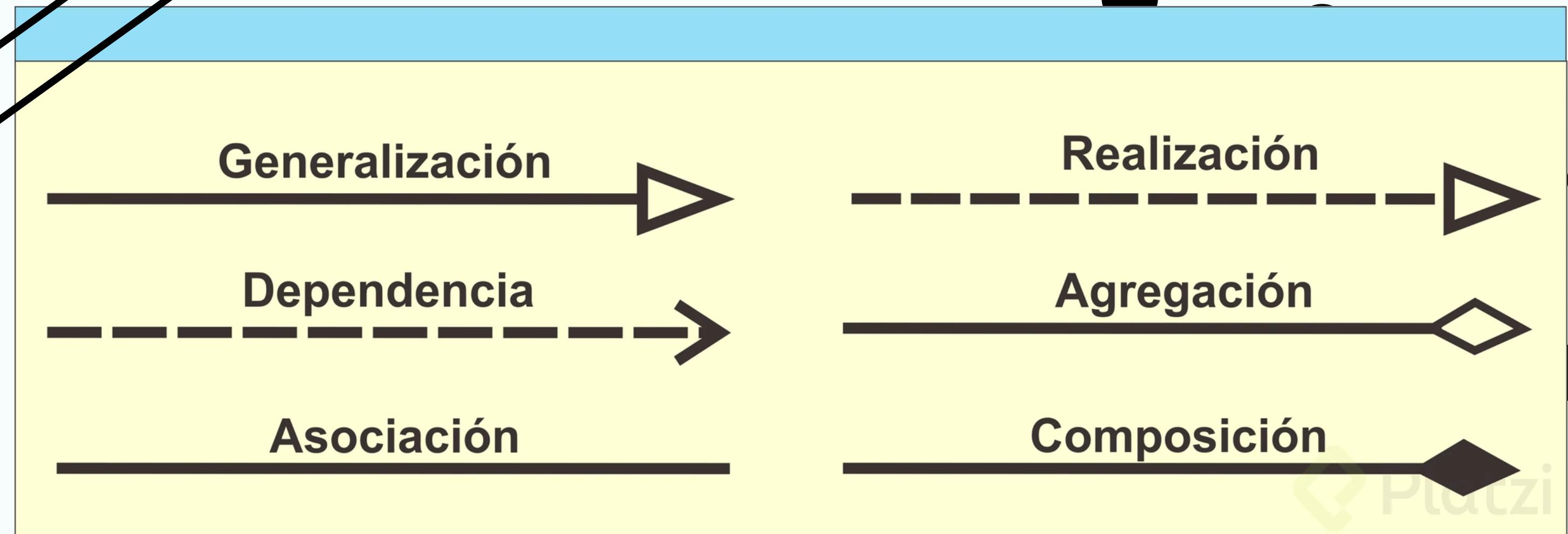
Asociación / Agregación/ Composición

Puede contener atributos y operaciones, por lo que cuando ocurre se tiene una clase de asociaciones.



Tiene un - Cielo → Estrellas

Usa un - Curso -> Alumno



Platzi

Conceptos de Herencia / Polimorfismo

```
class Cliente(Persona):
    def __init__(self, cedula, nombresYApellidos, edad):
        super().__init__(cedula, nombresYApellidos)
        self.edad= edad
```

```
pedro= Persona(2456543, "Pedro")
maria= Persona(5890093, "María")
lucia= Cliente(3456778, "Lucía", 12)
```

```
In [16]: isinstance(lucia, Persona)
Out[16]: True
```

```
In [17]: isinstance(lucia, Cliente)
Out[17]: True
```

```
In [18]: isinstance(pedro, Cliente)
Out[18]: False
```

```
In [19]: isinstance(pedro, Persona)
Out[19]: True
```

```
In [20]: issubclass(Cliente, Persona)
Out[20]: True
```

```
In [21]: issubclass(Persona, Cliente)
Out[21]: False
```

Conceptos de Herencia / Polimorfismo

```
pedro= Persona(2456543, "Pedro")
maria= Persona(5890093, "María")
lucia= Cliente(3456778, "Lucía", 12)
carlos= Empleado(3456789, "Carlos", 1800000)

print("")
prueba.agregar(pedro)
print(pedro.crear_etiqueta())

print("")
prueba.agregar(maria)
print(maria.crear_etiqueta())

print("")
prueba.agregar(lucia)
print(lucia.crear_etiqueta())

print("")
prueba.agregar(carlos)
print(carlos.crear_etiqueta())
```

Se agregó
Pedro
2456543

Se agregó
María
5890093

Se agregó
Lucía
3456778
Edad: 12

Se agregó
Carlos
3456789
Sueldo: 1800000

GUI: Desktop, CLI (comandos/consola), Web y móvil.

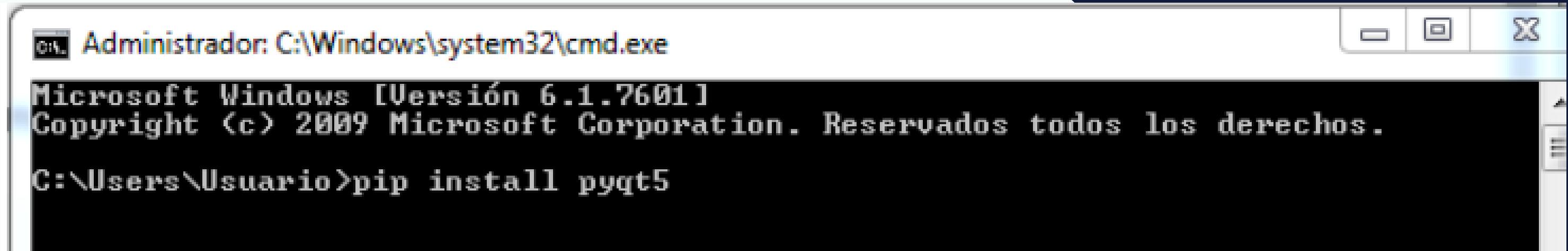
Aplicaciones de escritorio



CLI o interfaz de línea de comandos

```
c:\ C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.16299.461]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\>
```



Administrator: C:\Windows\system32\cmd.exe

Microsoft Windows [Versión 6.1.7601]
Copyright © 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\Usuario>pip install PyQt5

Instalar las herramientas de PyQt5.

Actualizando la versión del “pip” en Windows.

pip install PyQt5-tools

python -m pip install -U pip

Gracias