USC
UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

7. TIPOS ESTRUCTURADOS: CLASES Y DICCIONARIOS

Programación I

Grado en Inteligencia Artificial

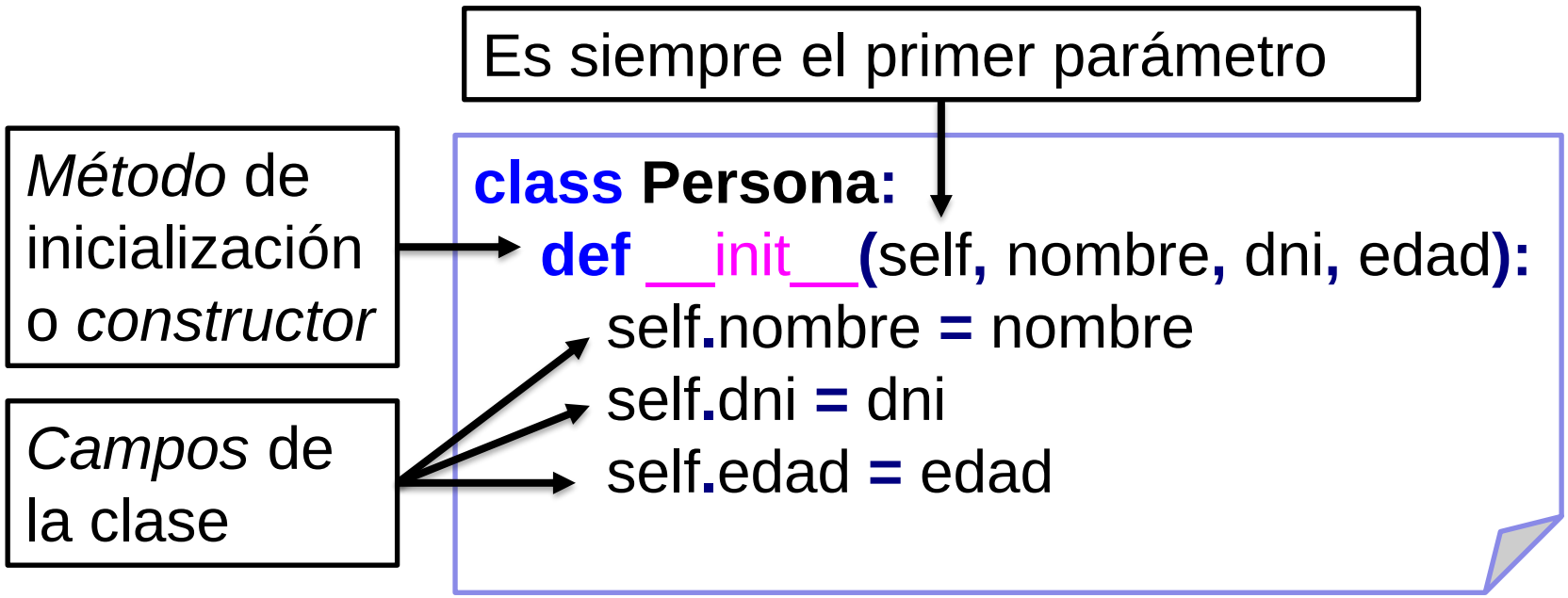
Curso 2022/2023

Contenidos

- Definición de clases en Python
- Referencias y objetos
- Diccionarios

Clases

- Permiten definir **nuevos tipos de datos** *componiendo* tipos existentes



Objetos

- Son *representantes* de las clases

```
toni = Persona('Antonio Pérez', '98761234Q', 20)
juan = Persona('Juan Pérez', '12345678Z', 19)
pedro = Persona('Pedro López', '23456789D', 18)

alumnos = [toni, juan, pedro]
print(toni.edad)
print(alumnos[0].edad)
print(juan.dni)
pedro.edad = 19
```

Especificación más completa

```
class Persona:
    def __init__(self, nombre, dni, edad):
        self.nombre = nombre
        self.dni = dni
        self.edad = edad

    def iniciales(self):
        cadena = ''
        for caracter in self.nombre:
            if caracter >= 'A' and caracter <= 'Z':
                cadena = cadena + caracter + '. '
        return cadena

    def __str__(self):
        cadena = 'Nombre: {0}\n'.format(self.nombre)
        cadena = cadena + 'DNI: {0}\n'.format(self.dni)
        cadena = cadena + 'Edad: {0}\n'.format(self.edad)
        return cadena
```

```
toni = Persona(…)
juan = Persona(…)
pedro = Persona(...)

alumnos = [toni, juan, pedro]
for alumno in alumnos:
    print(alumno.dni)

print(juan.iniciales())
print(juan)
```

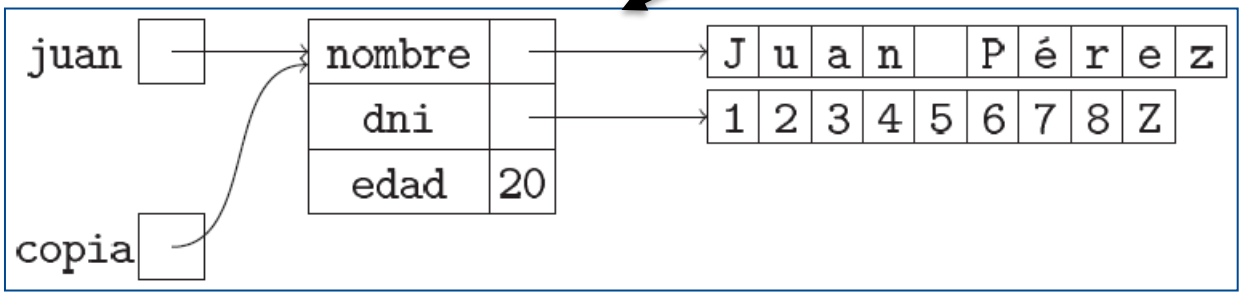
Más métodos especiales

- **__len__(self)**: permite aplicar **len()** para obtener la longitud del objeto
- **__add__(self, otro)**: permite aplicar el operador **+** entre objetos de la clase
- **__mul__(self, otro)**: permite aplicar el operador ***** entre objetos de la clase
- **__cmp__(self, otro)**: permite aplicar los operadores de comparación

Copia de objetos

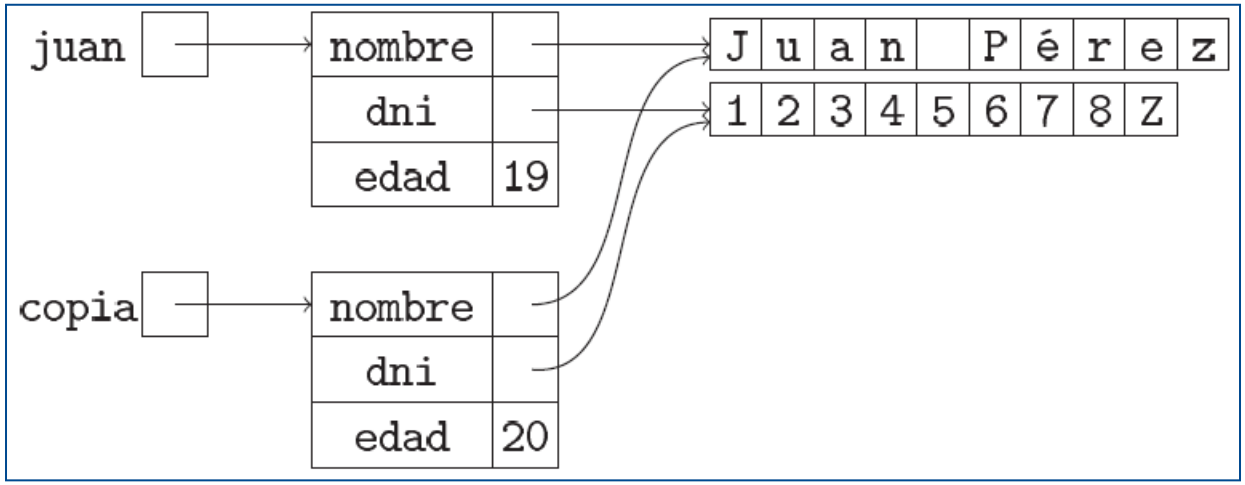
```
juan = Persona('Juan Pérez', '12345678Z', 19)
copia = juan
copia.edad = 20
print(juan.edad)
```

Ojo, se está copiando una referencia



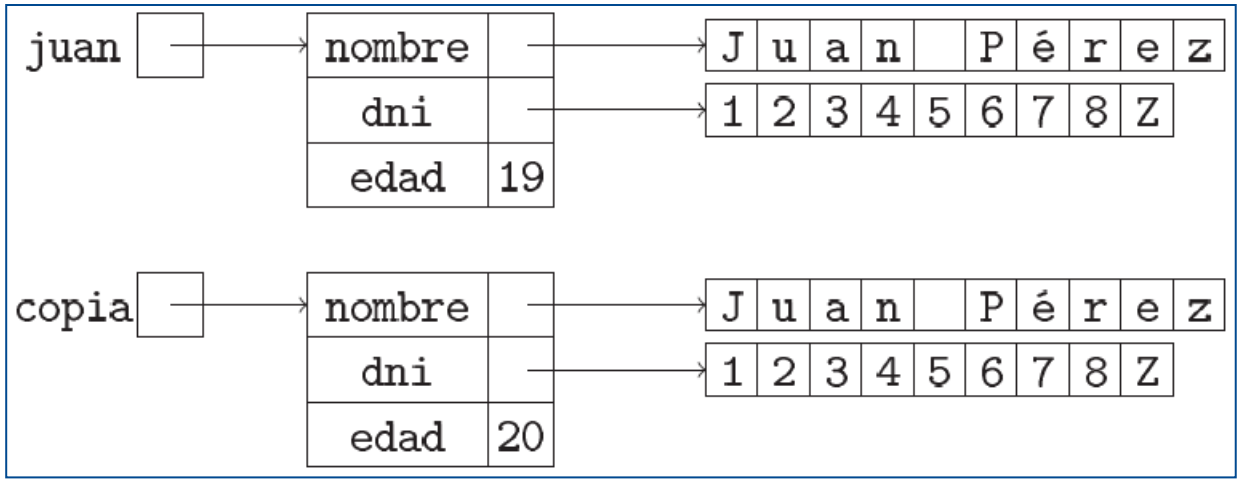
Copia superficial

```
class Persona:
    ...
    def copiar(self):
        nuevo = Persona(self.nombre, self.dni, self.edad)
        return nuevo
```



Copia profunda

```
class Persona:
    ...
    def copiar(self):
        nuevo = Persona(self.nombre[:], self.dni[:], self.edad)
        return nuevo
```



Ejemplos de clases en Marzal et al.

- Gestión de **calificaciones** de estudiantes: pp. 348 – 351
- Clase para manejar **fechas**: pp. 352 – 354
- **Colas y pilas**: pp. 355 – 358
- **Conjuntos**: pp. 358 – 360
- Gestión de un **videoclub**: pp. 360 – 375

Diccionarios

- Establecen **correspondencias** entre *claves* y *valores*
- El **tipo de las claves** puede ser cualquiera de los inmutables: cadenas, enteros, flotantes (y otros)
- Se accede al valor que corresponde a una clave dada usando los **corchetes**
- Los diccionarios **no están ordenados**

Ejemplo de diccionario

```
d = {} #diccionario inicialmente vacío
d['Juan'] = '964 37 64 32'
d['Luis'] = '964 73 46 23'
d['Ana'] = '96 287 98 99'
d['María'] = '964 22 10 00'

print(d['Juan']) #valor correspondiente a 'Juan'
print('Pedro' in d) #comprobación de clave válida
for e in d: #recorrido del diccionario
    print(e)
for k in d.keys(): #recorrido de las claves
    print(k)
for v in d.values(): #recorrido de los valores
    print(v)

del d['María'] #borrado de una entrada
```

Ejemplos de diccionarios en Marzal

- **Listín telefónico**: pp. 378 – 380
- **Contador de palabras**: pp. 380 – 381
- **Videoclub**: pp. 381 – 383