



Taller de Programación Web

Clases y Objetos



Subsecretaría de
Empleo
Chaco Gobierno de todos



Ministerio de
Producción, Industria y Empleo
Chaco Gobierno de todos

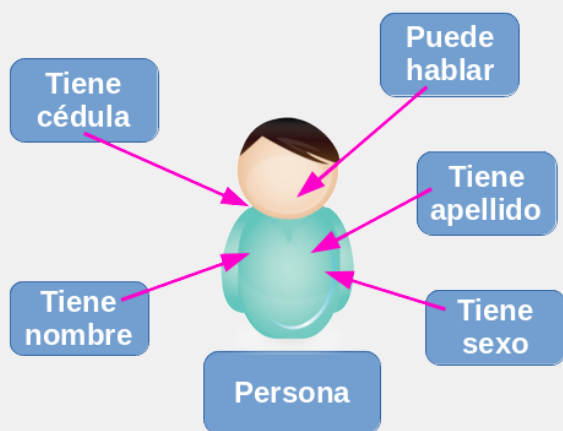


CHACO
Gobierno de todos



PARADIGMA DE PROGRAMACIÓN ORIENTADA A OBJETOS

Un **paradigma** es el resultado de un proceso social en el cual un grupo de personas desarrolla nuevas ideas y crea principios y prácticas alrededor de estas ideas.



En programación, se trata de un enfoque concreto de desarrollar y estructurar el desarrollo de programas en base a Objetos.

Clase

Una **clase** es un modelo que define las propiedades y comportamiento de un tipo de objeto **concreto**, en la instanciación se hace la lectura de estas definiciones y se crea un objeto a partir de ellas. Las clases son abstracciones que representan a un conjunto de objetos con un comportamiento e interfaz común.

Una clase se compone de dos partes:

- **Atributos:** esto es, los datos que se refieren al estado del objeto.
- **Métodos:** son funciones que pueden aplicarse a objetos.



Objeto

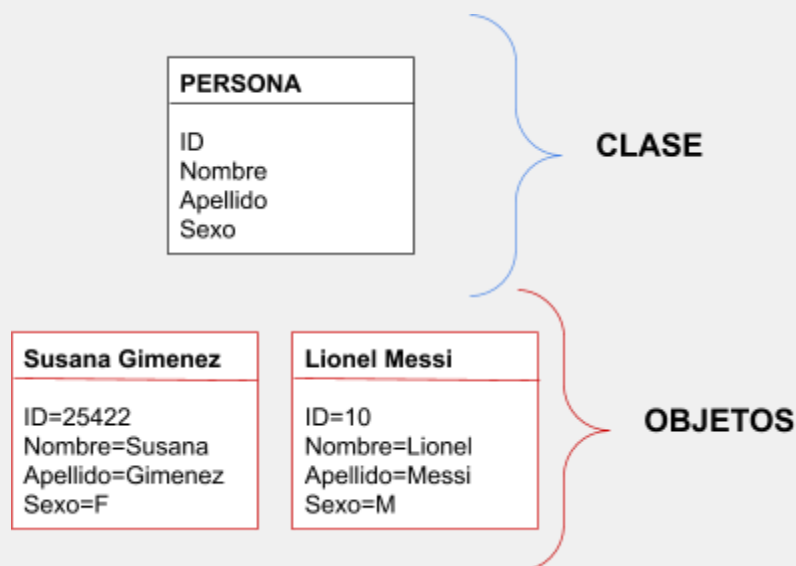
Entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos), los mismos que consecuentemente reaccionan a eventos. Se corresponden con los objetos reales del mundo que nos rodea, o con objetos internos del sistema (del programa). En realidad, un objeto es una instancia de una clase, por lo que se pueden intercambiar los términos objeto o instancia. Un ejemplo de objeto podría ser un automóvil, en el que tendríamos atributos como la marca, el número de puertas o el color y métodos como arrancar y parar. O bien cualquier otra combinación de atributos y métodos según lo que fuera relevante para nuestro programa.

Un objeto consta de:

- **Tiempo de vida:** La duración de un objeto en un programa siempre está limitada en el tiempo. La mayoría de los objetos sólo existen durante una parte de la ejecución del programa. Los objetos son creados mediante un mecanismo denominado instanciación, y cuando dejan de existir se dice que son destruidos.
- **Estado:** Todo objeto posee un estado, definido por sus atributos. Con él se definen las propiedades del objeto, y el estado en que se encuentra en un momento determinado de su existencia.
- **Comportamiento:** Todo objeto ha de presentar una interfaz, definida por sus métodos, para que el resto de objetos que componen los programas puedan interactuar con él.
- **Identidad:** Propiedad del objeto que lo distingue de los demás.

El equivalente de un objeto en el paradigma estructurado sería una variable. Así mismo la instanciación de objetos equivaldría a la declaración de variables, y el tiempo de vida de un objeto al ámbito* de una variable.

*La zona del programa en la que una variable puede ser utilizada es lo que se conoce como su ámbito. Decimos que el ámbito de una variable local está limitado al código de la función en la que está definida.



Para modelar soluciones usando Paradigma Orientado a Objetos, comúnmente se utilizan los **Diagramas de Clases**, es uno de los diagramas incluidos en UML 2.5 clasificado dentro de los diagramas de estructura y, como tal, se utiliza para representar los elementos que componen un sistema de información desde un punto de vista estático.

Nombre de la Clase
+atributos
+métodos

Visibilidad

(+) Pública. Representa que se puede acceder al atributo o función desde cualquier lugar de la aplicación.

(-) Privada. Representa que se puede acceder al atributo o función únicamente desde la misma clase.

(#) Protegida. Representa que el atributo o función puede ser accedida únicamente desde la misma clase o desde las clases que hereden de ella (clases derivadas).

Animal
-Especie -Nombre
+Animal() +setEspecie() +getEspecie() +setNombre() +getNombre()

Te recomendamos revisar el Tutorial Online sobre Diagrama de Clases del Aula virtual



En Python las clases se definen mediante la palabra clave **class** seguida del nombre de la clase, dos puntos (:) y a continuación, indentado, el cuerpo de la clase.

Código

Definir la clase Persona con sus métodos y atributos .

```
class Persona:
    def __init__(self, nombre, edad, genero, estatura, peso):
        self.nombre = nombre
        self.edad = edad
        self.genero = genero
        self.estatura = estatura
        self.peso = peso

    def hablar(self):
        print("Hola soy {}".format(self.nombre))

    def correr(self):
        print("{} está corriendo".format(self.nombre))

    def caminar(self):
        print("{} está caminando".format(self.nombre))
```

El método **__init__**, con una doble barra baja al principio y final del nombre, se ejecuta justo después de crear un nuevo objeto a partir de la clase, proceso que se conoce con el nombre de instanciación. El método **__init__** sirve, como sugiere su nombre, para realizar cualquier proceso de inicialización que sea necesario como puede ser la asignación de valores a atributos, cálculos, llamadas a otras funciones o procedimientos.



Como vemos el primer parámetro de `__init__` y del resto de métodos de la clase es siempre ***self*** que sirve para referirse al objeto actual.

Este mecanismo es necesario para poder acceder a los atributos y métodos del objeto diferenciando, por ejemplo, una variable local ***mi_var*** de un atributo del objeto ***self.mi_var***.

Si volvemos al método `__init__` de nuestra clase Persona veremos cómo se utiliza ***self*** para asignar al atributo nombre del objeto (***self.nombre***) el valor que el programador especifica para el parámetro nombre. El parámetro nombre se destruye al final de la función, mientras que el atributo nombre se conserva (y puede ser accedido) mientras el objeto viva.

Para crear un objeto se escribiría el nombre de la clase seguido de cualquier parámetro que sea necesario entre paréntesis. Estos parámetros son los que se pasarán al método `__init__`, que como decíamos es el método que se llama al instanciar la clase.

Código

Definir la clase Persona con sus métodos y atributos y crear objetos a partir de ella.

```
class Persona:
    def __init__(self, nombre, edad, sexo, estatura, peso):
        self.nombre = nombre
        self.edad = edad
        self.sexo = sexo
        self.estatura = estatura
        self.peso = peso

    def hablar(self):
        print("Hola soy {}".format(self.nombre))

    def correr(self):
```



```
print("{} está corriendo".format(self.nombre))

def caminar(self):
    print("{} está caminando".format(self.nombre))

persona1 = Persona("Juan", 23, "Masculino", 170, 90)

persona2 = Persona("Maria", 25, "Femenino", 160, 70)
```

En el ejemplo podemos ver que aunque la definición de la función indica claramente que precisa de seis parámetros (self, nombre, edad, sexo, estatura y peso), a la hora de crear un objeto de esta clase pasaremos solo cinco parámetros. Esto es así porque Python pasa el primer argumento (la referencia al objeto que se crea) automáticamente.

Ahora que ya hemos creado nuestros objetos, podemos acceder a sus atributos y métodos mediante la sintaxis ***objeto.atributo*** y ***objeto.metodo()***:

Código

Definir la clase Persona con sus métodos y atributos, crear un objeto y acceder a sus métodos y atributos.

```
class Persona:
    def __init__(self, nombre, edad, sexo, estatura, peso):
        self.nombre = nombre
        self.edad = edad
        self.sexo = sexo
        self.estatura = estatura
        self.peso = peso
```




```
def hablar(self):  
    print("Hola soy {}".format(self.nombre))  
  
def correr(self):  
    print("{} está corriendo".format(self.nombre))  
  
def caminar(self):  
    print("{} está caminando".format(self.nombre))  
  
persona1 = Persona("Juan", 23, "Masculino", 170, 90)  
  
persona2 = Persona("Maria", 25, "Femenino", 160, 70)  
  
print(persona1.nombre)  
print(persona1.edad)  
print(persona1.sexo)  
print(persona1.estatura)  
print(persona1.peso)  
persona1.hablar()  
persona1.correr()  
persona1.caminar()
```

Resultado

Juan

23

Masculino

170

90



INFORMATARIO

Hola soy Juan
Juan está corriendo
Juan está caminando

Clase: Persona

El objeto instanciado (creado a partir de esa clase): persona1

Atributos:

- nombre: "Juan"
- edad: 23
- genero: "Masculino"
- estatura: 170
- peso: 90

Métodos:

- hablar()
- correr()
- caminar()

