

Programación 2



**Tecnicatura en Desarrollo de Aplicaciones
Informáticas**

Repaso

P00 con Perritos



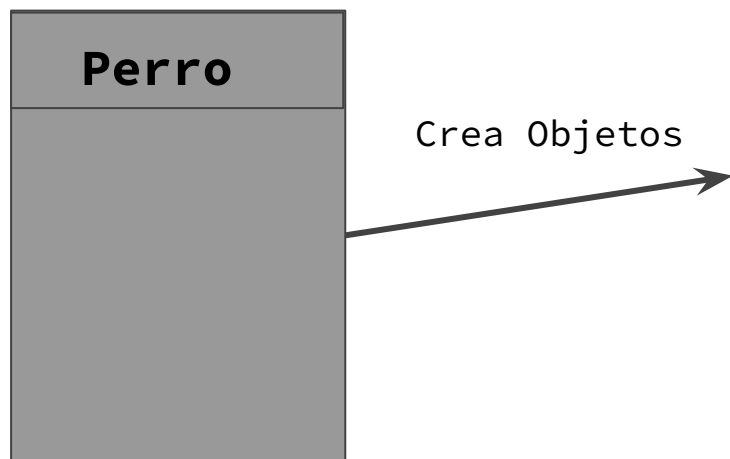
Clase

Una clase es una plantilla o modelo para crear objetos

En una clase se define el comportamiento y las propiedades que tendrán los objetos que se creen a partir de ella

Clase Perro

Con nuestro ejemplo el perro es una clase que define a los animales y las mascotas son objetos de esa clase.



Clase



Objetos
Instancias

Perro

La clase Perro es la que define los objetos que son considerados Perros, esa clase define LO QUE TIENE y LO QUE PUEDE HACER

ESTADO ---> LO QUE TIENE

COMPORTAMIENTO ---> LO QUE SABE HACER

Perro - Estado

Raza -----> Define la raza a la que pertenece el perro

Nombre ---> Define el nombre que tiene el perro

Edad ----> Define la edad que tiene el perro

~~Cantidad de Ruedas ----> Ruedas que tiene un perro~~

Consejo: El perro solo tiene cosas del perro

Perro - Comportamiento

Puedo preguntar el nombre, puedo cambiar el nombre.

Puedo preguntar la raza, ¿Puedo cambiar la raza?

Puedo preguntar la edad, puedo cambiar la edad.

Sabe ladrar

Sabe mover la cola

Sabe saltar

Perro Java - Definir la clase

```
public class Perro {
```

Definir la clase Perro, siempre el nombre de la clase empieza con Mayúscula

```
    LLaves {    }
```

Todo lo que está dentro de estas dos llaves es lo que pertenece al Perro, todo el código que conoce

```
}
```


Perro Java - Atributos

```
public class Perro {
```

```
    String nombre;
```

```
    String raza;
```

```
    int edad;
```

```
}
```

Tipo del atributo

Nombre del Atributo

la variable nombre es el nombre del perro, es una palabra por eso el *tipo* de la variable es **String**

la variable raza es la raza del perro, es una palabra por eso el *tipo* de la variable es **String**

la variable edad representa la edad del perro, es un número por eso el *tipo* de la variable es **int**

IMPORTANTE, las variables y atributos siempre van con minúscula así nos damos cuenta rápido de lo que es una Clase (cuando se usa como un tipo de variable) y el nombre de una variable

Perro Java - Comportamiento

El comportamiento se define por los métodos que le vamos a agregar al Perro

Son las cosas que dejamos que se le puede hacer a un perro y las cosas que no

Perro Java - pedir el nombre

Definición del Metodo

```
public String getNombre() {
```

QUE devuelve el método

```
return nombre;
```

Palabra reservada se usa para decir lo que devuelve el método

el **Atributo** del Objeto

IMPORTANTE: ENCAPSULAMIENTO

Nadie desde afuera conoce mis atributos, por eso si alguien quiere saber mi nombre me debe preguntar y yo se lo digo. Si no existe método, no me puede preguntar

Perro Java - Cambiar el nombre

Palabra reservada para decir que NO DEVUELVO NADA

```
public void setNombre(String nuevoNombre) {  
    nombre = nuevoNombre;  
}
```

PARÁMETROS: las “cosas” que recibe el método. Son variables nuevas

IMPORTANTE: Los nombres siguen una convención que se llama camelCase, empiezan con minúscula y cada palabra extra siempre empieza con mayúscula. por ejemplo `getNombre`, `setNombre`.

Siempre pedir un valor se usa `get` al inicio del método y para cambiar se usa `set`

Perro Java - Cambiar el nombre

Palabra reservada para decir que NO DEVUELVO NADA

```
public void setNombre(String nuevoNombre) {  
    nombre = nuevoNombre;  
}
```

PARÁMETROS: las “cosas” que recibe el método. Son variables nuevas

Los parámetros son las variables! no los valores. Son las “cajas” que van a llevar los valores de un lado a otro. Por eso en este caso se llama nuevoNombre (también siguen notación camelCase)

Perro Java - la raza

```
public String getRaza(){  
    return raza;  
}
```

```
public void setRaza(String nuevaRaza){  
    raza = nuevaRaza;  
}
```

Perro Java - la Edad

```
public int getEdad(){  
    return edad;  
}
```

```
public void setEdad(int nuevaEdad){  
    edad = nuevaEdad;  
}
```

Tiene sentido que le cambie la edad así de golpe o se suma de a 1 solo año?

Perro Java - la Edad

```
public int getEdad(){  
    return edad;  
}
```

Tiene sentido que le cambie la edad así de golpe o se suma de a 1 solo año?

Si no queremos que se pueda hacer no ponemos el método setEdad

Perro Java - la Edad

```
public int getEdad(){  
    return edad;  
}
```

Tiene sentido que le cambie la edad así de golpe o se suma de a 1 solo año?

Si no queremos que se pueda hacer no ponemos el método setEdad

```
public void sumarEdad() {  
    edad = edad + 1;  
}
```

Podemos agregar un método que sume de a uno la edad

Comportamiento

Se define por los métodos
del objeto

Los métodos que ponemos o
no ponemos son el
comportamiento de nuestro
objeto lo que sabe hacer y
lo que no sabe hacer

— — —

Definición de clases en Java- Métodos y atributos

¿hay algún orden entre ellos? No, los métodos, atributos, pueden ser declarados en cualquier orden

Entonces...¿Es recomendable ponerlos en cualquier orden?

¡No! Por convención de código, se recomienda colocar atributos **primero** y métodos **después**.

Nota: Todos los desarrolladores Java suelen adherir a las convenciones, para facilitar compartir el código en equipos de desarrollo.

Constructores

Constructores

Se usan para el momento en que se crea el objeto.

Nos permite usar los valores que nos dice el usuario o poner valores por defecto

Perro Java - Constructor por defecto

```
public Perro() {  
    nombre = "Pipa";  
    raza = "callejero";  
    edad = 0;  
}
```

Constructor

Es un método que se llama IGUAL que la clase y no tiene un "Valor" de retorno

Este es el constructor por defecto porque los valores los decidimos nosotros al hacer el código de la clase perro;
Se invoca cuando hacemos:

```
Perro miPerro = new Perro();
```

Perro Java - Constructor con parametros

```
public Perro(String nuevoNombre, String nuevaRaza) {  
    nombre = nuevoNombre;  
    raza = nuevaRaza;  
    edad = 0;  
}
```

Este es el constructor con parametros se llama cuando el usuario hace:

```
Perro miPerro = new Perro("Negrito","Shar Pei");
```

Perro Java - Constructor reuso de los set

```
public Perro(String nuevoNombre, String nuevaRaza) {  
    setNombre(nuevoNombre);  
    setRaza(nuevaRaza);  
    edad = 0;  
}
```

en lugar de asignar los valores directamente se puede invocar a los set, y si hay algún control se verifica en ese lugar

No existía el set edad por eso se inicializa en 0

Perro Java - Constructor reuso de constructores

```
public Perro() {  
    this("Pipa","Callejero");  
}
```

SOLO EN LA PRIMER LINEA se puede invocar a otro constructor, en este caso invocamos al anterior pero poniendo los valores por defecto

Evitamos repetir código

Constructores

Si quiero que SI O SI el usuario inicie el objeto con nombre y raza no agregé el constructor vacío. Cuando el usuario ponga:

~~Perro miPerro = new Perro();~~

Java solo le va a decir que hay un error;

Ejecutar en Java

¿Cómo ejecutar un programa en Java?

Para que una clase sea ejecutable debe tener un método especial que es el que se invoca.

```
public static void main(String[] args) {
```

```
    //El código a Ejecutar
```

```
}
```

Ejemplo

```
public static void main(String[] args) {
```

```
    //El código a Ejecutar
```

```
    Perro perro1= new Perro();  
    perro1.setNombre("Pakkun");  
    perro1.setRaza("Pug");
```

```
    Perro perro2 = new Perro("Akamaru", "Gran Pirineo");
```

```
}
```

Creó el objeto
y después le
"Seteo" los
valores

Creó el objeto ya con
los valores

Imprimir por consola

```
System.out.print("Mensaje"); // imprime y queda el cursor
System.out.println("Mensaje"); // imprime y hace salto de línea

System.out.println("nombre : " + perro1.getNombre());
System.out.println("nombre : " + perro2.getNombre());
```

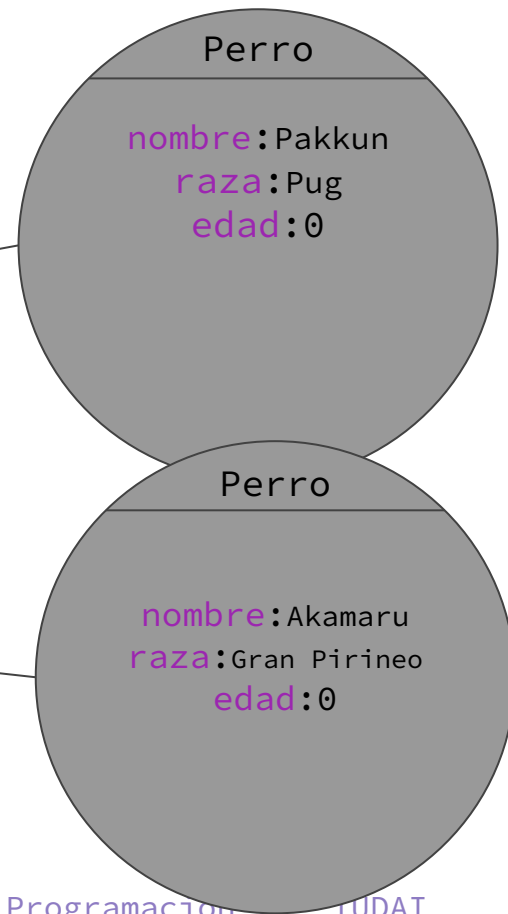
Asignación de objetos en Java

Las variables “Apuntan” a los objetos

VARIABLES:

perro1

perro2



Asignación de objetos en Java

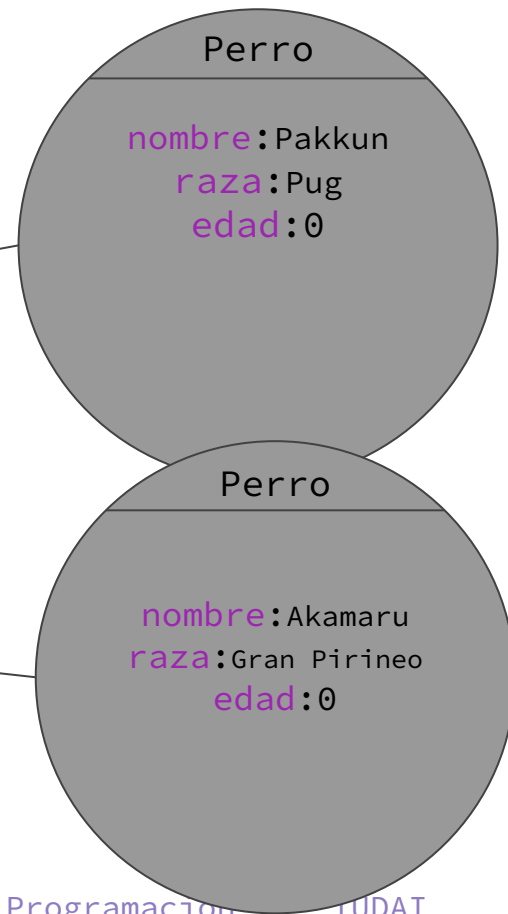
Las variables “Apuntan” a los objetos

VARIABLES:

perro1

perro2

```
perro1 = perro2;
```



Asignación de objetos en Java

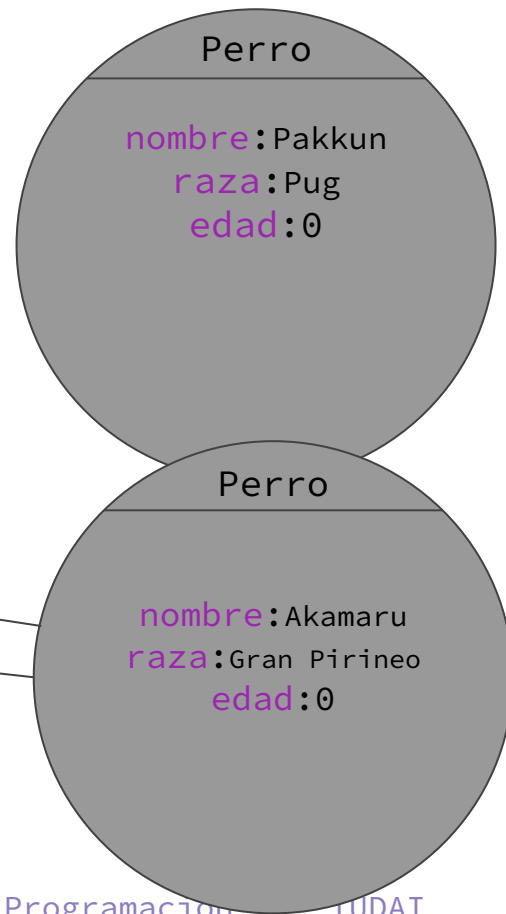
Las variables “Apuntan” a los objetos

VARIABLES:

perro1

perro2

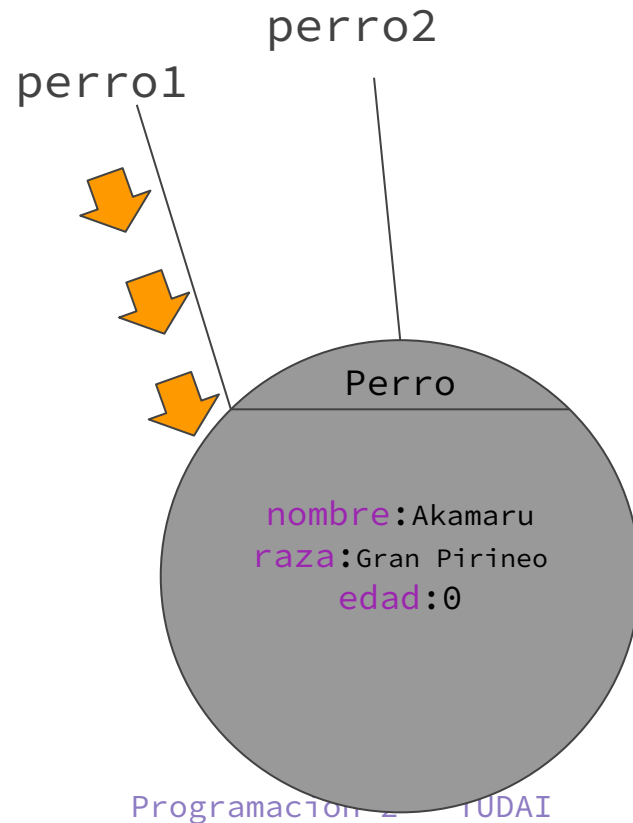
```
perro1 = perro2;
```



Asignación de objetos en Java

si ahora hago

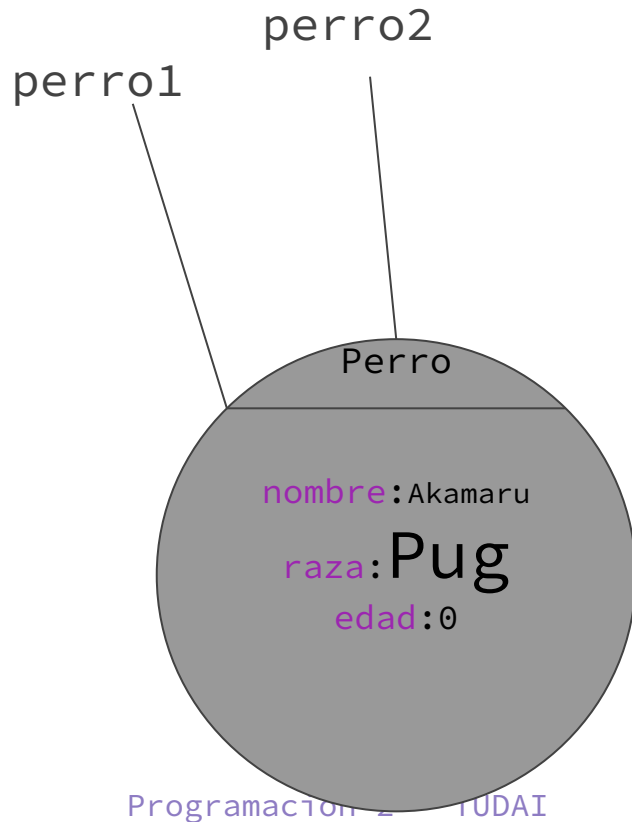
```
perro1.setRaza("Pug");
```



Asignación de objetos en Java

si ahora hago

```
perro1.setRaza("Pug");
```



Asignación de objetos en Java

si ahora hago

```
perro1.setRaza("Pug");  
System.out.println("Raza: " + perro2.getRaza());
```

Raza: Pug

