Java this vs this() es una pregunta habitual cuando comenzamos a trabajar con este lenguaje. ¿Para qué vale exactamente el operador this y en que situaciones la utilizamos?. Vamos a ver varios ejemplos de situaciones en las cuales se puede usar este operador.

CURSO SPRING BOOT
GRATIS
APUNTATE!!

```
package com.arquitecturajava.usothis;

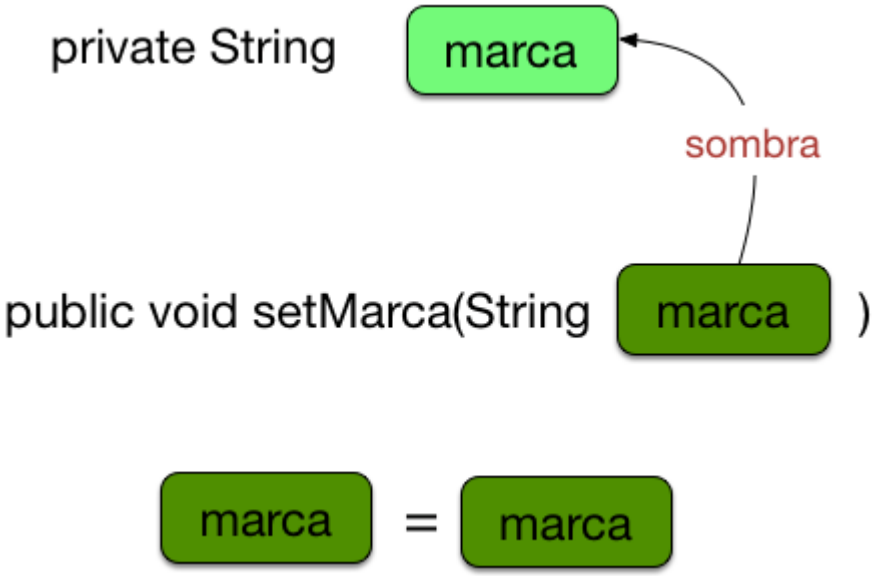
public class Ordenador {

 private String marca;

 public void setMarca(String marca) {

 marca=marca;
 }
}
```

Este código compila pero no funciona correctamente ya que estamos igualando la variable marca que recibe el método consigo misma debido a algo que en Java se denomina shadow de variables (una variable hace sombra a otra con su mismo nombre).



En este caso la variable local del método setMarca hace sombra a la variable de instancia marca. Por lo tanto si escribimos marca=marca , igualaremos la variable local consigo misma. Esto no tendrá ningún efecto en la variable de instancia. Una primera solución al problema sería cambiar el nombre de la variable local.

```
package com.arquitecturajava.usothis;

public class Ordenador {

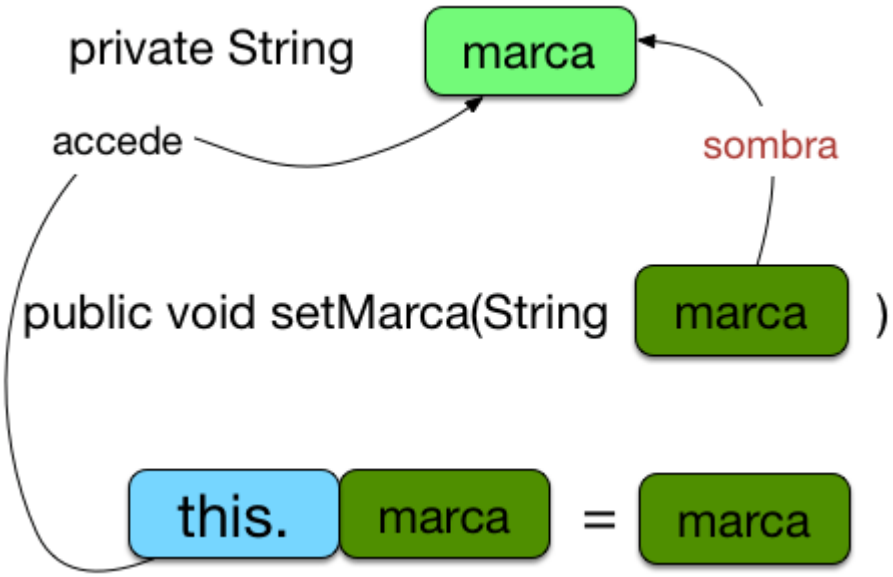
private String marca;

public void setMarca(String otraMarca) {

marca=otraMarca;
}
}
```

Aunque este código funciona no ayuda demasiado a clarificar ya que estamos añadiendo otro nombre más “otraMarca” a nuestra clase. Para no tener este problema podemos usar el operador de ámbito this. Este operador clarifica a que variable nos referimos (la variable de

instancia de nuestra clase).



Ahora si que cambiaremos el valor de la variable de instancia de nuestro objeto Ordenador.

```
package com.arquitecturajava;

public class Ordenador {

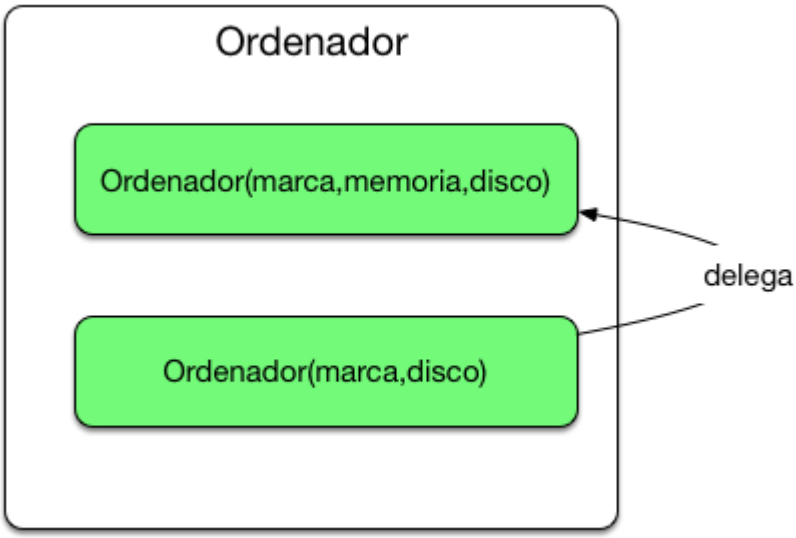
    private String marca;

    public void setMarca(String marca) {
        this.marca = marca;
    }

    public String getMarca() {
        return marca;
    }
}
```

Java this vs this()

Una vez que hemos entendido como funciona this como operador de ámbito ,es momento de ver la palabra reservada this() . Esta palabra permite realizar invocaciones entre constructores que están en la misma clase.



Podemos reutilizar el código de un constructor en otro:

```
package com.arquitecturajava.usothis;

public class Ordenador {

    private String marca;
    private int memoria;
    private int capacidadDisco;

    public Ordenador(String marca, int memoria, int capacidadDisco) {
        this.marca = marca;
        this.memoria = memoria;
        this.capacidadDisco = capacidadDisco;
```

```
    }

    public Ordenador(String marca, int capacidadDisco) {
        this(marca, 4, capacidadDisco);
    }

    public String getMarca() {
        return marca;
    }

    public void setMarca(String marca) {
        this.marca = marca;
    }

    public int getMemoria() {
        return memoria;
    }

    public void setMemoria(int memoria) {
        this.memoria = memoria;
    }

    public int getCapacidadDisco() {
        return capacidadDisco;
    }

    public void setCapacidadDisco(int capacidadDisco) {
        this.capacidadDisco = capacidadDisco;
    }
}
```

En este caso hemos usado java this() para invocar a un constructor desde otro. En esta

aspecto actua como un operador de ámbito para los diferentes constructores.

CURSO SPRING BOOT

GRATIS

APUNTATE!!

Otros artículos relacionados:

- Java Constructores this() y super()
- JavaScript apply vs call
- DTO y Inmutabilidad
- Java Encapsulamiento y reutilización