

★ OOP 02 ★
Nueva edición - 2020



PROGRA_AMANDO

holamundo.co

Agenda

Valores por defecto
¿Qué es un constructor?
Constructor por defecto
Firma de un constructor
Palabra reservada this
Relaciones entre clases
Ejercicios

Valores por defecto

Cuando se crea un objeto a partir de una clase. A los miembros de datos, se les asigna un valor por defecto, que depende del tipo de dato.

Galleta	
-	nombre:cadena
-	valor:flotante
-	tiene_mani:booleano
+	setter y getter nombre
+	setter y getter valor
+	setter y getter tiene_mani
+	FechaVencimiento():cadena
+	RemojarEnLeche():booleano

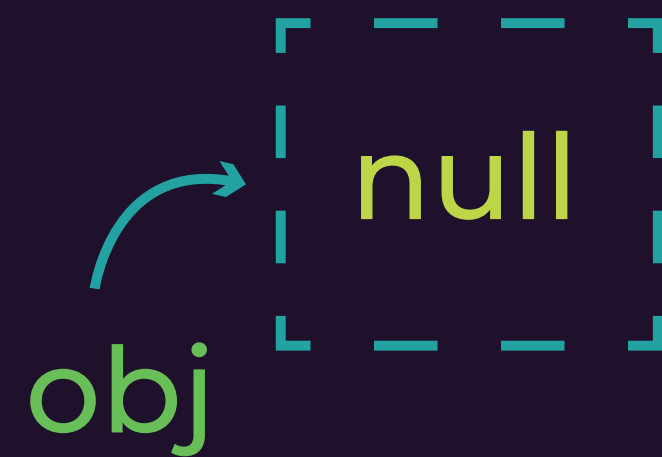
Si se crea una instancia de la clase Galleta, al usar los getters se obtienen un valor, es decir las variables se inicializan cuando se crea el objeto.

!

No se recomienda asignar un valor directamente al miembro de datos en la clase.

Valores por defecto

```
Galleta obj;
```



Al declarar la variable `obj`, solo se reserva un espacio **null** en memoria, **NO hay** información, ya que no existe el objeto.

Al usar el operador **new** se crea el objeto y se asignan los valores iniciales a todos los miembros dato.

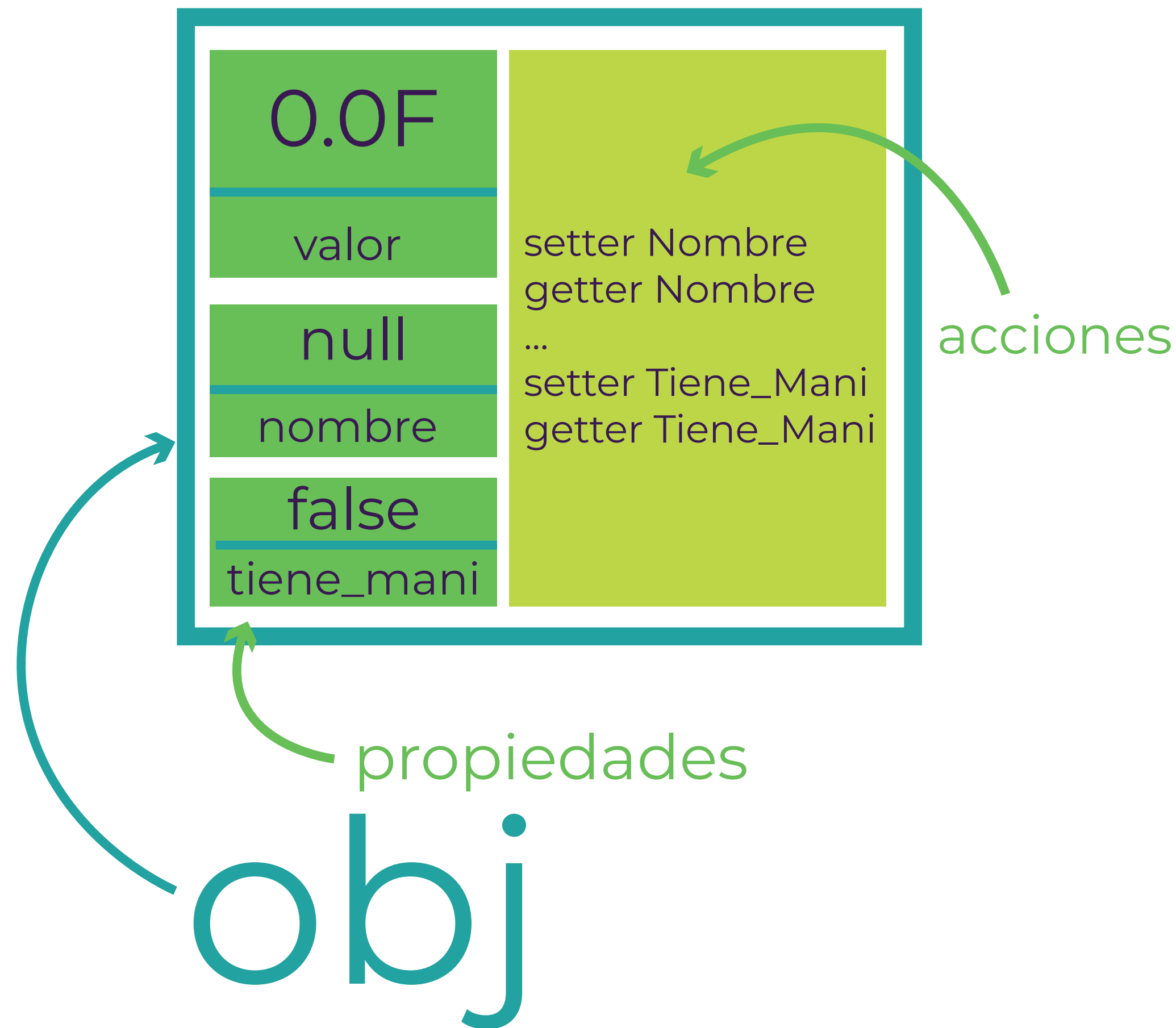
```
obj = new Galleta();
```

```
Console.Write(obj.Tiene_mani);
```

```
Cmd > false
```

Si se imprime usando el getter de la clase de algún miembro dato, obtenemos un valor.

Valores por defecto



Cuando se crea un objeto a partir de una clase, sus miembros datos, se inicializan con un valor por defecto, dependiendo del tipo de dato.

!

Para ver una lista de los valores por defecto de los datos primitivos visitar el siguiente enlace



La imagen muestra la representación de una instancia acabada de crear

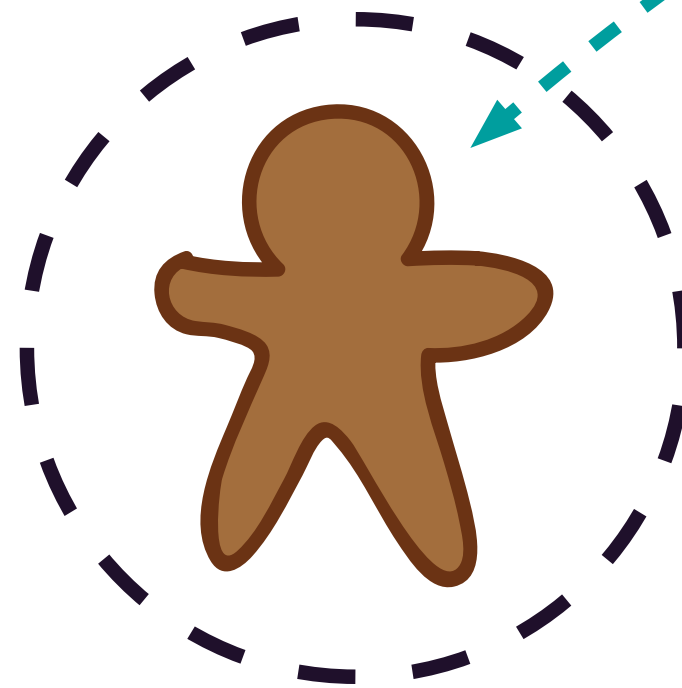
Valores por defecto

Cuando se crea un objeto a partir de la definición de su clase, se crean copias genéricas.

Para personalizar la copia, es decir el objeto, se modifican sus características, cambiando sus propiedades usando los setter y getters.



Recuerde
encapsular los
miembros dato



```
Galleta galleta_01 = new Galleta();
```

Se crea una instancia genérica de la clase.



```
Galleta galleta_02 = new Galleta();  
galleta_02.Nombre = "James";  
galleta_02.Apellido = "Bond";
```

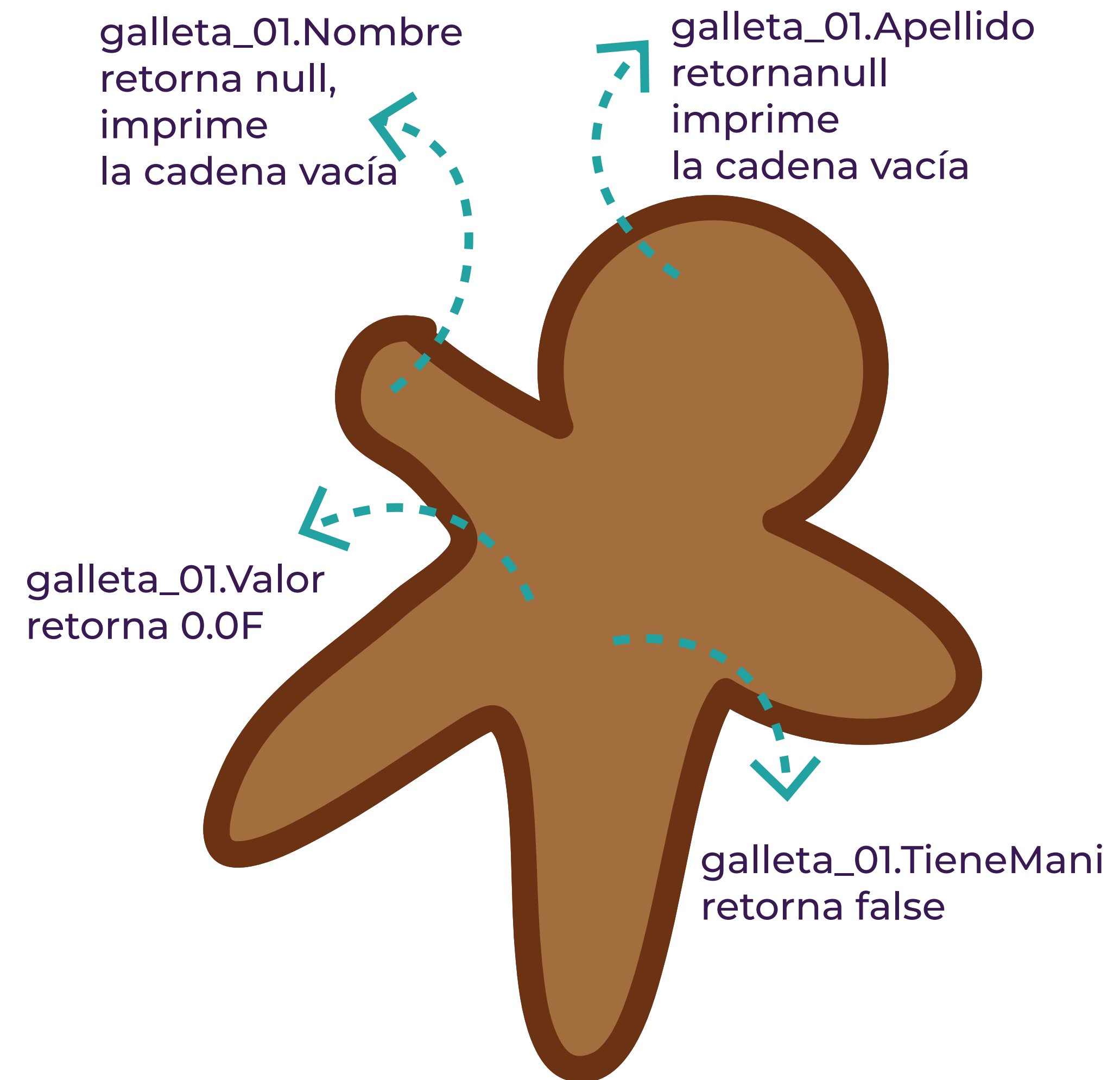
Se personaliza, el objeto, cambiando sus propiedades.

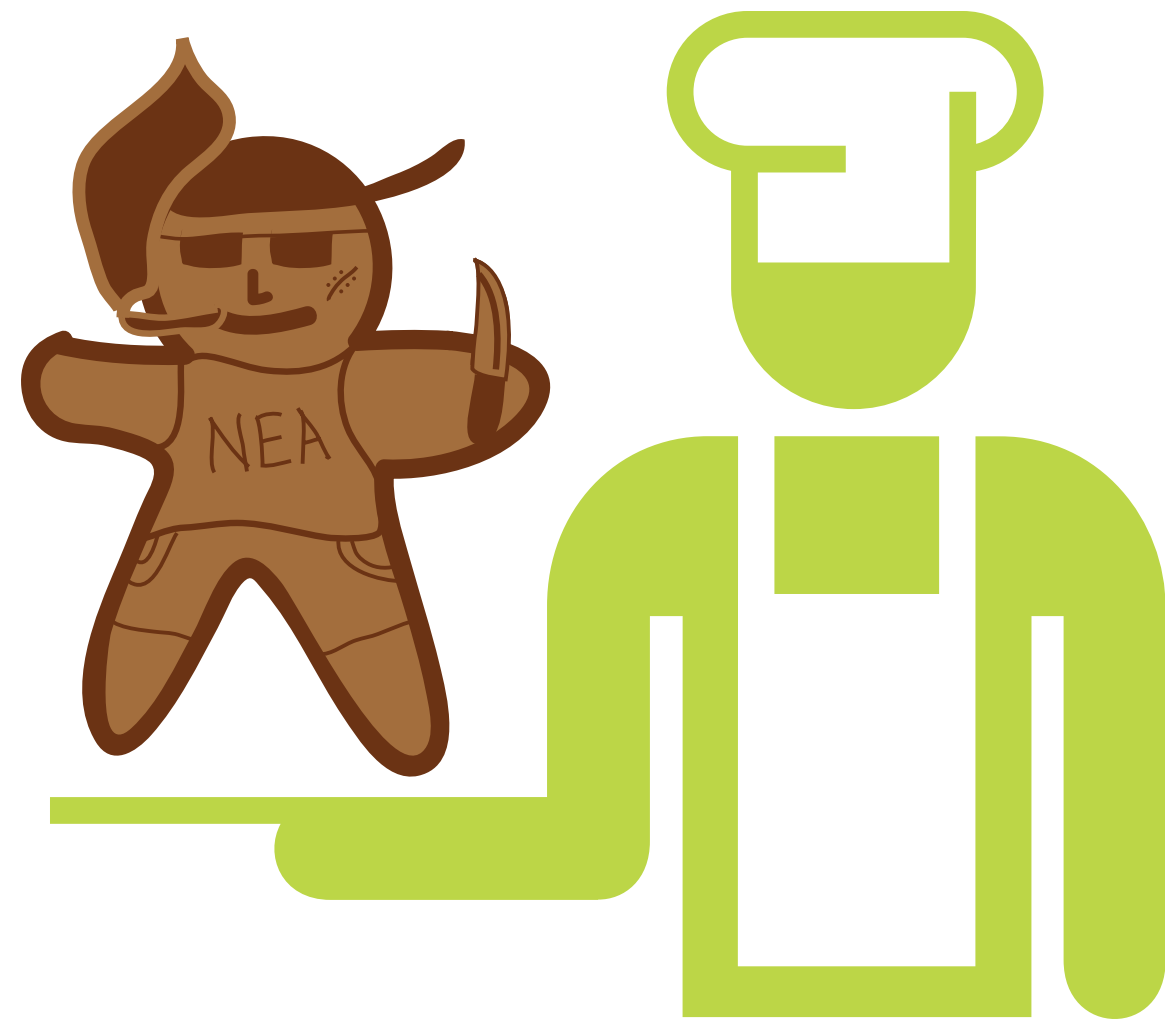
Valores por defecto

Todos los objetos, **tienen valores por defecto**, si los consultamos utilizando los getters de la clase, obtenemos resultados parecidos a los que muestra la imagen.

Para cambiar los valores usamos los setters. De esta forma se personaliza el objeto, como se mostro anteriormente con la galleta James Bond.

El valor por defecto, depende del tipo de dato.

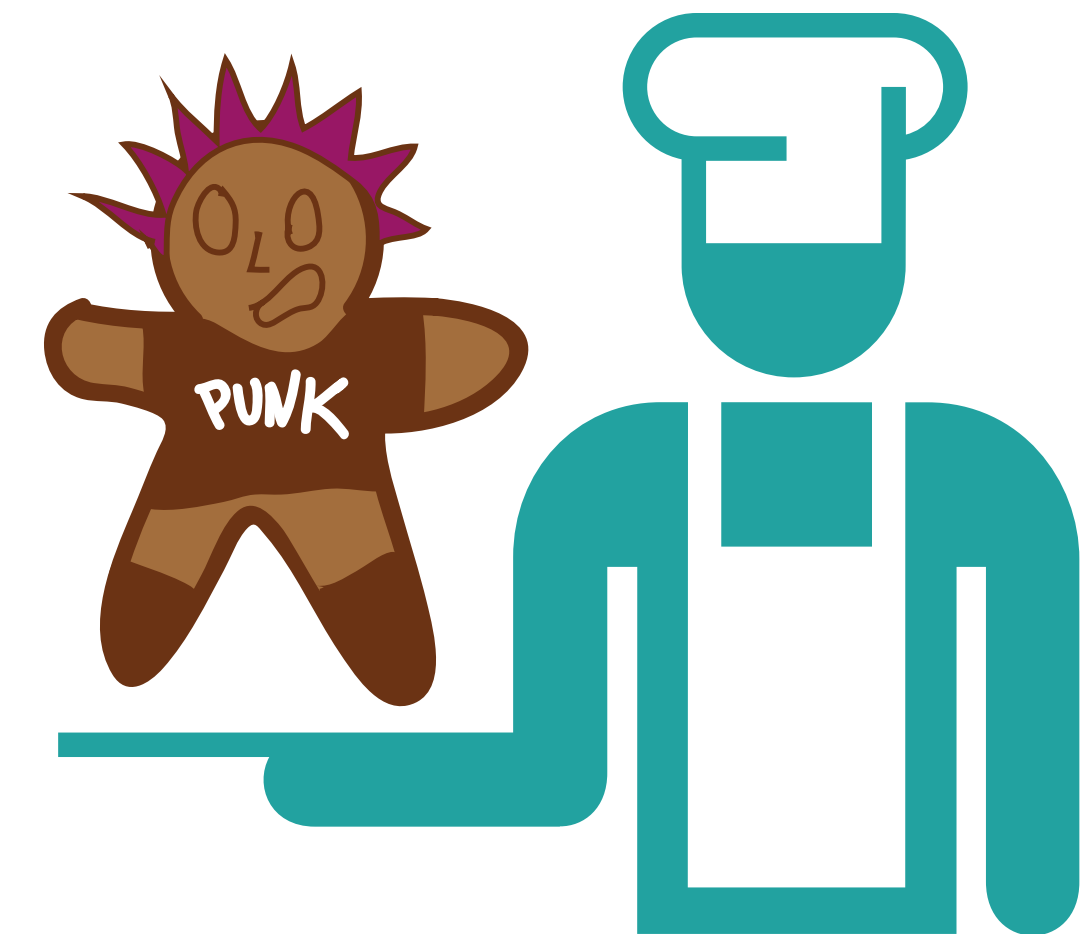




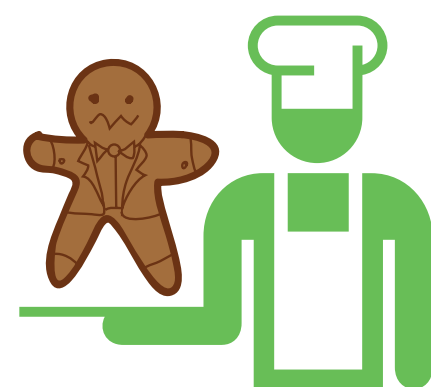
Frecuentemente se requiere inicializar un objeto con valores personalizados inmediatamente cuando se crea el objeto, en este caso se usan los constructores.

¿Qué es un constructor?

Los constructores también se pueden usar para asignar valores por defecto, diferentes a los tradicionales al objeto.



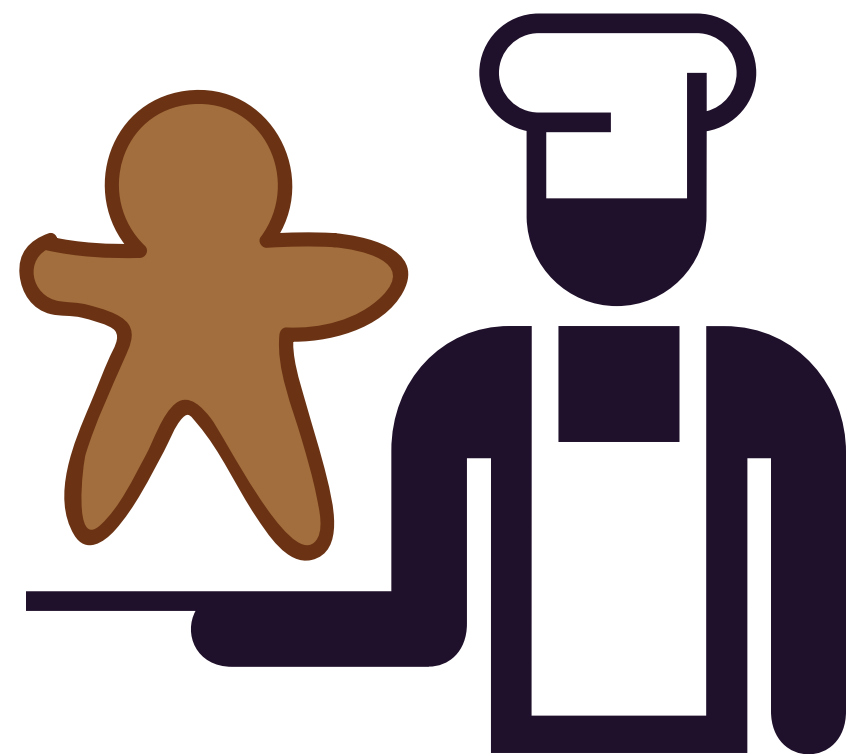
El constructor, sería el panadero, el cual personaliza los productos.



Siempre hay un constructor que crea objetos con valores por defecto.

Si se usa un constructor diferente al constructor por defecto, se debe incluir el constructor vacío, si se quiere seguir usando esta característica para crear instancias, sin pasar argumentos.

Constructor por defecto



Galleta g = new Galleta();

Constructor por defecto

```
Galleta(){  
    //Sentencias  
}
```

Firma de un constructor

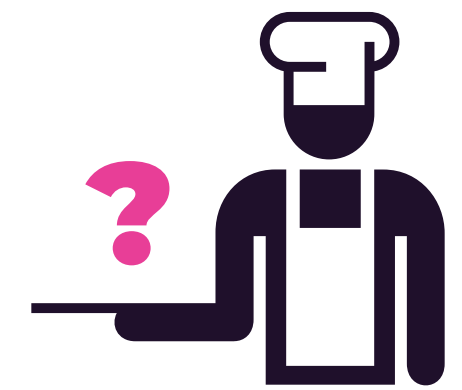
```
[Modificador] NombreClase(listaparametros){  
    //sentencias de inicialización de objeto  
}
```

Un constructor es similar a una función, la diferencia es que los constructores no tienen retorno, ya que su objetivo es inicializar o ejecutar funciones al momento del instanciamiento de la clase.

Constructores personalizados

Una clase puede tener, varios constructores, los cuales deben tener el mismo nombre pero con diferente lista de parámetros o tipo de dato en el caso en el que los constructores reciban el mismo número de valores, como muestra la imagen.

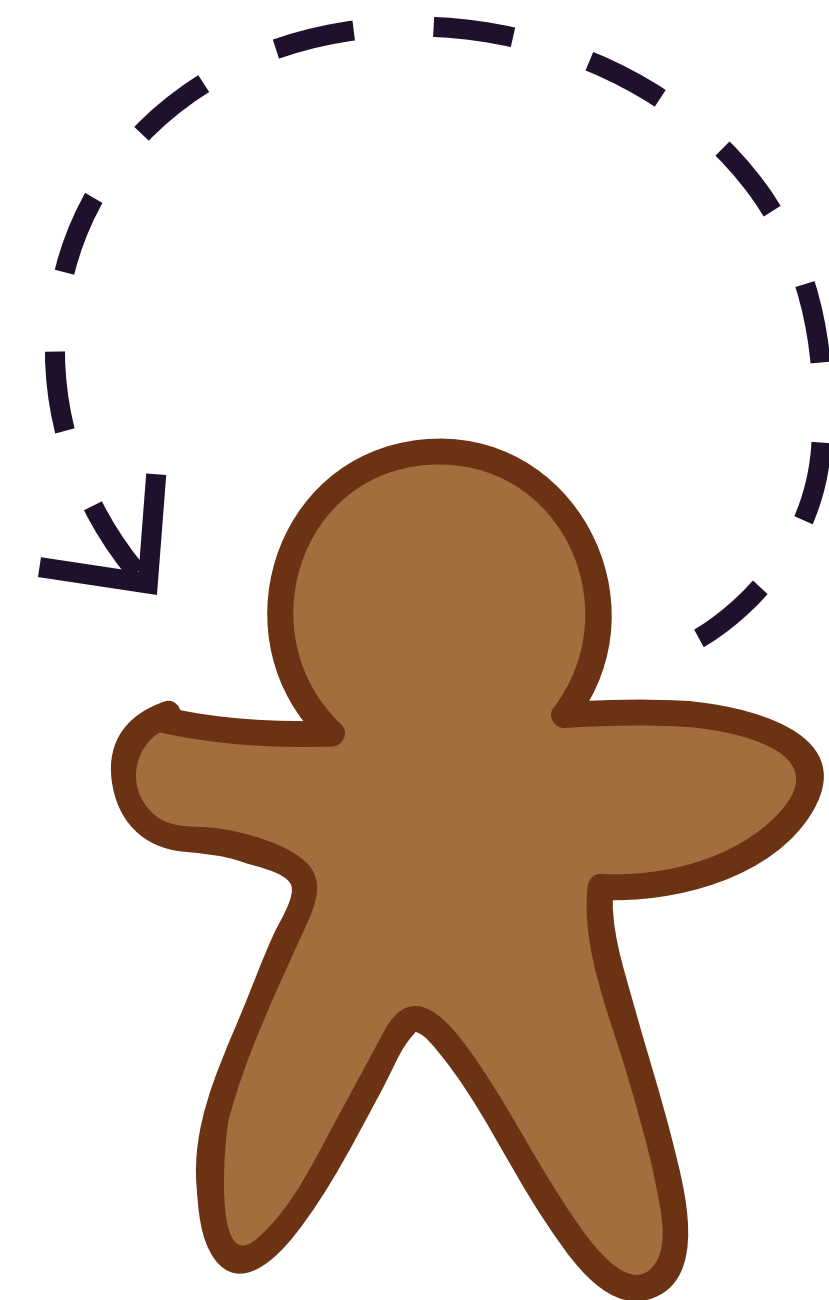
```
Galleta(){  
    Precio = 1000;  
}  
  
Galleta(float precio){  
    Precio = precio;  
}  
  
Galleta(string tipo, float peso){  
    Tipo = tipo;  
    Peso = peso;  
    if(tipo=="fortuna")  
    {  
        CrearFrase();  
    }  
}
```



Palabra reservada this

Es una referencia al mismo objeto, se usa para autoasignación de propiedades en los constructores o en cualquier contexto dentro de la clase. Evita además conflictos con nombres de valores que llegan a la clase.

```
this.Nombre = "John Doe";
```



Relaciones entre clases

Ejercicios

- ¿Cuál es el valor por defecto de los principales tipos de variable?
- Escriba la función a partir de su invocación
 - `float x = Comprar(3F,5.3,2)`
 - `Valorizar(3,true,'a')`
 - `double y = CalcularInteres()`
 - `DarLike()`
 - `uint z = CantidadLikesNuevos()`
- Dibuja la firma y cuerpo de una función e identifica sus partes
- Escriba una función que reciba 3 parametros, luego invoque la función con valores compatibles pero de diferente tipo.
- Realice una función que sume dos arreglos, tengo en cuenta las restricciones para realizar la operación correctamente.
- ¿Cuántos return puede tener una función?
- ¿Qué pasa si el valor a retornar no se inicializa antes de su retorno?
- ¿Qué diferencia hay entre argumentos y parámetros?
- ¿Cuales son las recomendaciones para escoger el nombre de una función?
- ¿Qué pasa si una función que retorne un valor este no se asigne?