



PROGRA_AMANDO

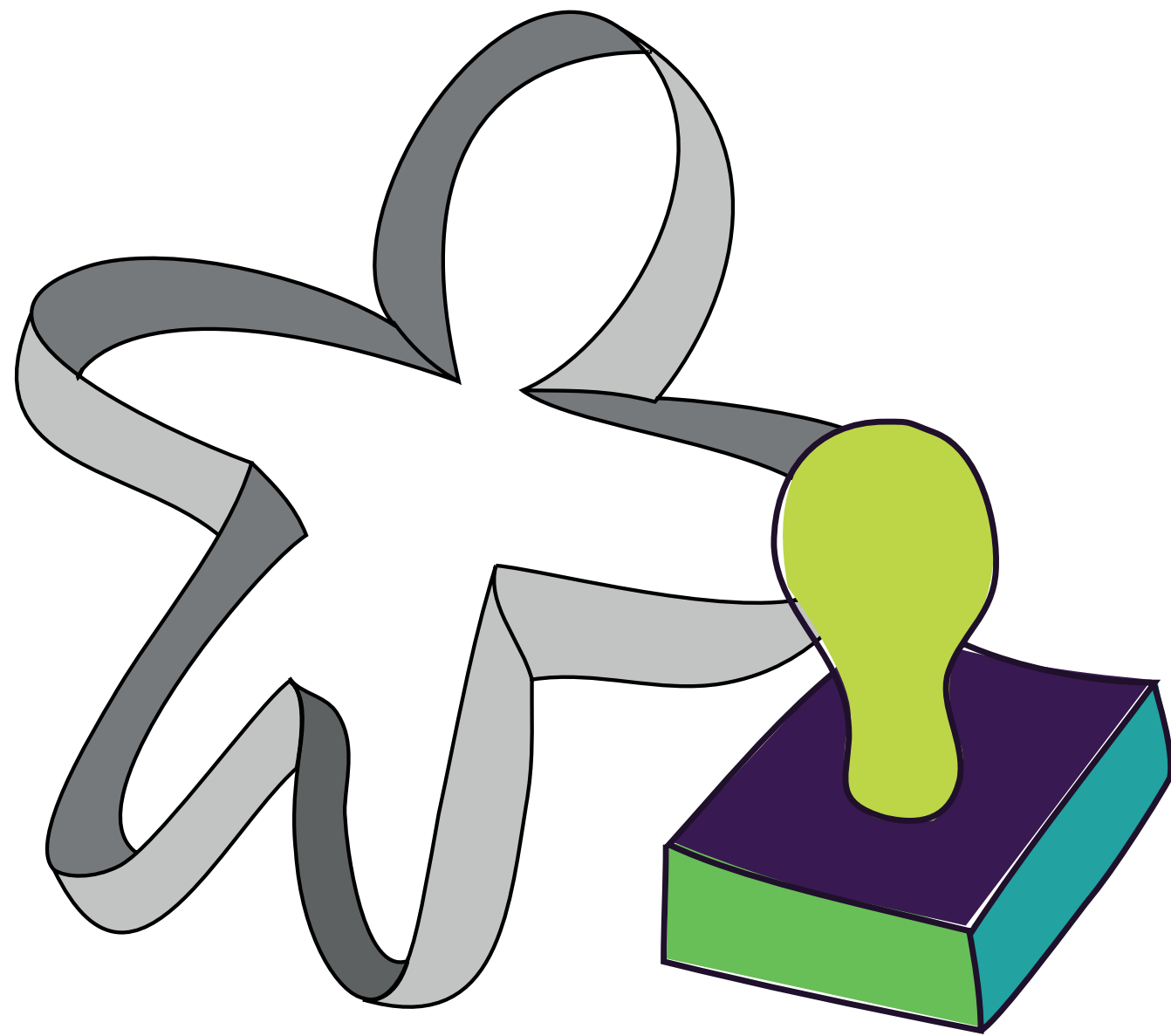
holamundo.co

Agenda

¿Qué es una clase?
¿Qué es un objeto?
Operaciones básicas
Encapsulamiento
Alta cohesión
Bajo acoplamiento
Ejercicios

¿Qué es una clase?

Es un molde o plantilla, que contiene la información necesaria para crear copias.



En una panadería las clases son los moldes de las galletas, en una caja de pagos la clase puede ser un sello que se pone a los recibos pagados, en una presentación de dispositivas, el molde es la plantilla con el diseño inicial, en una empresa de carros, el molde es el diseño del chasis, el stencil es el molde de un graffiti, en una impresión 3D el molde es el diseño de la figura.

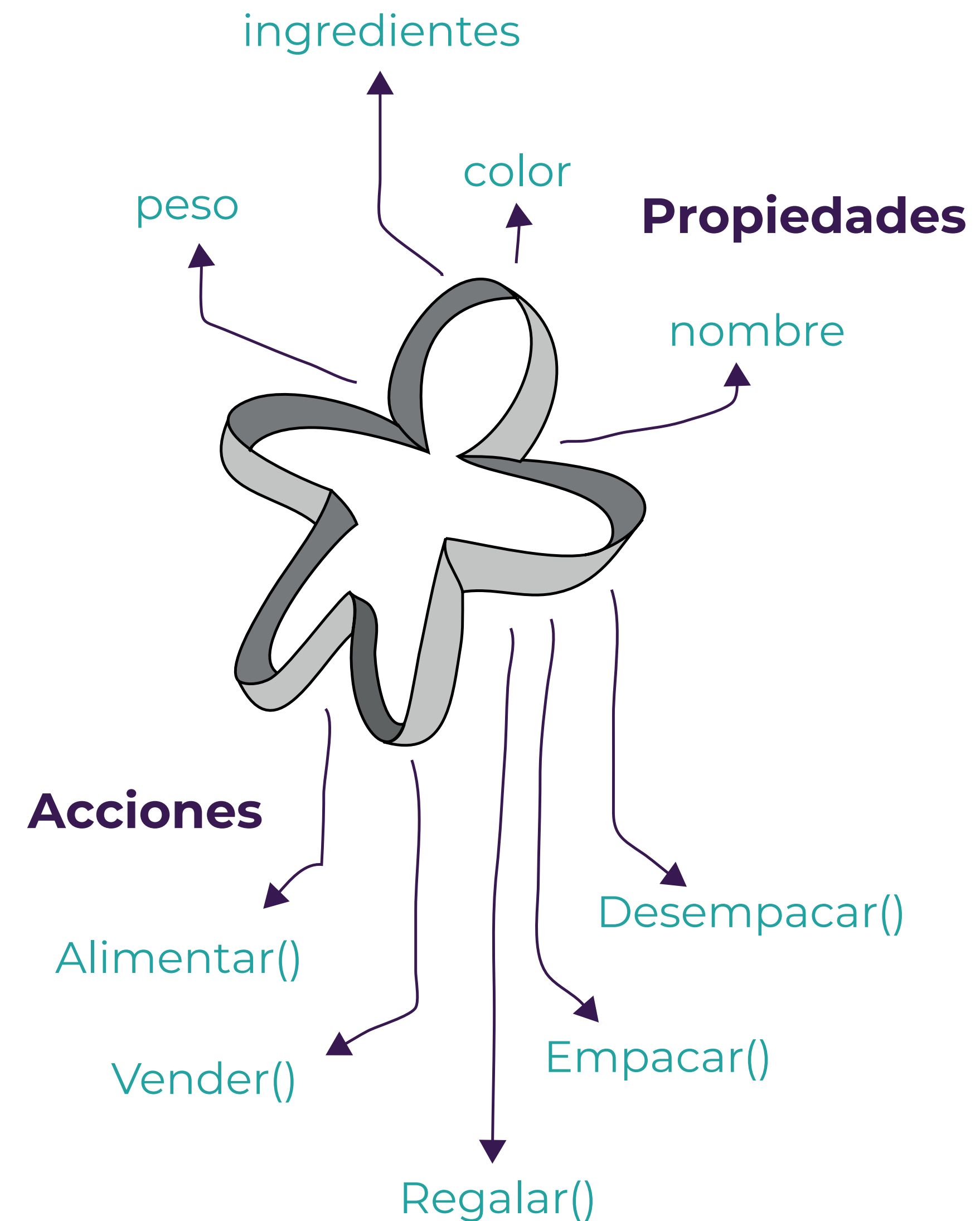
El concepto de clase se ve representado en muchas situaciones de la vida cotidiana, como ejercicio identifica otras situaciones donde se usen moldes.

¿Qué es una clase?

Una clase, es una **abstracción** de un concepto, que se desea representar en un programa de computador, por ejemplo un producto de una tienda online.

Para representar en el programa el concepto concreto, se recomienda realizar un análisis donde se identifiquen características principales y las acciones que este desempeña.

Se diseña la clase, usando un gráfico con la información específica de propiedades y acciones, llamada **diagrama de clase**, para luego realizar su implementación en código.



Escritura de una clase

```
public class Galleta
{
    private string propiedad;

    public void Accion()
    {
        Console.WriteLine();
    }

    public int Propiedad
    {
        get { return propiedad; }
        set { propiedad = value; }
    }
}
```

El nombre de la clase, por convención debe empezar con mayúscula y debe ser plural.

Se recomienda escribir todos los miembros dato al principio de la clase, esto facilita la ubicación en la clase, además los nombres de las propiedades deben escribirse en minúscula.

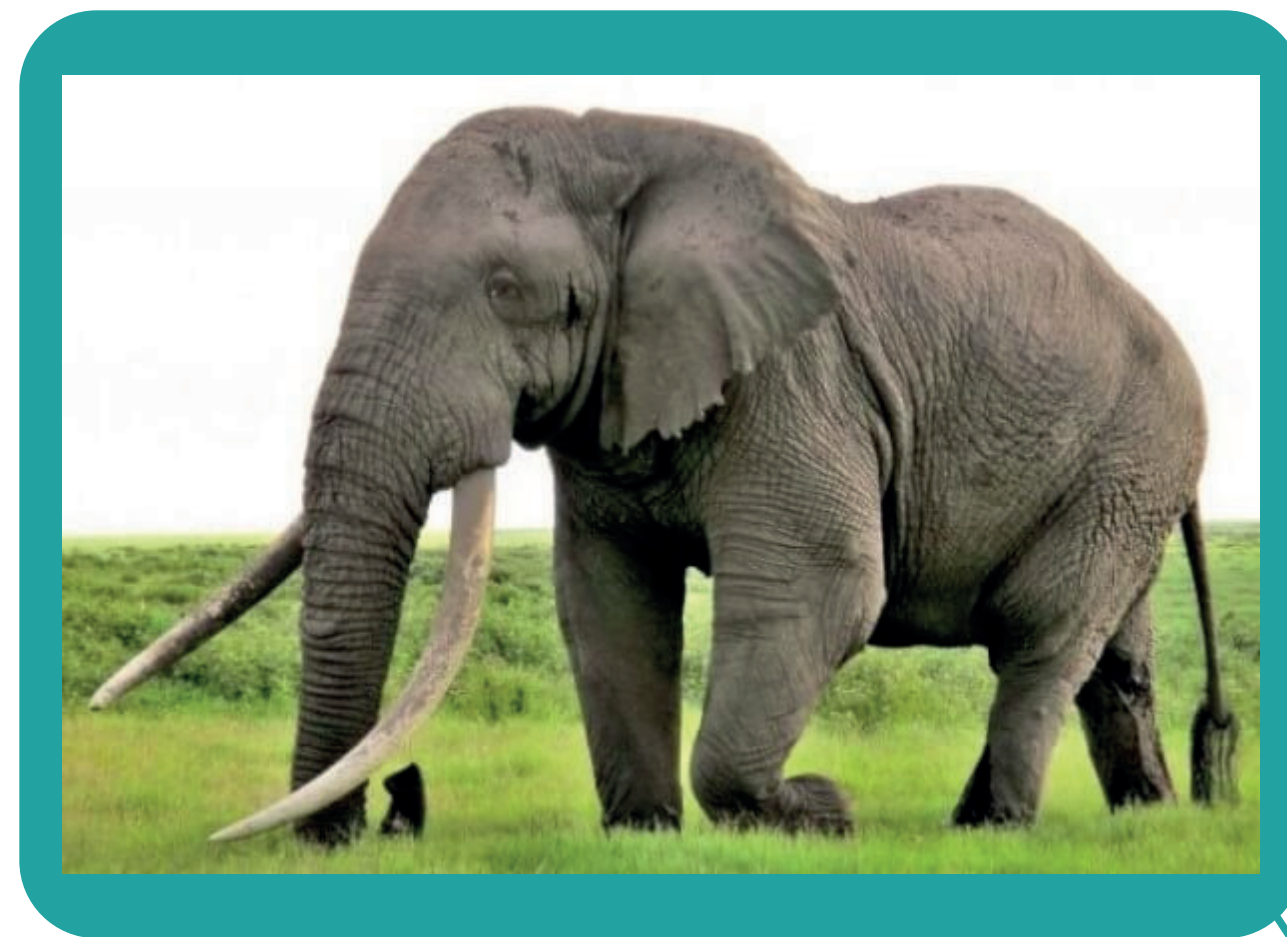
Las acciones son las funciones, es muy importante saber usar los diferentes tipos de funciones.

Los setters y getters permiten que el código sea escalable.

¿Qué es abstraer?

Empieza con la observación de un concepto de la vida real, por ejemplo un **Elefante**.

- ! Para hacer la abstracción se recomienda emplear como herramienta el dibujo.

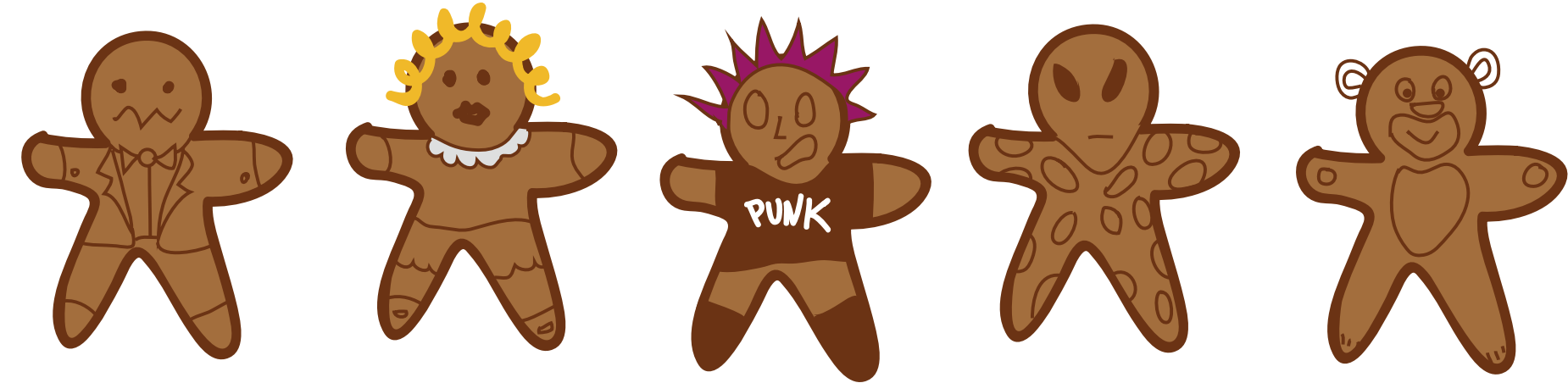


Simplificar la idea (imagen), donde queden los detalles mínimos y útiles para representar el término analizado.

Las abstracciones pueden ser diferentes, pero el concepto específico se mantiene.

Podemos ver varios dibujos del mismo animal, pero sin importar el estilo, el concepto reflejado debe ser el mismo.

¿Qué es un objeto?

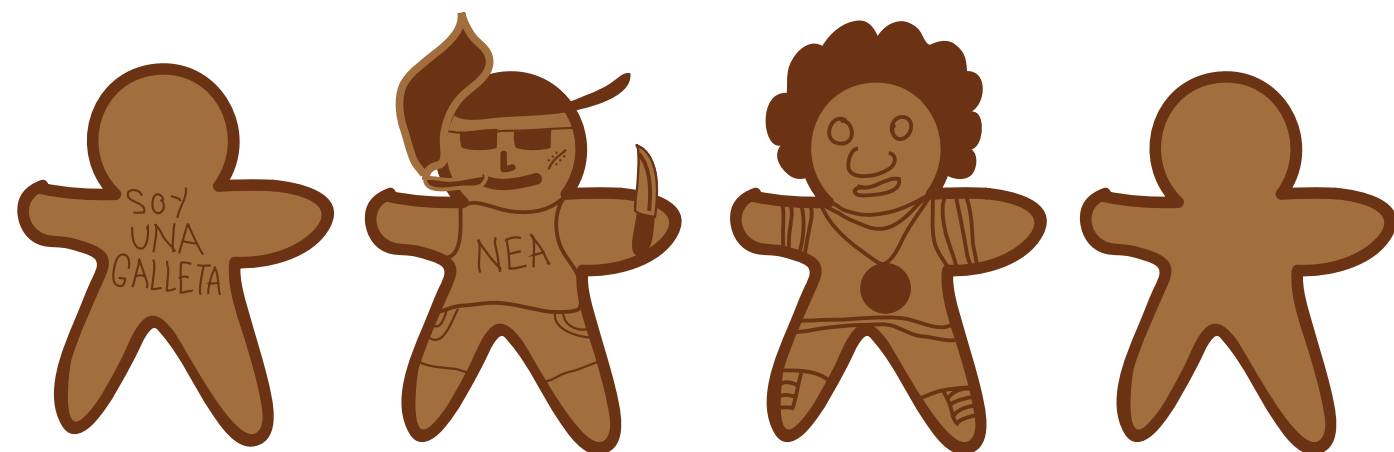


Son las **copias** que se crean a partir de una clase.

Las copias se crean con el molde, a este proceso se le llama instanciamiento de la clase, y permite crear objetos genéricos, también es posible luego personalizar el objeto creado.

Se puede decir que la clase define el ADN del objeto, por tanto todos los objetos creados con un molde son similares porque comparten propiedades y acciones, pero son únicos porque son independientes.

No hay copias iguales, aunque tengan similitudes entre sí.



¿Qué es un objeto?



El objeto tiene propiedades que lo personalizan, por ejemplo: número de ojos, decoración, forma de la boca, precio, peso, color.

Además tiene acciones por ejemplo, puede Alimentar()

Los objetos pueden cambiar sus propiedades en **tiempo de ejecución**, se podría agregar por ejemplo chocolate para cambiar el valor actual de la cobertura.



¿Qué es tiempo de ejecución?


```
public static void Main(string []args)
{
    Galleta galleta;

    galleta = new Galleta();

    galleta.Nombre = "Galleta Fortuna";
    galleta.Mensaje = "Hoy compila :)";

    Console.WriteLine(galleta.Mensaje);
}
```

Por fuera de la clase, es posible crear objetos o instancias de la clase Galleta, la cual es la plantilla o molde que permite crear los objetos.

!

Reconozca cada una de sus partes, identifique donde se usa el setter y donde se usa el getter.

¿Para que sirve usar la palabra new?

Operaciones básicas OOP

1

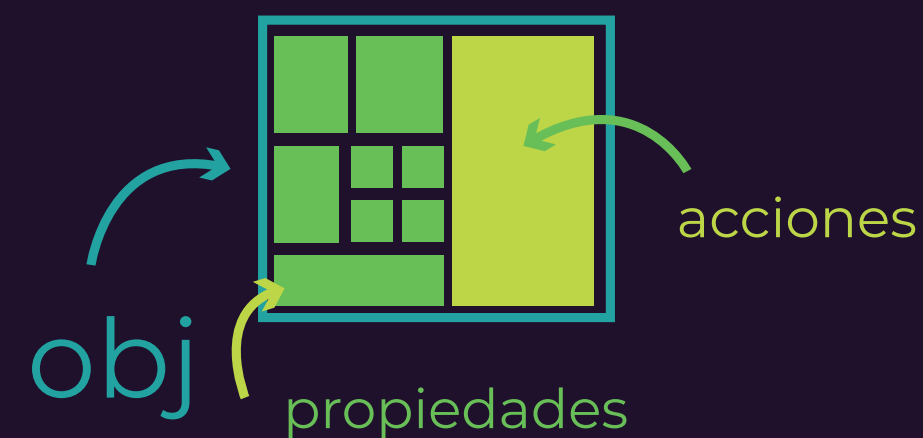
```
Clase obj;
```



Igual que con las variables primitivas (int, double, float, bool,...), el primer paso es declarar el objeto, se usa como tipo de dato el nombre de la clase seguido por el nombre de la variable que va a tener el espacio reservado para almacenar el objeto. En este paso todavía **NO hay copia**.

2

```
obj = new Clase();
```

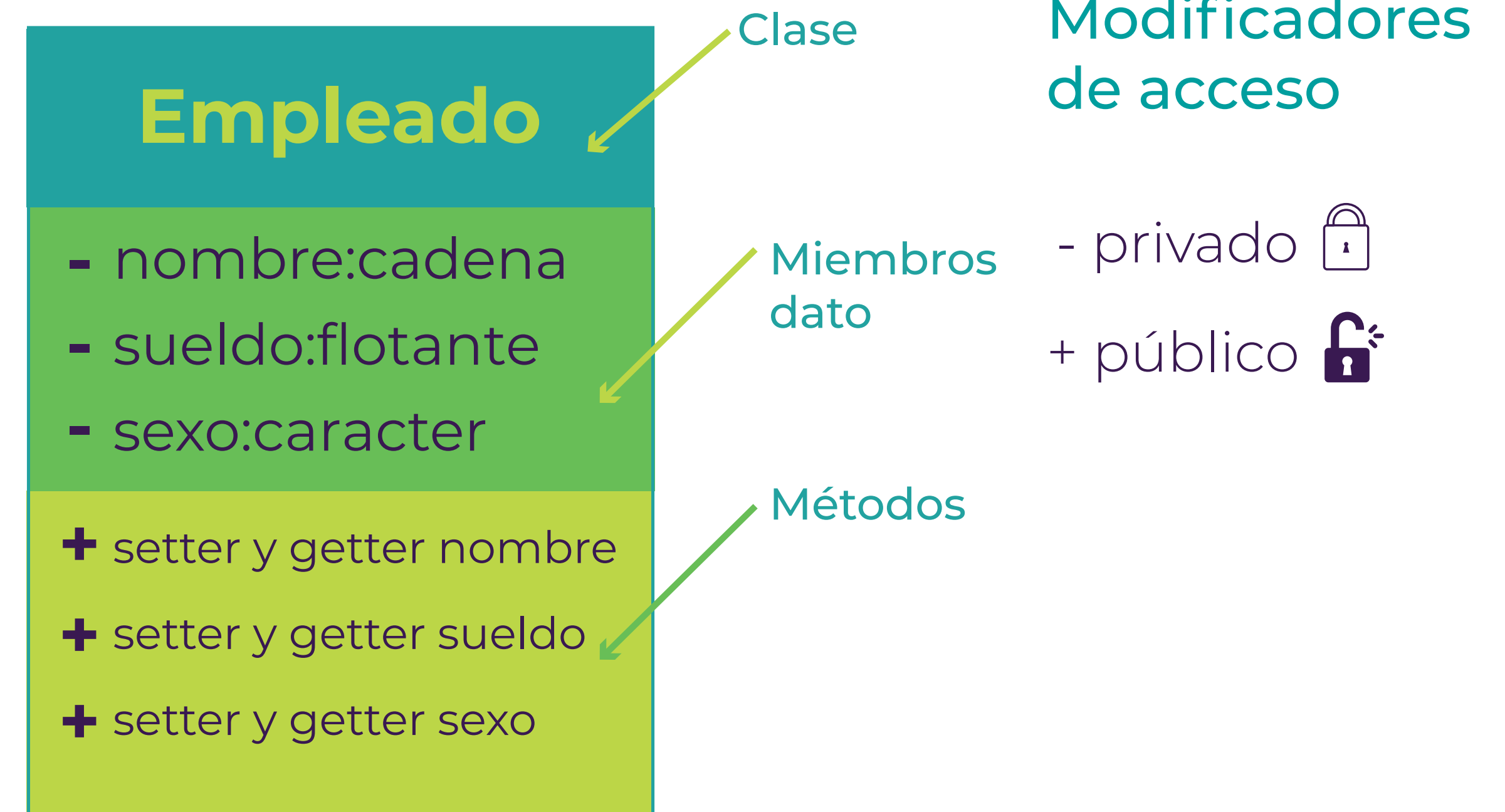


En este momento ya hay una copia en la memoria.

Se crea la instancia del objeto (se inicializa), usando la palabra reservada **new**.

Encapsulamiento

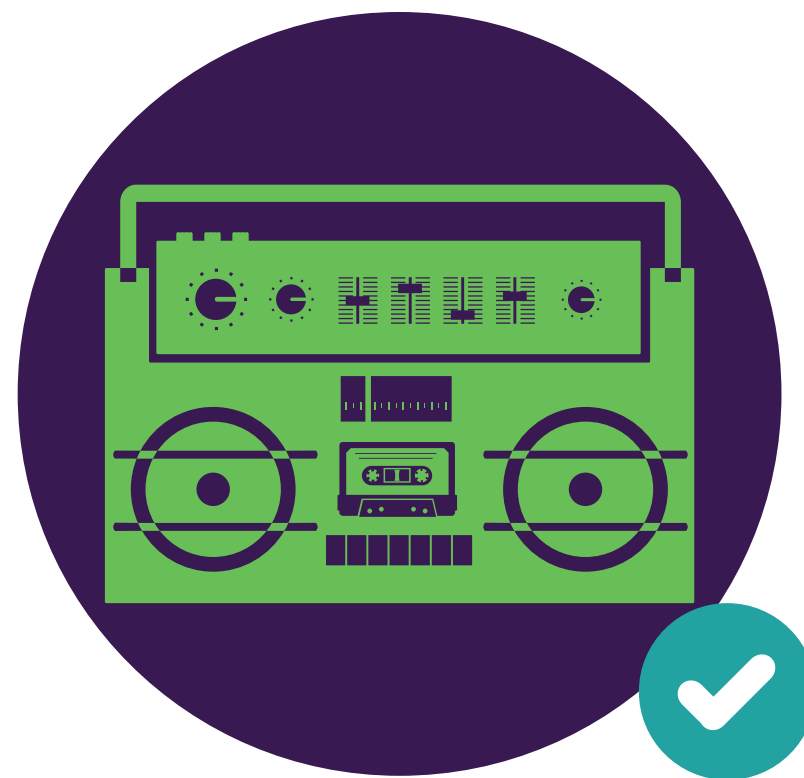
Se refiere a una característica del diseño de la clase, se oculta el acceso a las propiedades, para evitar que puedan ser accedidas o modificadas desde fuera.



Sin encapsulamiento hay problemas de escalabilidad, porque se crean dependencias en el código difíciles de modificar.

Alta cohesión

Es una característica que debe tener una clase, se refiere a hacer una sola cosa bien, busca especializar la clase.



La navaja, realiza muy bien su labor, también el radio, pero no funcionarían igual, si fueran un solo elemento que tuviera las dos funciones.



Mientras diseña una clase, se da cuenta que, ésta realiza múltiples funciones, es suficiente razón para dividir la funcionalidad en varias clases especializadas.

Bajo acoplamiento

Para lograr el encapsulamiento en una clase, se deben declarar los miembros datos o propiedades de la clase, con el modificador de acceso `private`, y solo permitir modificar o consultar el valor de cada miembro dato, por medio de setter y getters, es decir métodos `públicos`.

 `private string dato;`

Los miembros dato deben ser privados, salvo algunas excepciones que se estudiarán más adelante en otros temas.

Se asignan el setter y el getter de la propiedad encapsulada. No es necesario crear el setter y getter para todas las propiedades encapsuladas, esto depende del diseño de la clase.

 `public string Dato { get; set; }`

Ejercicios

- Realice la abstracción de los siguientes conceptos: cuenta, receta, pantalon, hamburguesa, regalo
- Realice una clase con alto acoplamiento y baja cohesión
- Diseñe la clase prestamo, cree 3 instancias, use sus setters y getters para modificar sus propiedades
- Escriba 3 ejemplos de la vida cotidiana donde no se aplique la alta cohesión, por ejemplo coche de bebe patineta, o una corbata sombrilla.
- Realice un ejemplo y explique porque es un error No encapsular los miembros dato.
- ¿Cuanto es el número mínimo y máximo de miembros datos de una clase?
- Diseñe una clase con 10 miembros datos de diferente tipo
- ¿Qué es instanciar e inicializar?
- ¿Para que sirven las palabras reservadas new, null, get, set,void, private y public?
- ¿Cómo se valida si dos objetos son iguales?