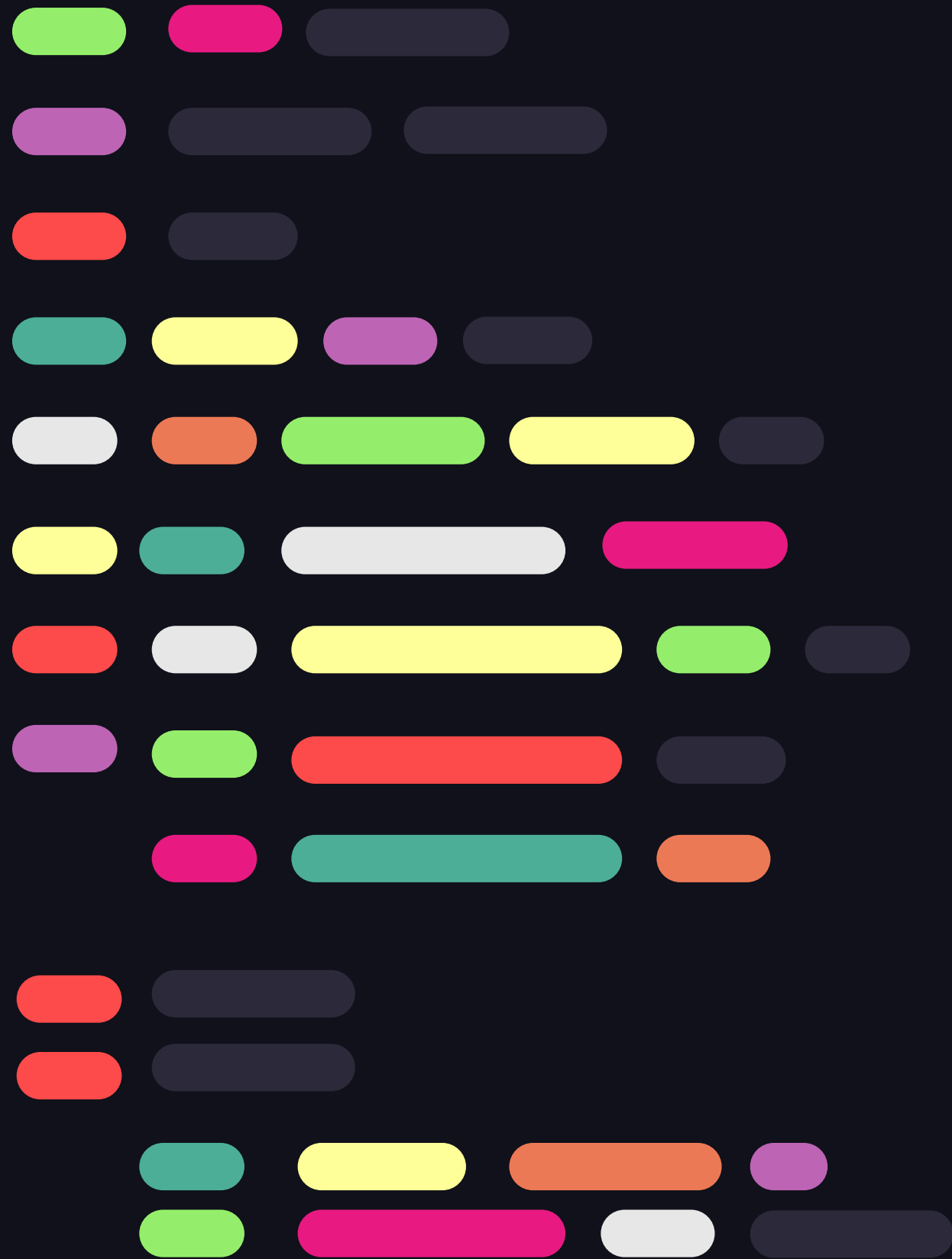


Paradigmas de Progamacion y Fundamento de la orientacion de obj.

< TEACHER: Oscar Benito Pacheco >





Integrantes

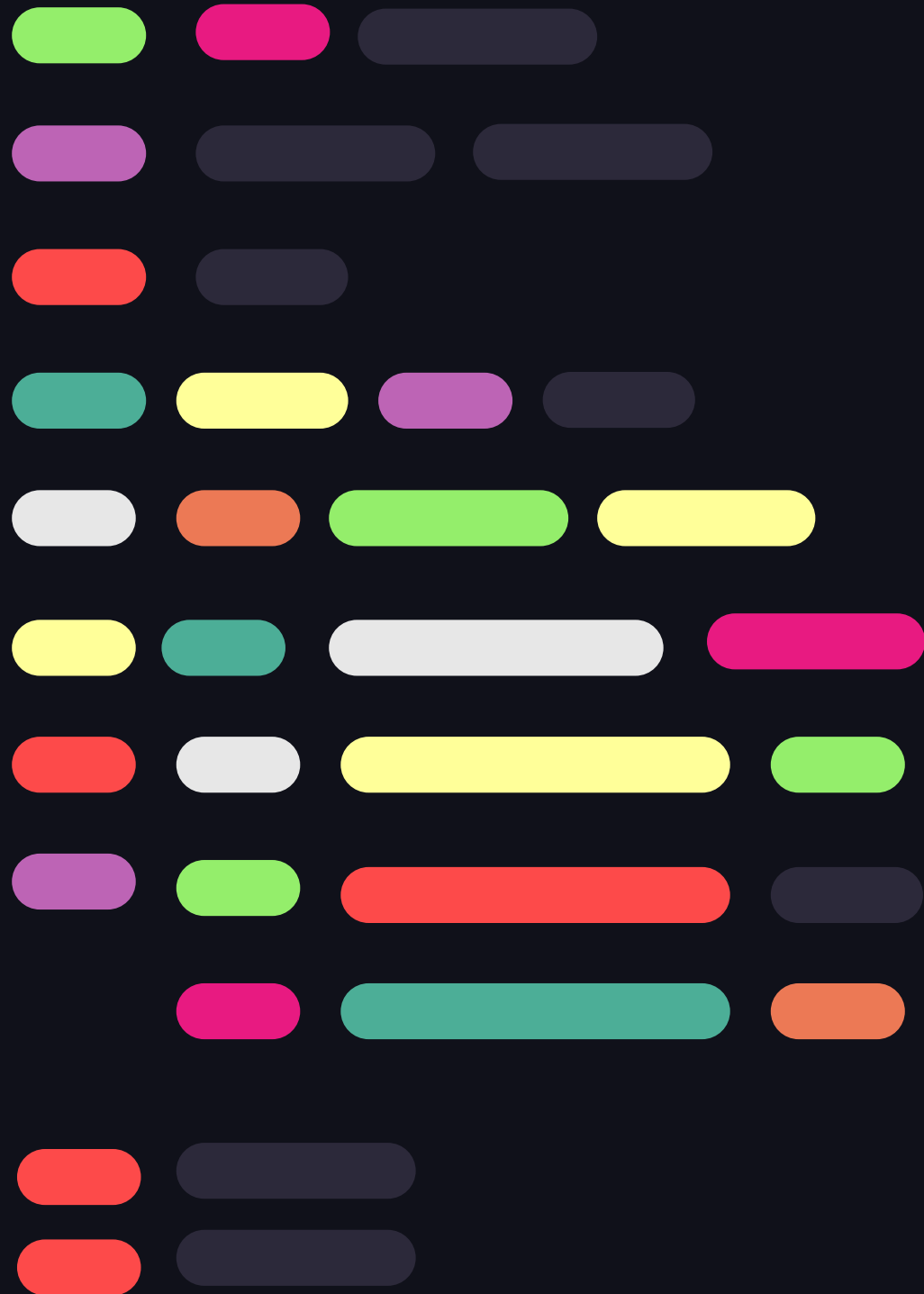
- < Palacios Carrasco, Roberto >
- < Avalos Palomino, Ismael >
- < Zapata Alania Carolina >



Paradigmas de programación!

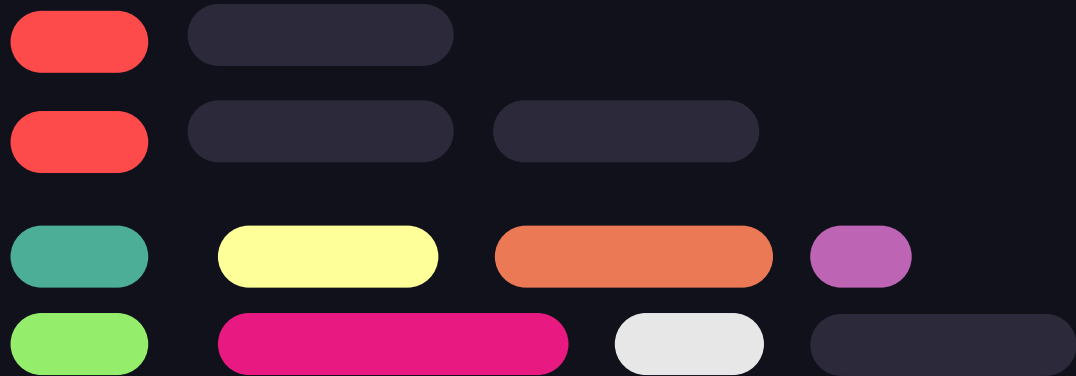
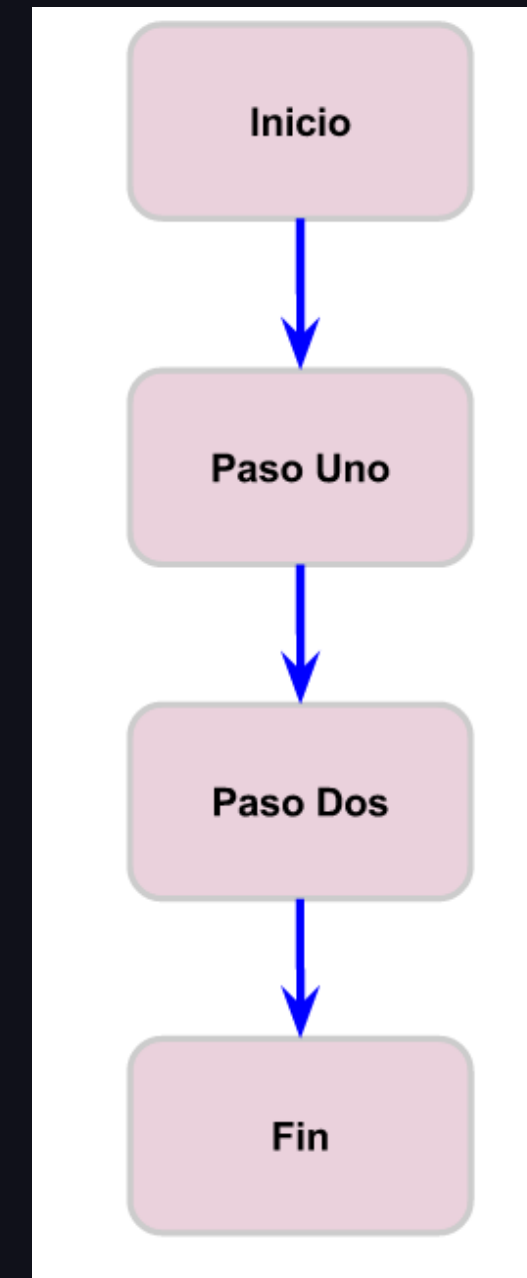
Es un enfoque o estilo utilizado para desarrollar software. Define métodos sistemáticos para diseñar programas que resuelvan problemas computacionales. Los lenguajes de programación pueden seguir uno o varios paradigmas según las instrucciones que permiten implementar, como es el caso de Python o JavaScript, que son lenguajes multiparadigmas.

- Procedural
- Estructurada
- Orientado a objetos



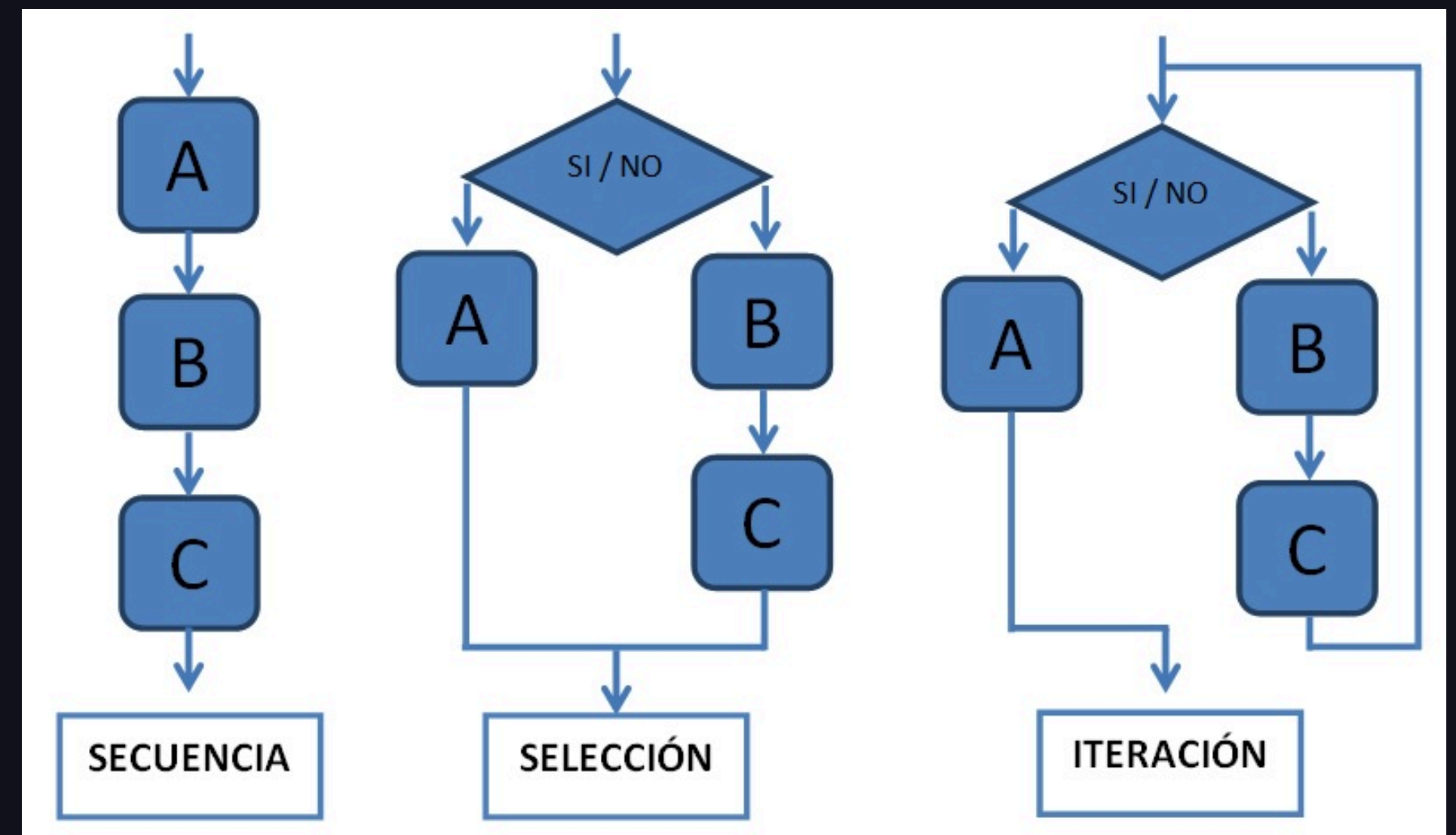
Procedural

Se basa en la secuencia de instrucciones para resolver un problema. El programa se divide en funciones o procedimientos, y el control del flujo lo determina el orden de ejecución. Es como seguir una receta paso a paso.



Estructurada

Es una rama del paradigma procedural, pero se enfoca en el uso de estructuras de control como if, while, for, para mejorar la claridad y organización del código. Promueve código más legible, mantenible y modular.





{ Orientado a objetos

Este enfoque propone una manera diferente de construir soluciones computacionales, comparado con otros paradigmas. En el lenguaje Simula, pionero en este paradigma, los objetos se crean a partir de clases y se manejan mediante referencias. Cuando ya no hay referencias hacia un objeto, un recolector de basura se encarga de liberarlo de la memoria. En Simula, los objetos pueden ser:

- **Activos** (cuando están ejecutando su bloque de instrucciones)
- **Inactivos** (cuando ya lo han terminado).

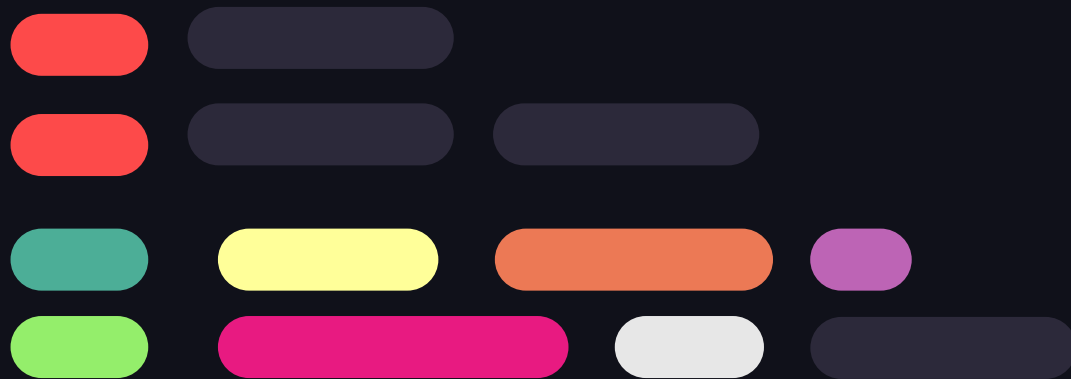
Aun así, se pueden utilizar sus atributos y procedimientos en cualquier momento.

```
! todo programa empieza con un begin y termina con un end ;
Begin

  Class Saludos;
  Begin
    OutText("¡Hola Mundo!");
    OutImage;
  End of class saludos;

  REF(Saludos) objeto;
  objeto :- New Saludos;

End of module program;
```



¿Qué es la Programación Orientada a Objetos (POO)?

La Programación Orientada a Objetos (en inglés, Object-Oriented Programming, OOP) es un paradigma de programación que organiza el software en objetos que interactúan entre sí. Estos objetos representan cosas del mundo real o conceptos abstractos, y combinan datos y comportamientos en una sola unidad.

Ejemplo: Un "auto" es un objeto. Tiene propiedades como color, modelo y marca, y tiene comportamientos como avanzar, frenar o girar.



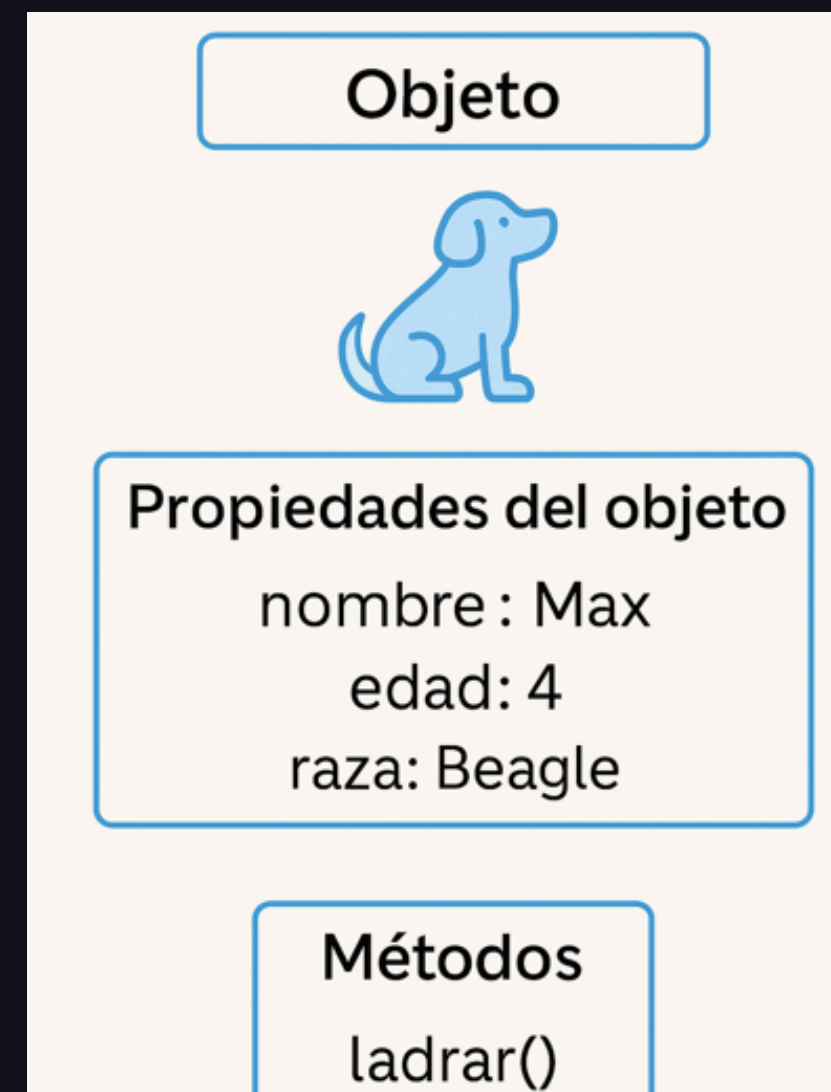
¿Qué es un Objeto?

Un objeto es la unidad básica de la programación orientada a objetos. Se trata de una instancia de una clase.

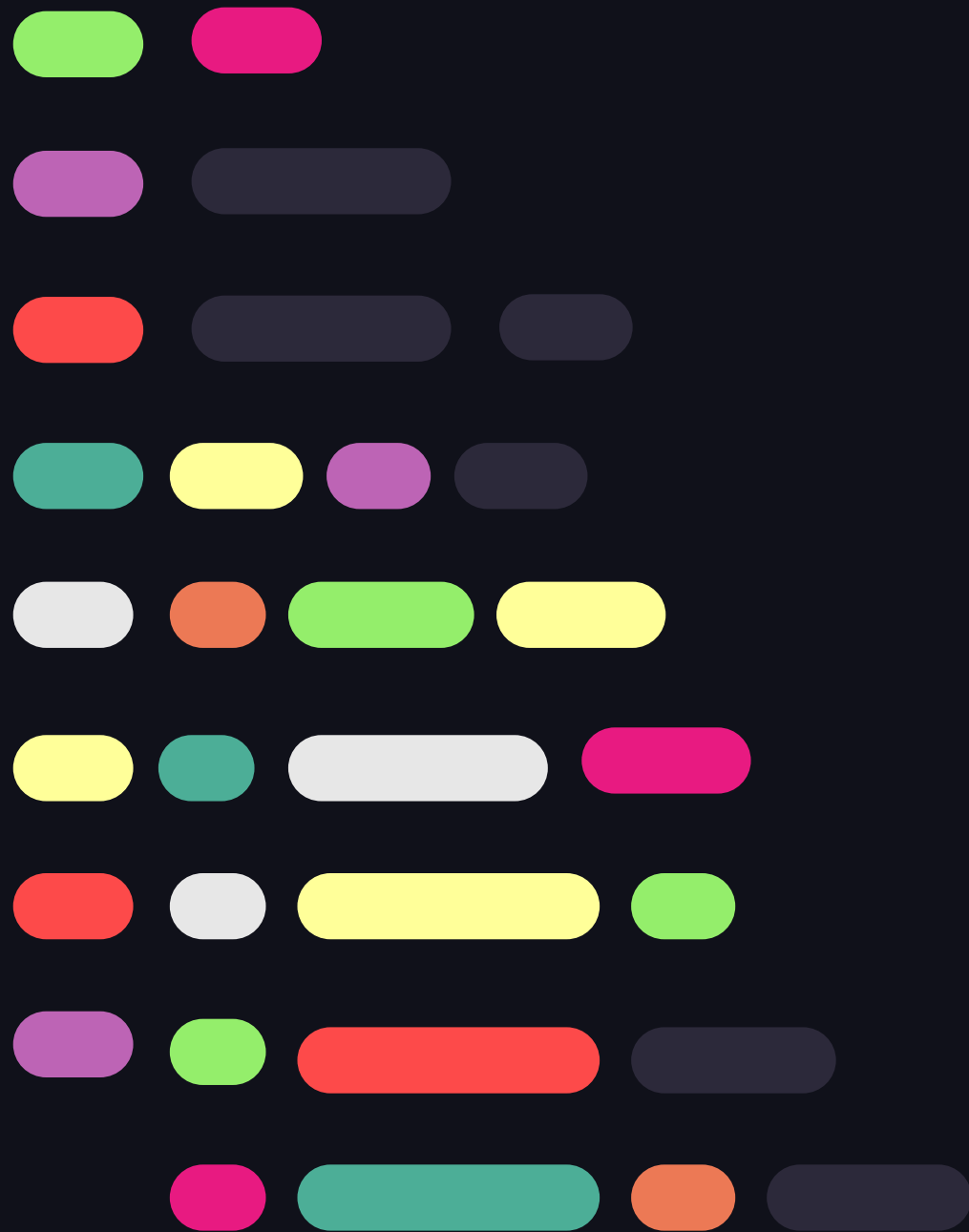


Un objeto tiene:

- Identidad: Cada objeto es único.
- Estado: Representado por las propiedades (atributos).
- Comportamiento: Representado por los métodos (funciones que puede realizar).



{ Clases



Una clase es una estructura que se utiliza para definir un tipo de objeto. Es como un molde o plano que describe cómo serán los objetos: qué datos tendrán y qué podrán hacer.

Atributos (o propiedades):

- Son las características del objeto.
- Se definen dentro de la clase y cambian según cada objeto.
- Ejemplo: nombre, edad, color, tamaño.

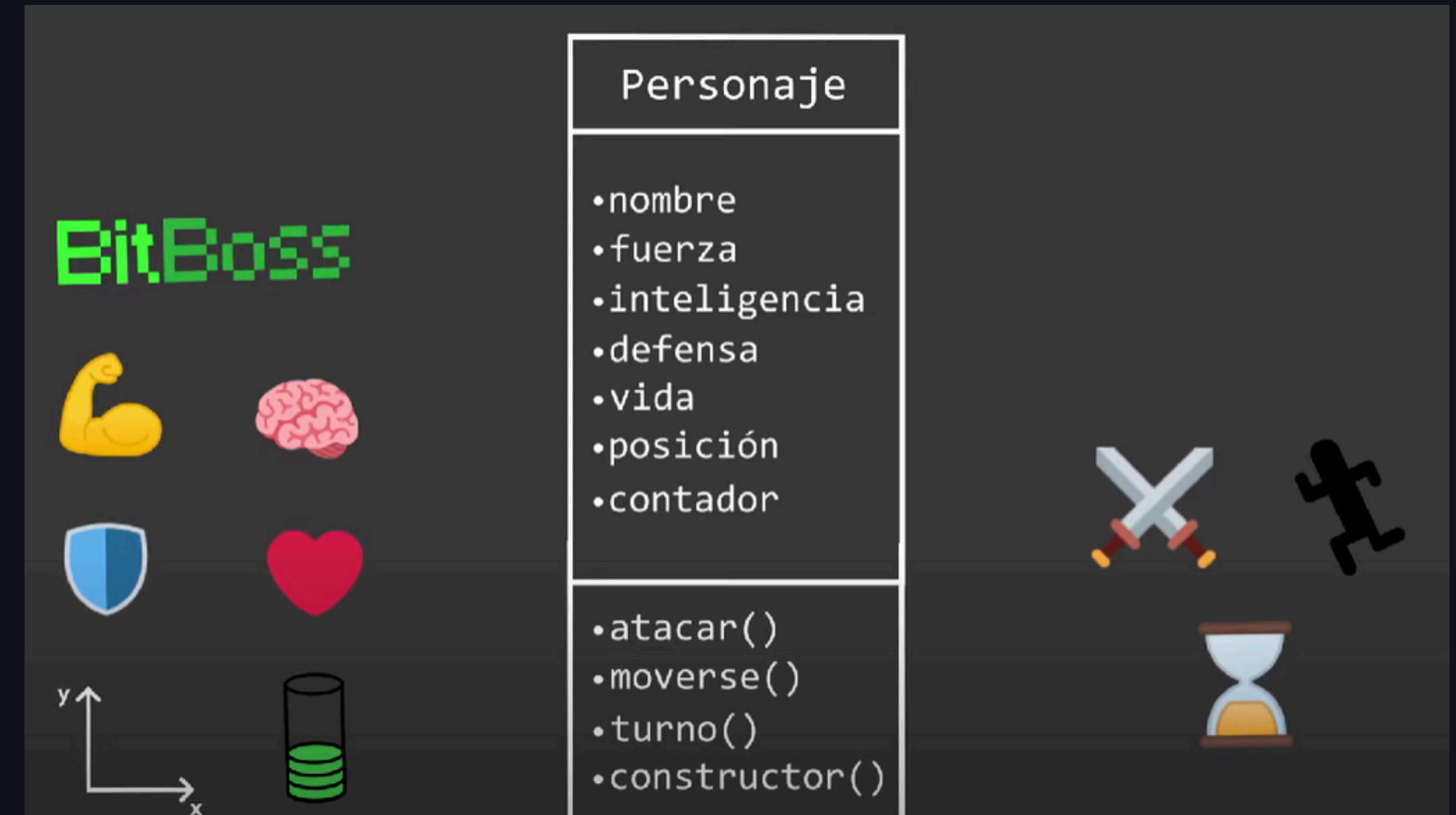
Métodos (o funciones de la clase):

- Son las acciones que el objeto puede hacer.
- Ejemplo: correr, hablar, saltar, guardar datos, calcular.



Abstracción

- Oculta los detalles internos paara mostrar solo lo necesario.



Ejemplo:

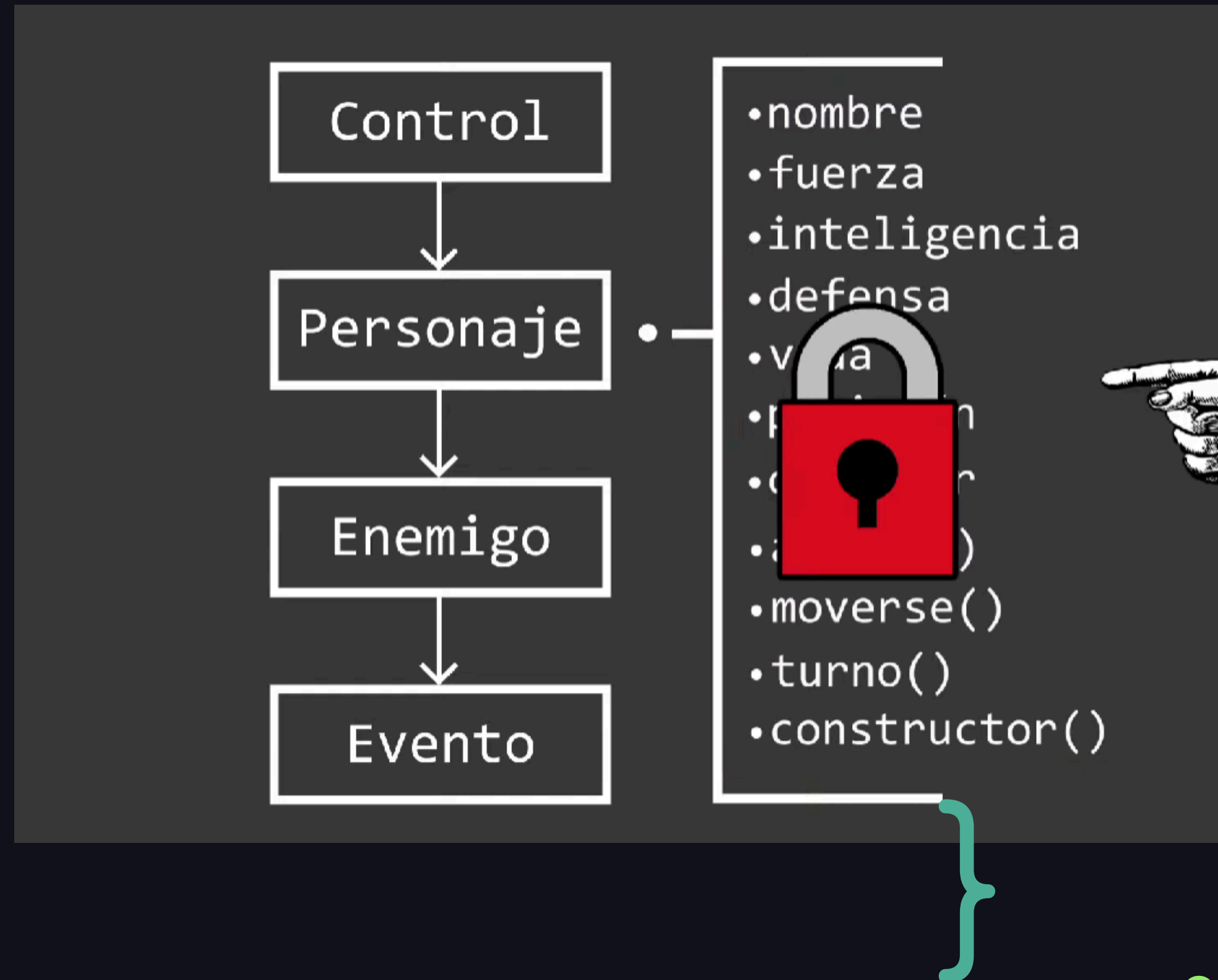
- La clase Auto permite encender el vehículo sin necesidad de saber cómo funciona el motor internamente.



Encapsulamiento

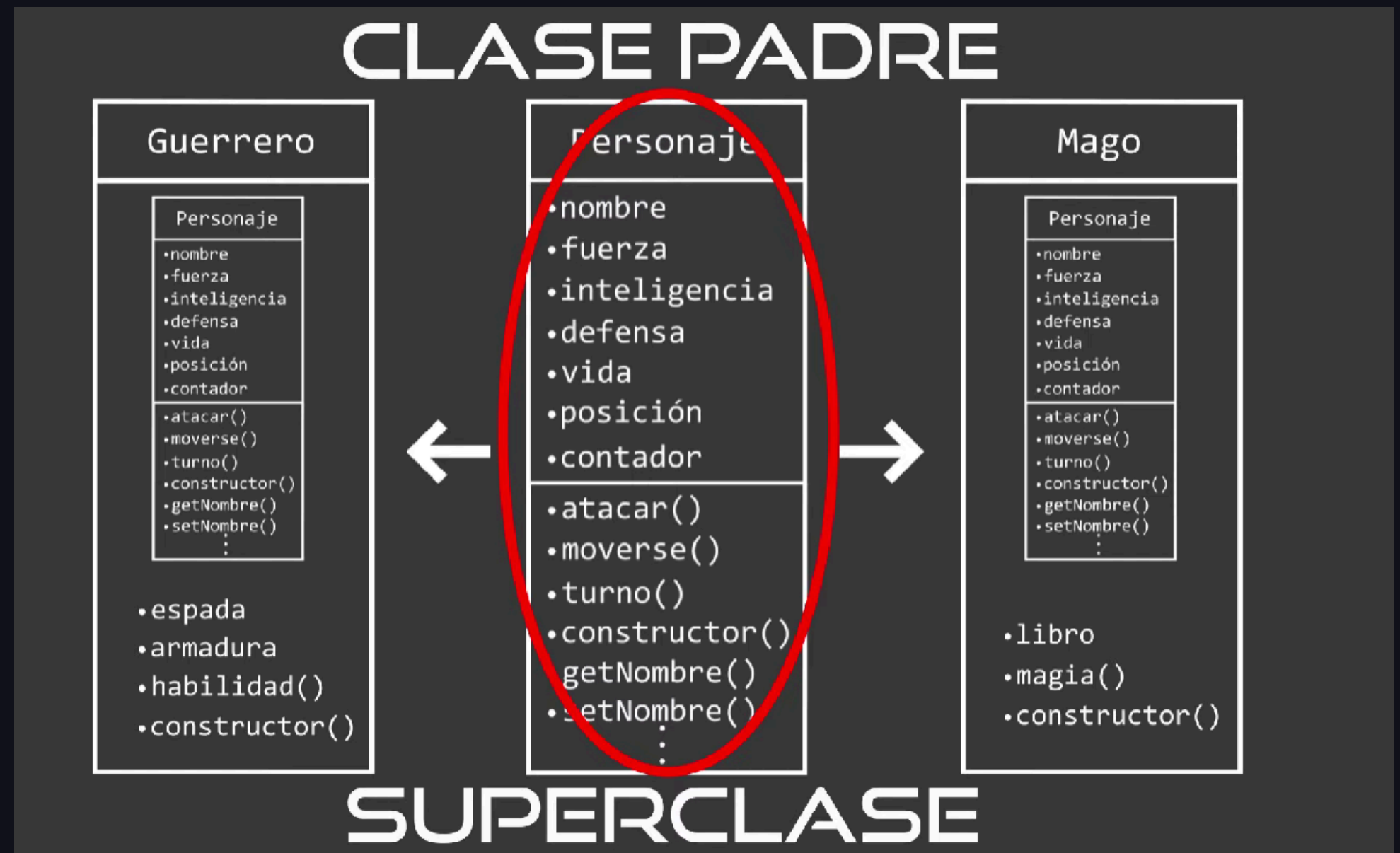
Ocultar los datos internos de un objeto y controlar su acceso.

Uso de modificadores de acceso (public, private, protected).



Herencia

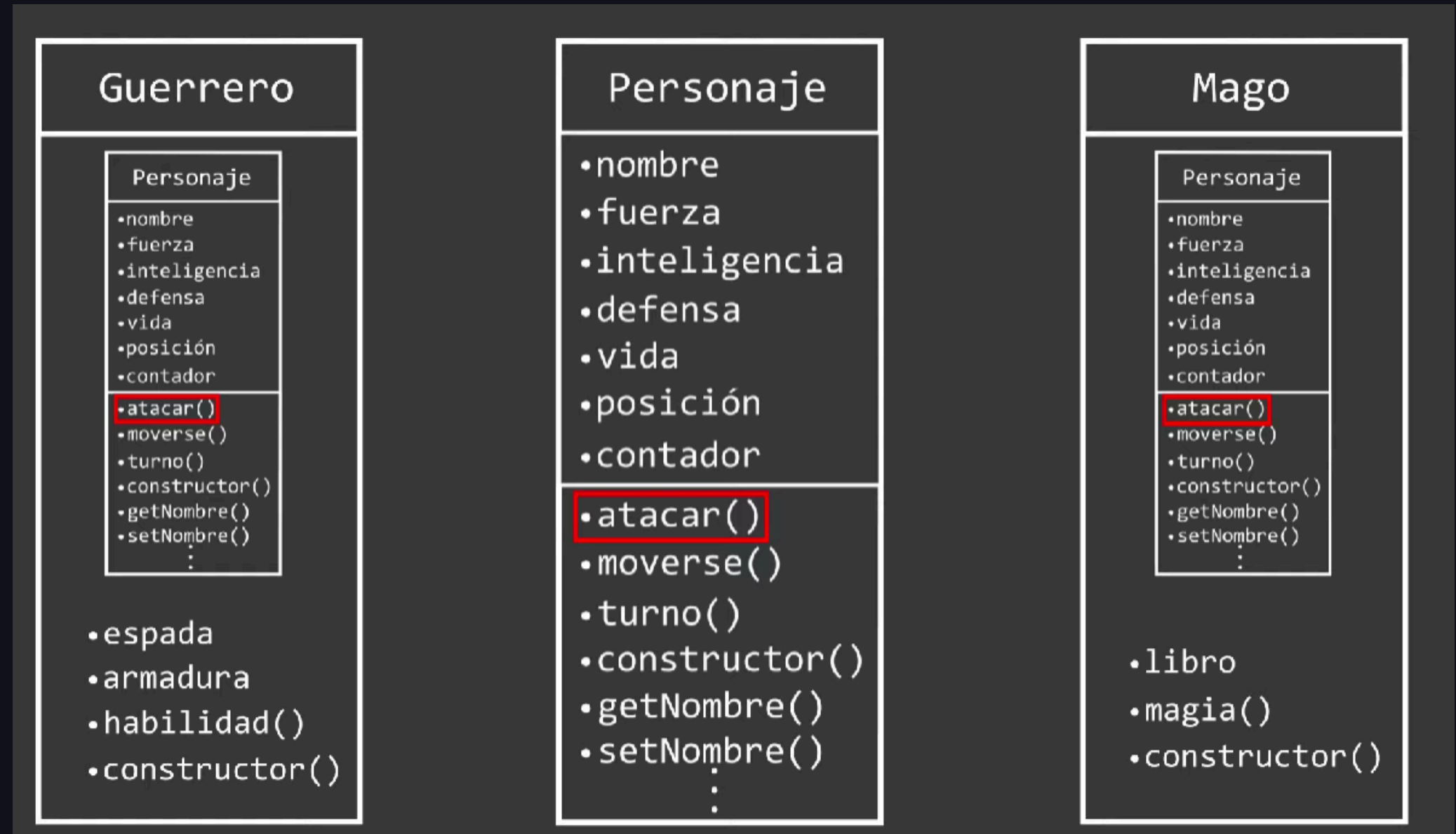
Permite que una clase (hija) herede atributos y métodos de otra (padre). Favorece la reutilización de código.

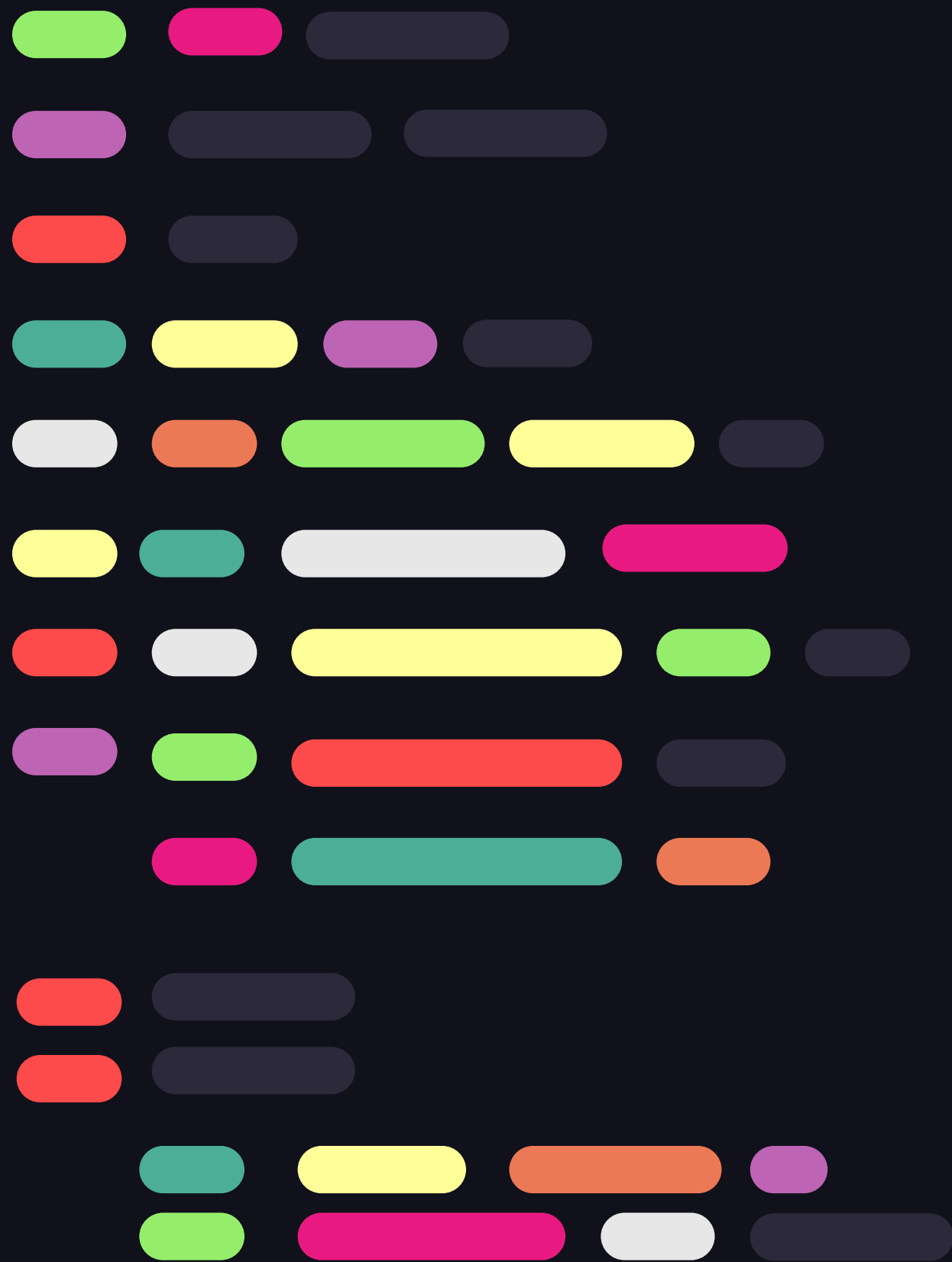


Polimorfismo



Capacidad de un método de comportarse de distintas maneras según el contexto.





Gracias

