
P00 - Conceptos

— Programación y Laboratorio III —

Agenda

- **Objeto**
- Clase
- Instancia
- Comunicación entre objetos
- Modificadores de acceso



Objeto

- Definición:

Es una entidad autónoma que contiene atributos y comportamiento.

- Se combinan datos y la lógica de programación.
- Tienen **estado** y **comportamiento**.
 - Estado: sustantivos
 - Comportamiento: verbos



Objeto: ejemplo

Objeto Persona

Atributos:

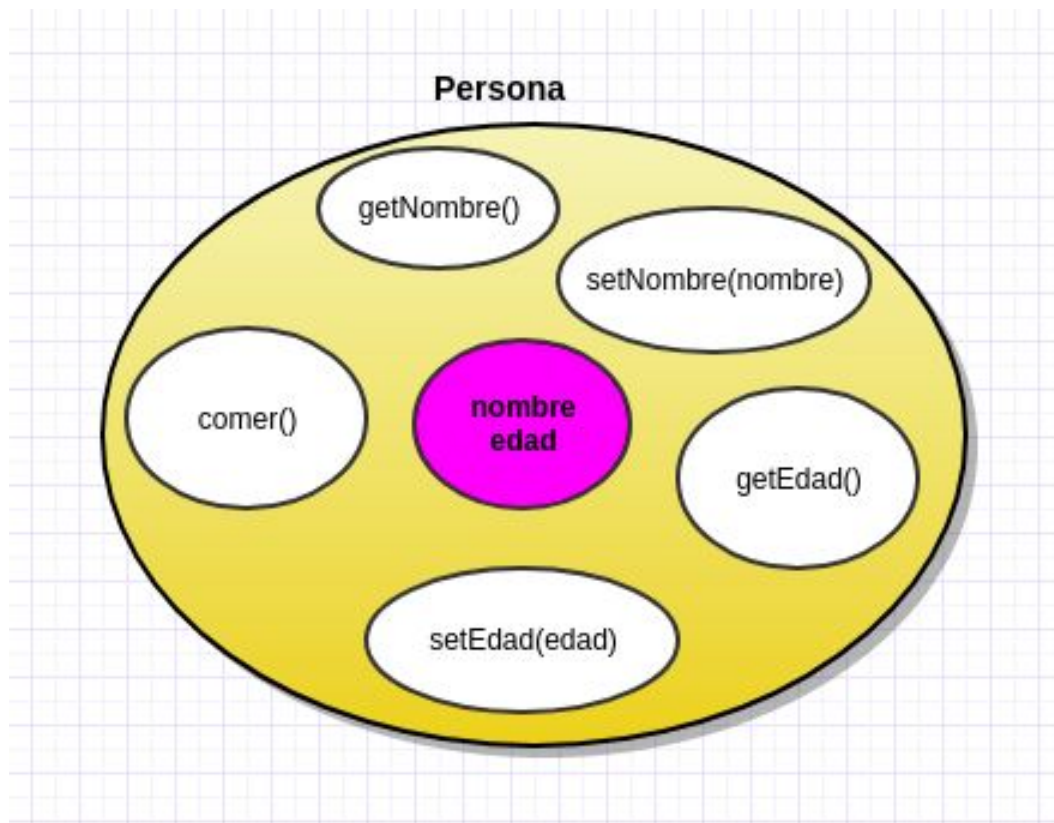
- Nombre
- Edad
- Peso
- Altura
- Fecha de nacimiento
- etc.

Comportamiento:

- Hablar
- Caminar
- Comer
- Imprimir datos
- etc.



Objeto: ejemplo



Agenda

- ~~Objeto~~
- **Clase**
- Instancia
- Comunicación entre objetos
- Modificadores de acceso



Clase

- **Definición:**

Es una plantilla para la creación de objetos.

- Cada clase es un modelo que define un conjunto de variables (atributos) y métodos (comportamiento).

Clase: ejemplo

```
class Persona {  
    //Atributos  
    String nombre;  
    int edad;  
  
    //Métodos  
    void caminar() {  
        //TODO: cuerpo del método  
    }  
  
    String getNombre() {  
        return nombre;  
    }  
}
```


Agenda

- Objeto
- Clase
- **Instancia**
- Comunicación entre objetos
- Modificadores de acceso



Instancia

- Definición:

Cada objeto creado a partir de una clase.

```
class Persona {  
    //Atributos  
    String nombre;  
    int edad;  
  
    //Métodos  
    void caminar() {  
        //TODO: cuerpo del método  
    }  
  
    String getNombre() {  
        return nombre;  
    }  
}
```

persona1: **Persona**

persona2: **Persona**

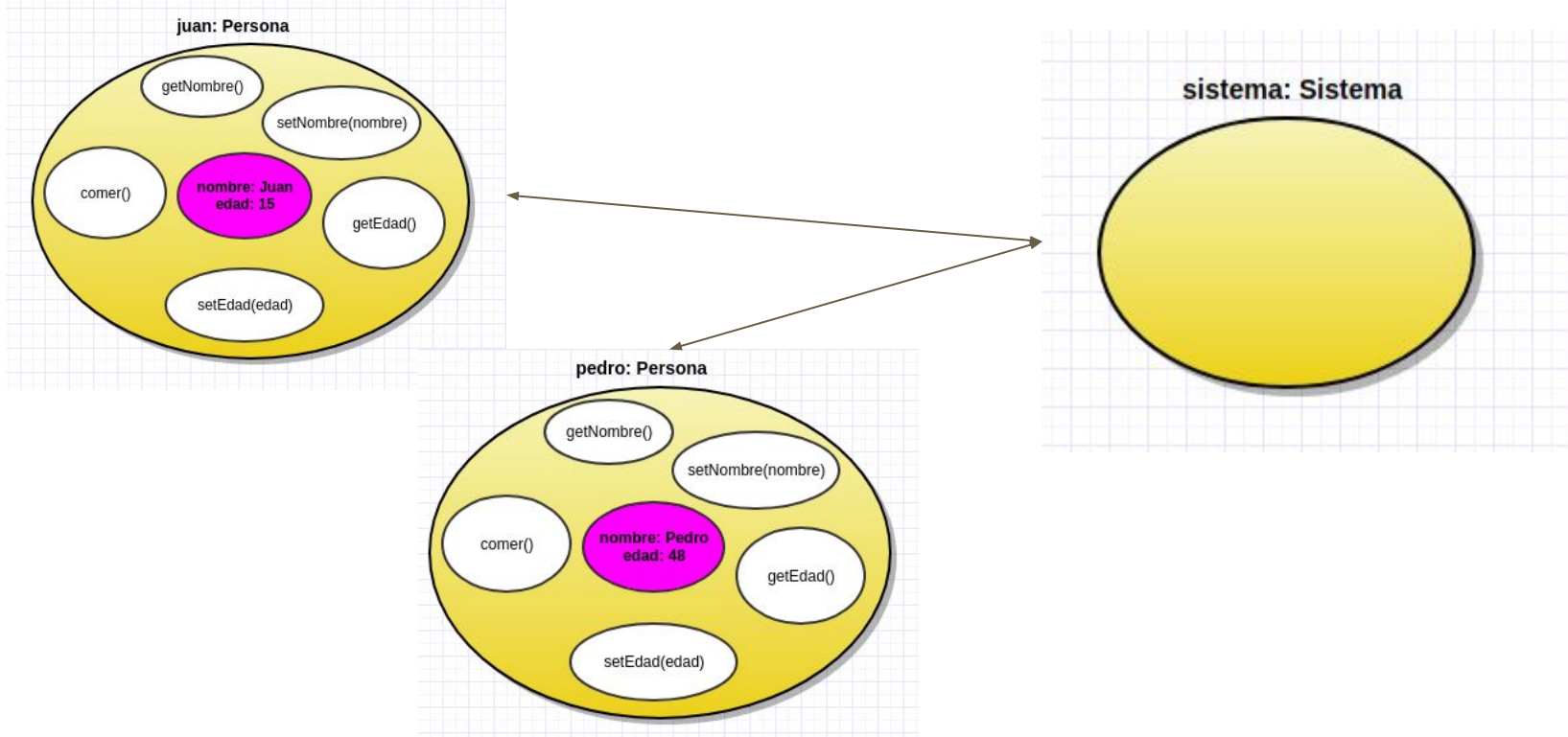
personaN: **Persona**

Agenda

- Objeto
- Clase
- Instancia
- **Comunicación entre objetos**
- Modificadores de acceso



Comunicación entre objetos



Agenda

- Objeto
- Clase
- Instancia
- Comunicación entre objetos
- **Modificadores de acceso**



Modificadores de acceso

- **public:** ofrece la máxima visibilidad. Una variable, método o clase será visible desde cualquier clase.
- **private:** cuando un método o un atributo es declarado como private, su uso queda restringido al interior de la misma clase.
- **protected:** un método o atributo declarado como protected es visible para las clases del mismo paquete y subclases.
- **(default):** visibilidad para clases del mismo paquete.

Modificadores de acceso

Visibilidad	Public	Protected	Default	Private
Desde la misma Clase	SI	SI	SI	SI
Desde cualquier Clase del mismo Paquete	SI	SI	SI	NO
Desde una SubClase del mismo Paquete	SI	SI	SI	NO
Desde una SubClase fuera del mismo Paquete	SI	SI, a través de la herencia	NO	NO
Desde cualquier Clase fuera del Paquete	SI	NO	NO	NO

Modificadores de acceso: ejemplo

```
public class Persona {  
    //Atributos  
    private String nombre;  
    private int edad;  
  
    //Métodos  
    public void caminar() {  
        //TODO: cuerpo del método  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
}
```