

# Programación orientada a objetos

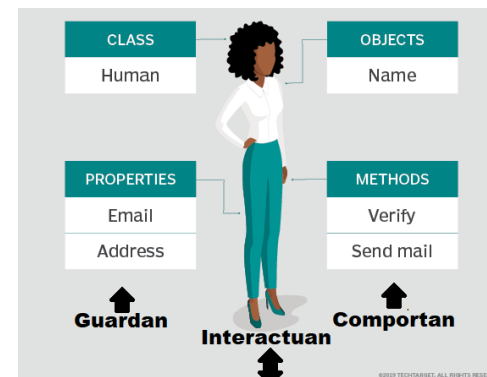
## CI-0112 Programación 1

Sivana Hamer - [sivana.hamer@ucr.ac.cr](mailto:sivana.hamer@ucr.ac.cr)  
Escuela de Ciencias de la Computación e Informática  
Universidad de Costa Rica  
Licencia: CC BY-NC-SA 4.0



UNIVERSIDAD DE COSTA RICA

Programación orientada a objetos (OOP) nos enfocamos en los objetos...

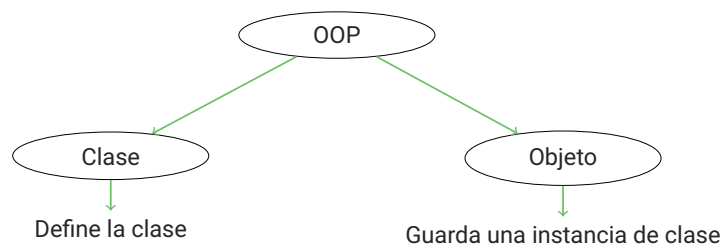


UNIVERSIDAD DE COSTA RICA

[sivana.hamer@ucr.ac.cr](mailto:sivana.hamer@ucr.ac.cr)

1

Existen dos definiciones sumamente importantes...



Una clase es una plantilla de como definir objetos

```
Keywords { public class Student{
Cabecera {
Atributos { private String name;
private String id;
Constructor { public Student(String name, String id){
this.name = name;
this.id = id;
}
Método { public void print(){
System.out.println("Hello! My name is " +
this.name + " and my student ID is " + this.id);
}
}
```

UNIVERSIDAD DE COSTA RICA

[sivana.hamer@ucr.ac.cr](mailto:sivana.hamer@ucr.ac.cr)

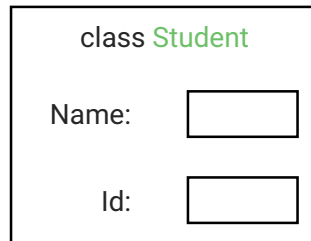
2

UNIVERSIDAD DE COSTA RICA

[sivana.hamer@ucr.ac.cr](mailto:sivana.hamer@ucr.ac.cr)

3

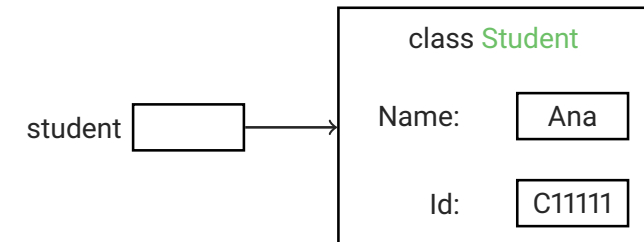
Los atributos (a.k.a. propiedades, variables de instancia o campos) son las características del objeto



### Nota

Generalmente se guardan los atributos en privado

El constructor es un método especial (sin una variable de retorno) que inializa un objeto



Un objeto es una instancia de una clase

```
public class School{
    public static void main (String []
    args){

        //Create Student Pedro
        Student pedro = new Student();
        pedro.name = "Pedro";
        pedro.id = "D10010";
        pedro.print();
        }

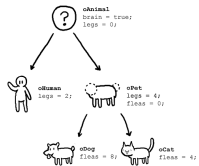
        //Create Student Ana
        Student ana = new Student();
        ana.name = "Ana";
        ana.id = "M10010";
        ana.print();
    }
}
```

Instancia de Pedro

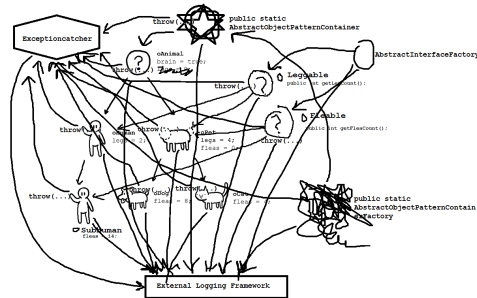
Instancia de Ana

## Vamos a crear nuestras propias clases en grupos

## What OOP users claim



## What actually happens



## Referencias I

u/emGiin, "The world seen by an "object-oriented" programmer."  
[Online]. Available:

[https://www.reddit.com/r/ProgrammerHumor/comments/61sav3/the\\_world\\_seen\\_by\\_an\\_objectoriented\\_programmer/](https://www.reddit.com/r/ProgrammerHumor/comments/61sav3/the_world_seen_by_an_objectoriented_programmer/)

u/r0c97, "I love oop." [Online]. Available: [https://www.reddit.com/r/ProgrammerHumor/comments/efzvct/i\\_oop/](https://www.reddit.com/r/ProgrammerHumor/comments/efzvct/i_oop/)

A. B. Downey and C. Mayfield, *Think Java: How to Think Like a Computer Scientist*, second edition ed., 2020.

D. J. Eck, *Introduction to Programming Using Java*, eighth edition ed. Geneva (NY): Hobart and William Smith Colleges, Department of mathematics and computer science, 2020.

D. J. Barnes and M. Kölling, *Objects First with Java: A Practical Introduction Using BlueJ*, sixth edition ed. Boston: Pearson, 2017.

u/r0c97, "I love oop." [Online]. Available: [https://www.reddit.com/r/ProgrammerHumor/comments/efzvct/i\\_oop/](https://www.reddit.com/r/ProgrammerHumor/comments/efzvct/i_oop/)

## Referencias II

[Online]. Available: <https://searchapparchitecture.techtarget.com/definition/object-oriented-programming-OOP>