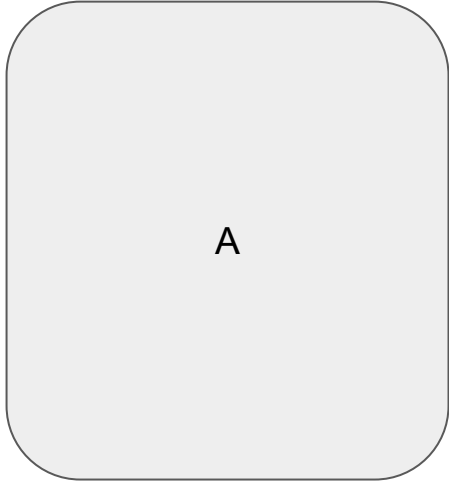


# Programación Orientada a Objetos

Francisco Riquelme

## Objeto



**Características internas:**

**Atributos**

**Es algo propio del objeto**

**No depende de nada más**



**Objeto: Puerta**

**Características internas:**

**Color**

**Forma**

**Decoración**

**Material**



**Objeto: Puerta**

**Características internas:**

**Otros objetos:**

- **Manilla**
- **Periscopio**



## **Objeto: Manilla**

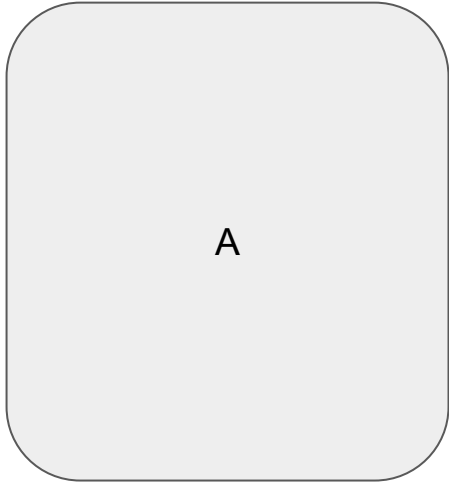
### **Características internas:**

- **Color**
- **Material**

### **Otros objetos:**

- **Cerradura**

**Objeto**



**Comportamiento:**

**Métodos**



**Objeto: Puerta**

**Comportamiento:**

- **Girar manilla (abrir o cerrar)**
- **Ver por periscopio**



**Objeto: Manilla**

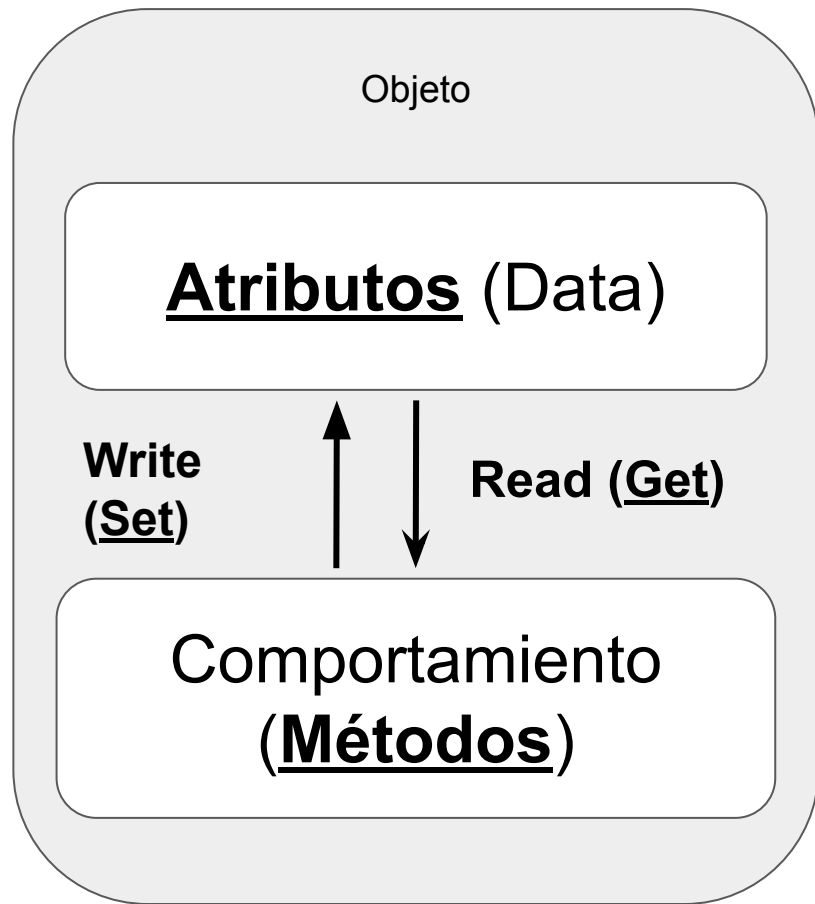
**Comportamiento**

- **Abrir y cerrar**

**Otros objetos con comportamiento:**

- **Cerradura**
  - **Abierto o Cerrado**





Un objeto  
representa una  
abstracción de la  
realidad





¿Qué hay en  
común entre este  
objeto y los  
siguientes?



Objeto: Puerta

Atributos (Data)

Write  
(Set)    ↑    ↓    Read  
(Get)

Comportamiento  
(Métodos)

Objeto: Puerta

Atributos (Data)

Write  
(Set)    ↑    ↓    Read  
(Get)

Comportamiento  
(Métodos)

Objeto: Puerta

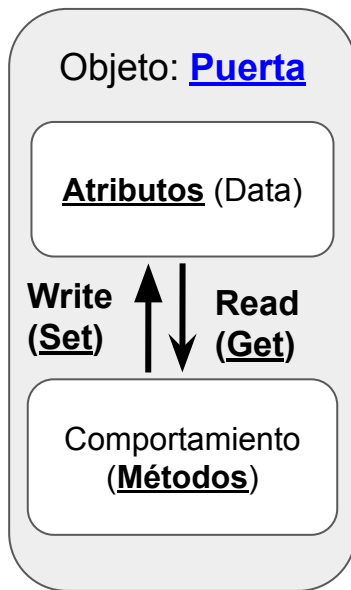
Atributos (Data)

Write  
(Set)    ↑    ↓    Read  
(Get)

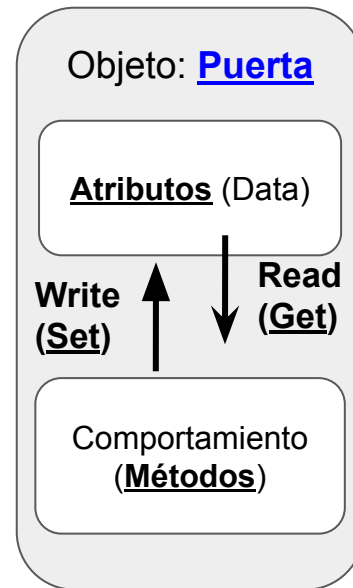
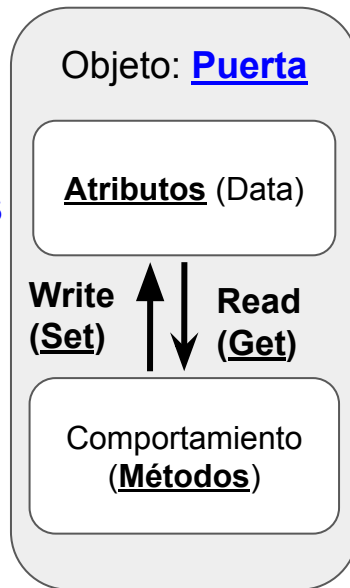
Comportamiento  
(Métodos)



¿Qué hay en  
común entre este  
objeto y los  
siguientes?



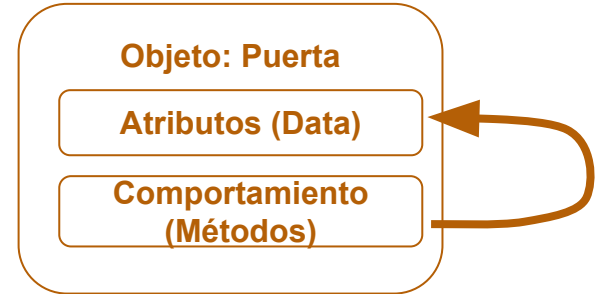
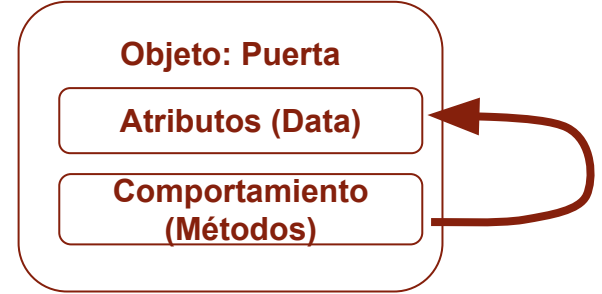
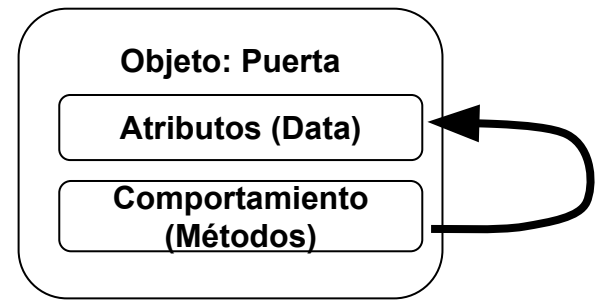
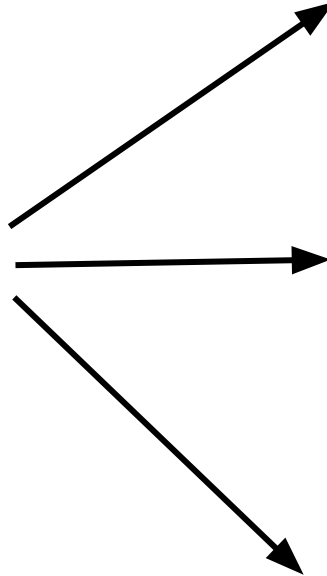
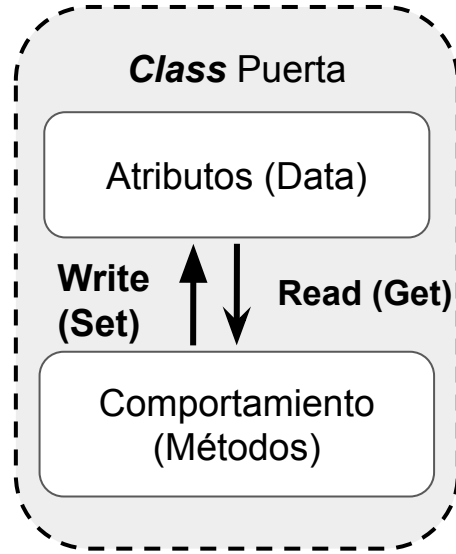
Cambia  
contenido  
de atributos  
pero  
métodos  
se  
mantienen



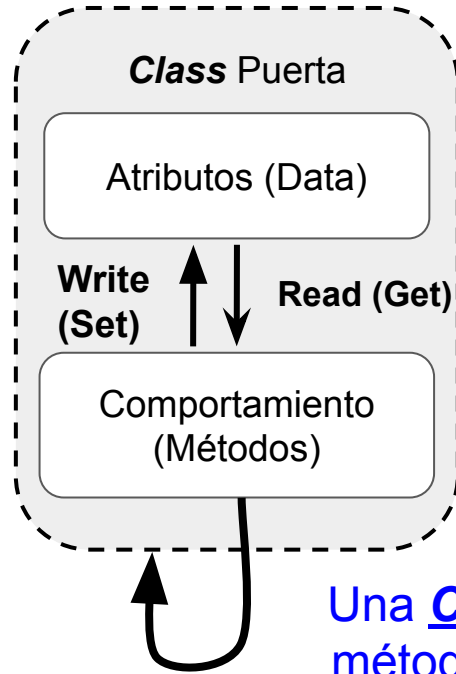
Todos siguen el comportamiento de una **Puerta**

Todos estos objetos siguen el mismo **molde**

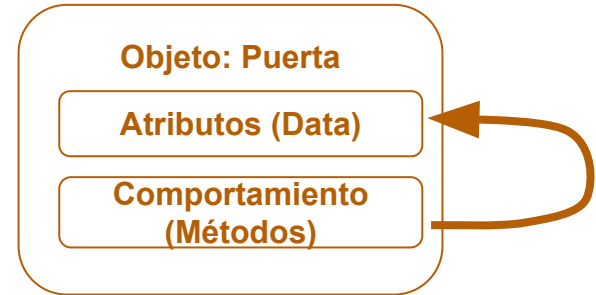
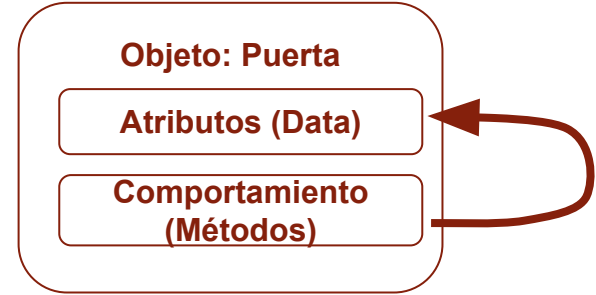
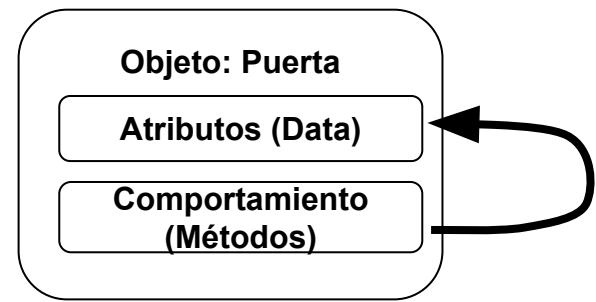
Molde/  
Plano/  
Planilla/



Una **Clase** es una plantilla que guía la creación de objetos



Una **Clase** contiene métodos para crear objetos  
**(Constructor)**



# Constructor de un objeto

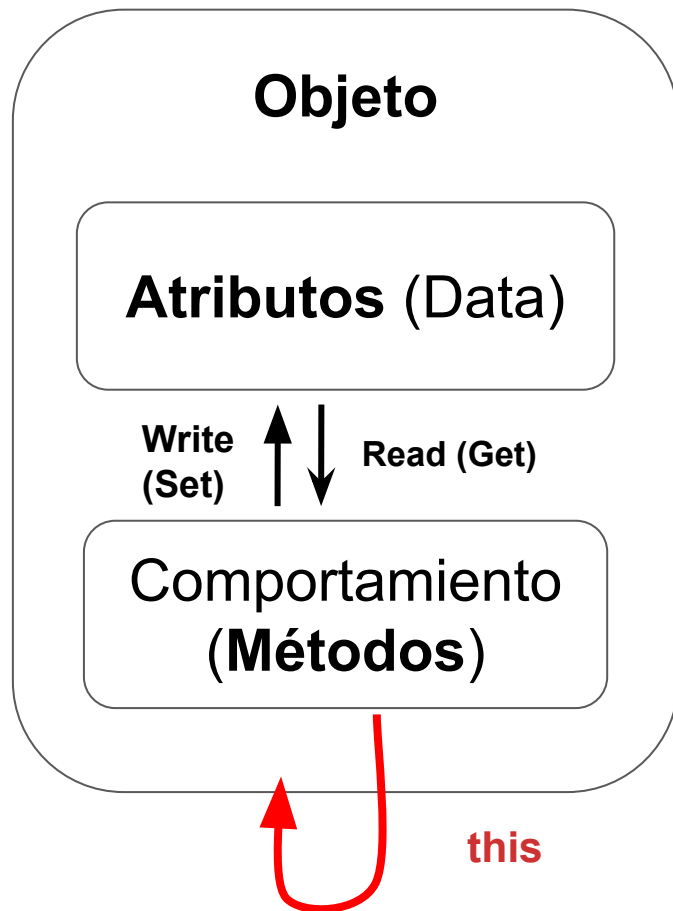
```
class Puerta {  
    int precio;  
    String color;
```

El constructor es un  
método de la clase que  
recibe el mismo nombre de  
la clase

```
Puerta(int precio, String color) {  
    this.precio = precio;  
    this.color = color;  
}
```

this indica “este objeto”  
this.precio -> el precio de  
ESTE OBJETO QUE ESTOY  
CREANDO

```
}
```



```
class Puerta {  
    ... // Atributos de precio y color  
    Puerta(int precio, String color) {  
        this.precio = precio;  
        this.color = color;  
    }  
}
```

**Definición:**

Un objeto es una instancia de una clase

**Definición:**

Una instancia es una copia de la clase con sus valores inicializados almacenado en memoria

MyClass@hashValue

## Memoria

Java Memory  
Address

**Objeto: Puerta**

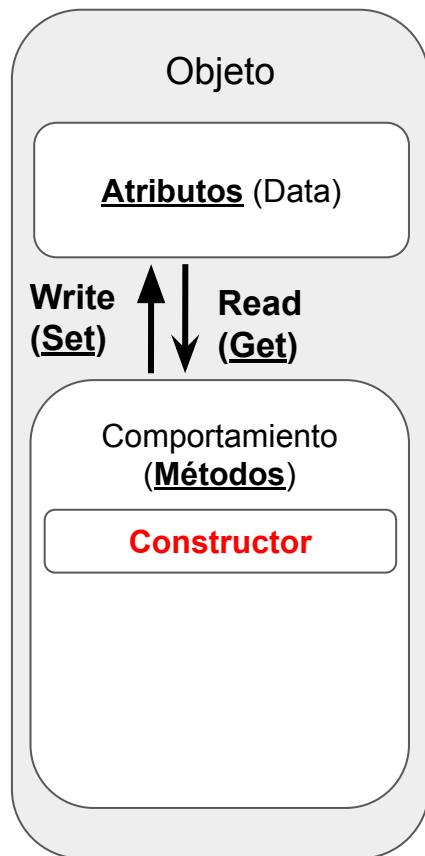
Puerta@74a14482

**Objeto: Puerta**

Puerta@3d4eac69

**Objeto: Puerta**

Puerta@7f8c42a3



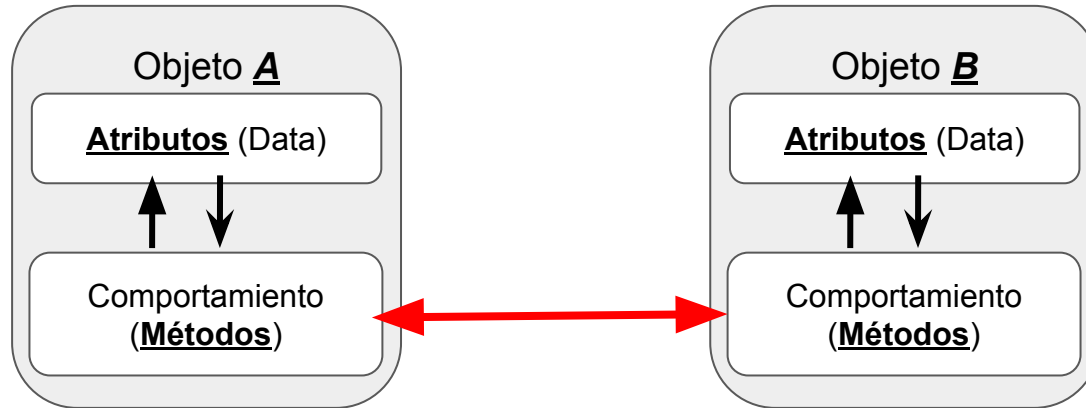
**En la realidad un objeto no actúa solo sin alguna necesidad.**

**En la realidad un objeto no se construye sólo, si no que **algo o alguien lo necesita.****

**Una puerta no se construye por que la puerta quiera construirse por sí misma, si no porque otro objeto la necesita (**colaboración**).**



# Realidad

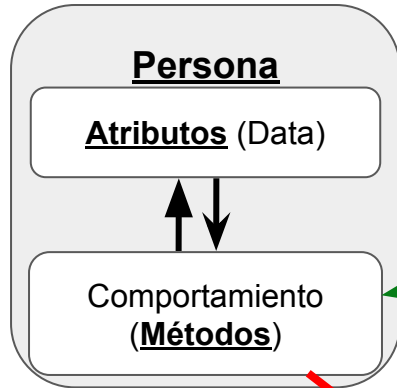


**Los objetos colaboran  
entre sí**

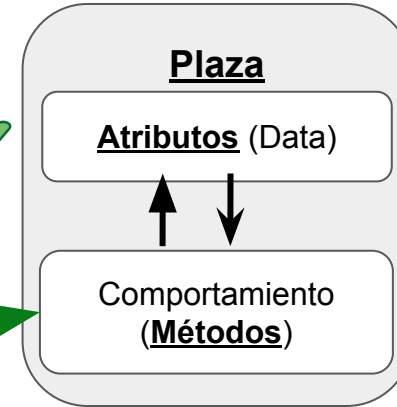
# Encapsulamiento

¿Por qué una persona puede usar una plaza y no debería acceder a la casa de otra persona?

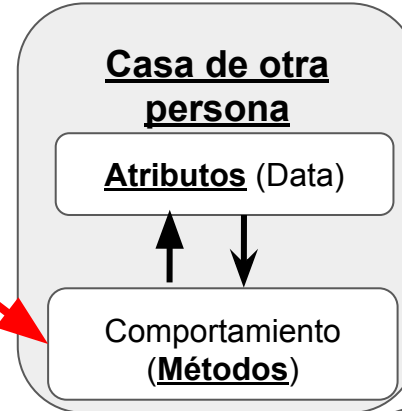
Realidad



Acceso



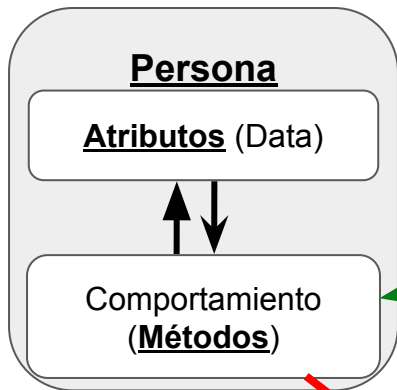
Acceso



Es ilegal, pero, ¿por qué?

Plaza es PÚBLICA  
Casa de otra persona es  
PRIVADA

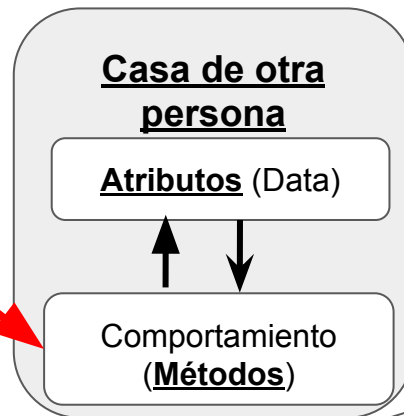
Realidad



Acceso

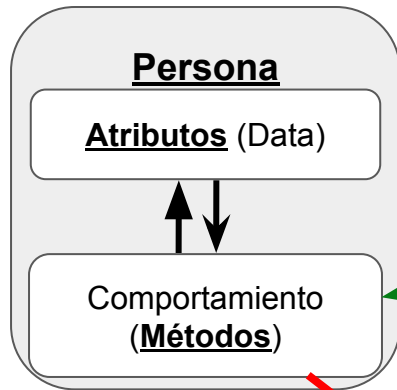


Acceso

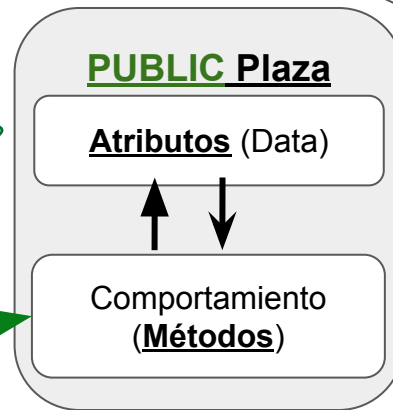


Plaza es PÚBLICA  
Casa de otra persona es  
PRIVADA

Realidad



Acceso



**PRIVATE** Casa de  
otra persona

**Atributos** (Data)

Comportamiento  
(Métodos)

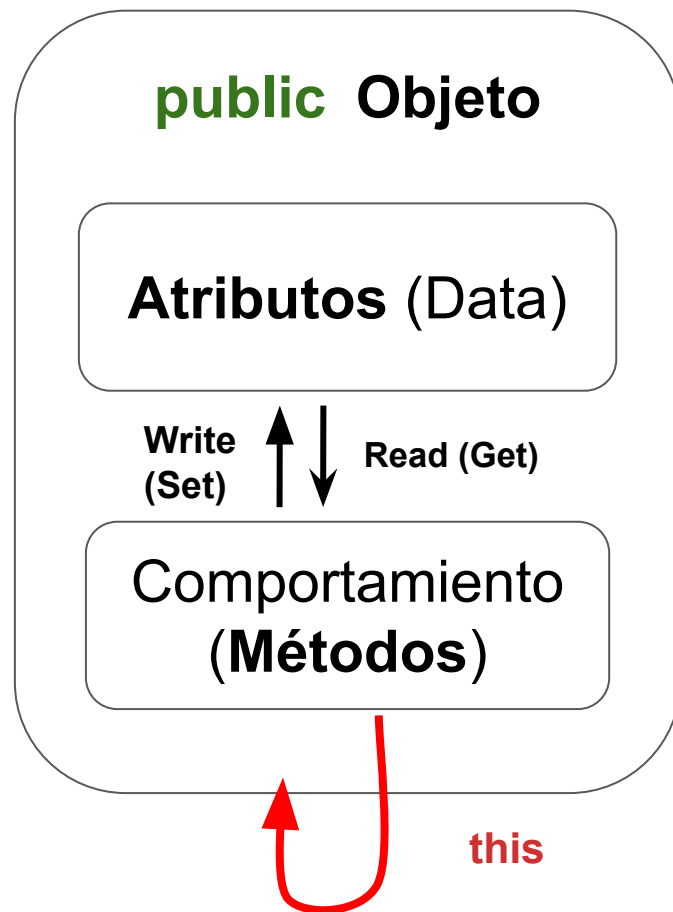
Acceso

**Definición:**  
"public" y "private" son modificadores de  
acceso. Determinan cómo se pueden  
acceder a los atributos y métodos de una  
clase desde otras partes del código

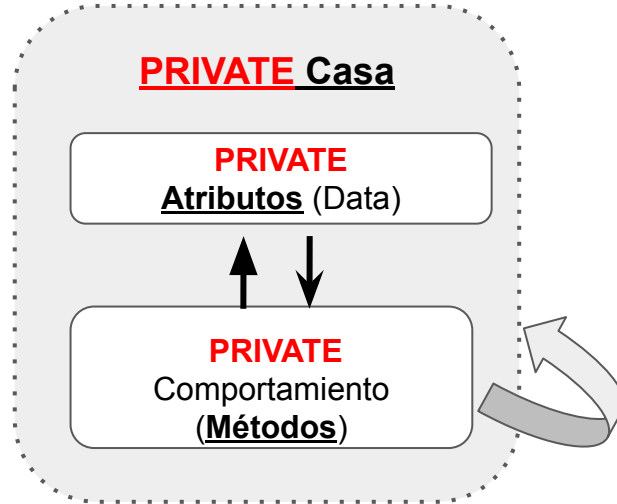
```
public class Puerta {  
    int precio;  
    String color;  
  
    Puerta(int precio, String color) {  
        this.precio = precio;  
        this.color = color;  
    }  
}
```

**Definición:**  
**public** es una *palabra reservada* que expone el contenido de ese campo al sistema

**Definición:**  
**public** puede ser usado para exponer clase, atributos y/o métodos



Todo lo **PRIVATE** es sólo visible para la misma clase y nadie más



Private es alcanzable solo por la misma clase

Todo lo que realizan dentro de sus casas es privado (vida privada), no es expuesto al resto y nadie debería acceder

## Importancia de la POO en el desarrollo de aplicaciones web.

- Mejora la organización, reutilización y mantenimiento del código.
- Permite estructurar el sistema en **clases y objetos**, facilitando la modularidad y escalabilidad.



# Sus principales ventajas incluyen:

**Modularidad:** Divide el código en componentes reutilizables.

**Reutilización:** Usa herencia para evitar duplicaciones.

**Mantenimiento y seguridad:** Encapsula datos y protege la información.

**Integración con frameworks:** Facilita la implementación de patrones arquitecturales.

Los frameworks modernos, como **Spring Boot, Django y Laravel**, están basados en POO, lo que hace más eficiente el desarrollo de aplicaciones web.

# Actividad 1: Abstraer un perro

## Atributos:

Color, Forma, tamaño, sexo, altura, pelaje, nombre, raza, edad, nivel de felicidad, temperamento.

## Métodos:

jugar(juguete), saltar(), correr(), morder(), obedecer(), llorar(), sentarse(), caminar(), ladrar()

comer(comida), beber(agua), irAlBaño().



## Actividad 2: Llevar la abstracción a Código

francisco2435 / Clases

Type to search

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Clases

Private

Unwatch 1

Fork 0

Star 0

main

1 Branch

0 Tags

Go to file

Add file

<> Code

francisco2435

Initial commit

README.md

Initial commit

README

Clases

Clone

Local

Codespaces

Copilot

HTTPS

SSH

GitHub CLI

https://github.com/francisco2435/Clases.git

Clone using the web URL

Open with GitHub Desktop

Download ZIP

About

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

© 2025 GitHub, Inc.

Terms

Privacy

Security

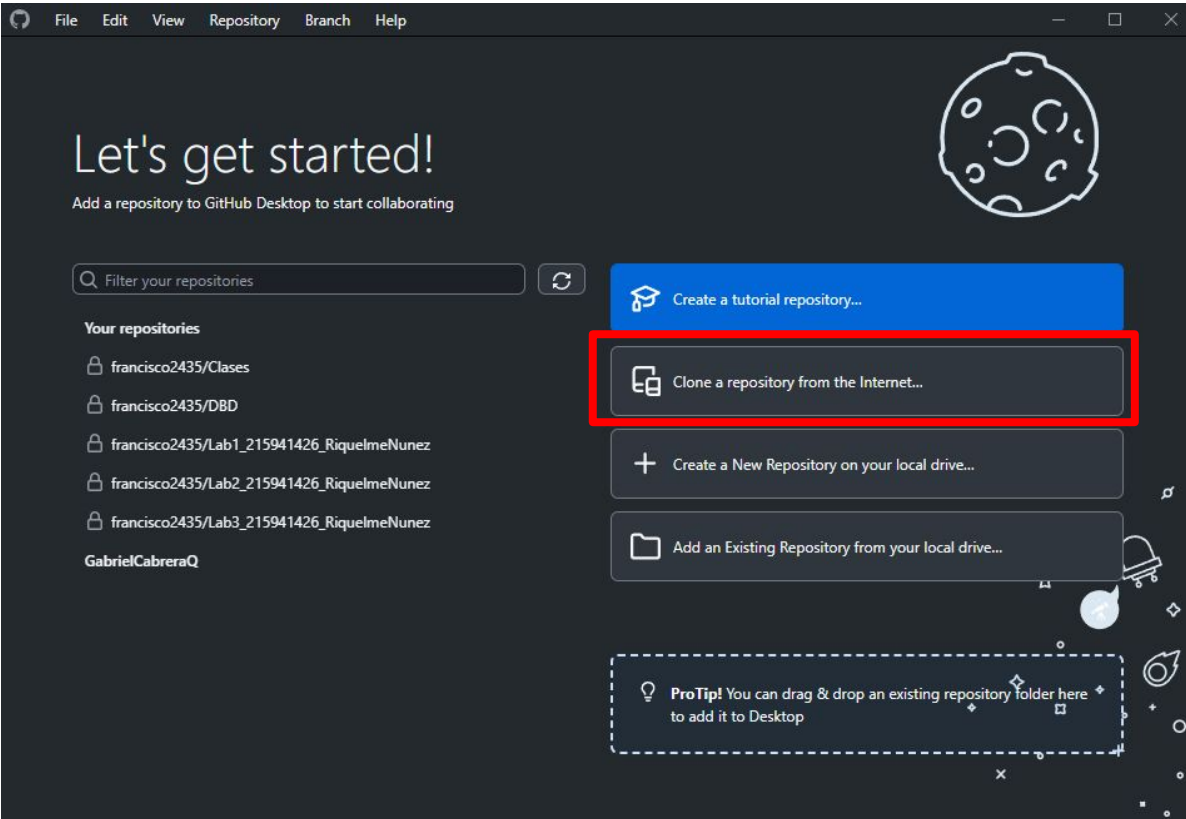
Status

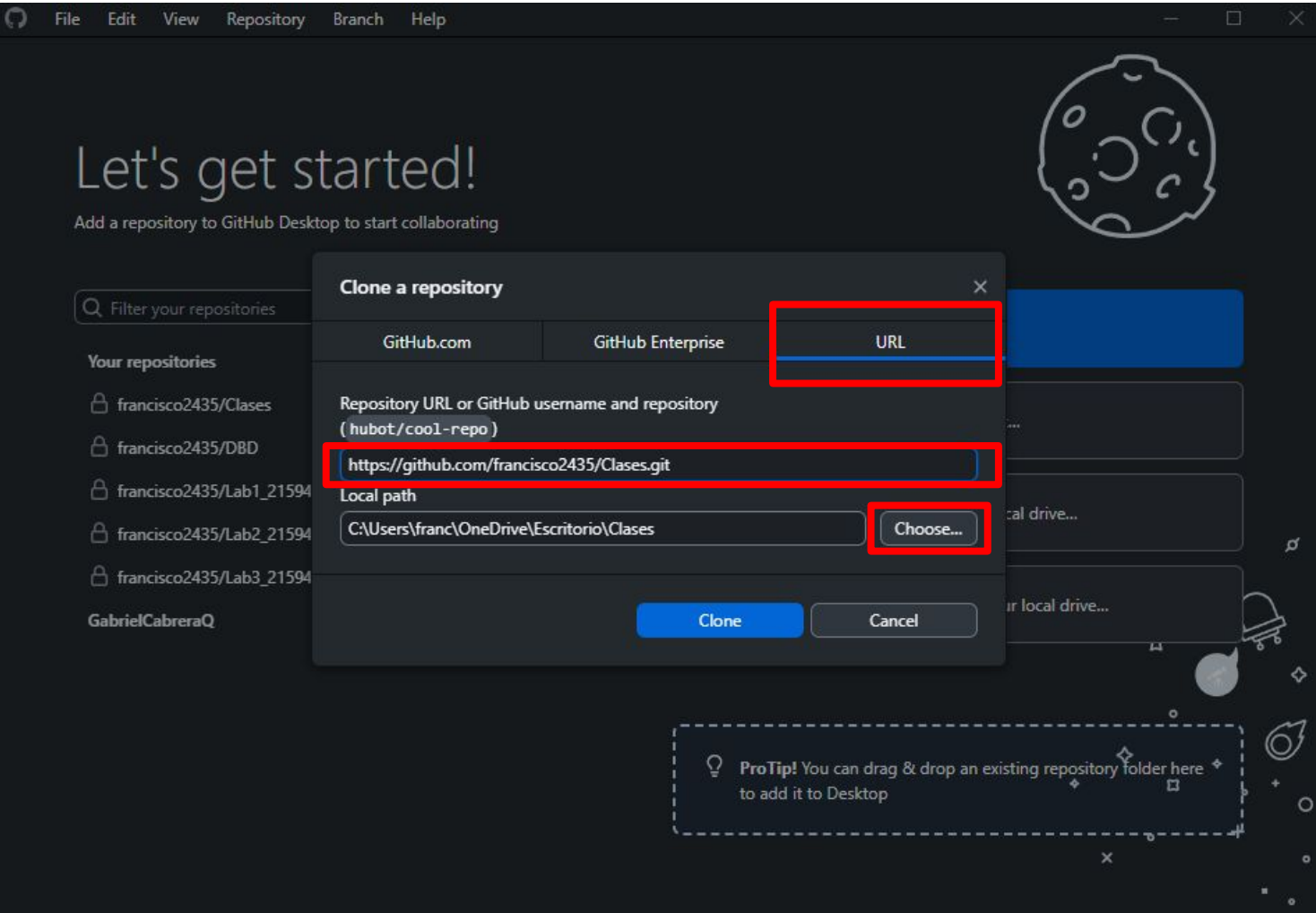
Docs

Contact

Manage cookies

Do not share my personal information





# Pista:

```
public class Animal {  
    private final String nombre;  
  
    public Animal(String nombre) {  
        this.nombre = nombre;  
    }  
}
```

