

A graphic on the left side of the slide. It features a 3D effect with four overlapping rectangular blocks in purple, orange, yellow, and blue. The text 'Agencia de Aprendizaje a lo largo de la vida' is written across these blocks in white. An orange arrow points to the right from the orange block.

Agencia de  
Aprendizaje  
a lo largo  
de la vida

# DJANGO

## Reunión 3

Python - Introducción y POO

# Les damos la bienvenida

Vamos a comenzar a grabar la clase

## Reunión 03

### Python - Introducción

- Fundamentos del lenguaje
- Debug en Python
- Entorno virtual
- Módulos y librerías
- Tipos de datos
- Funciones

## Reunión 04

### Python – Diseño POO

- Diseño de clase (draw.io, EA, Visual Paradigm, etc)
- Modelo de Dominio
- Diagrama de Clases
- Identidad, estado y comportamiento
- Relaciones entre clases
- Polimorfismo

# ¿Qué es Python?

Lenguaje de programación de alto nivel (visto en el curso de Full Stack)

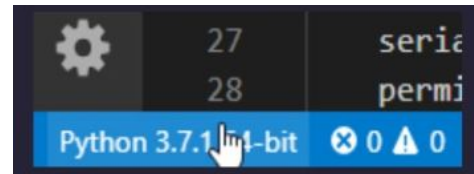
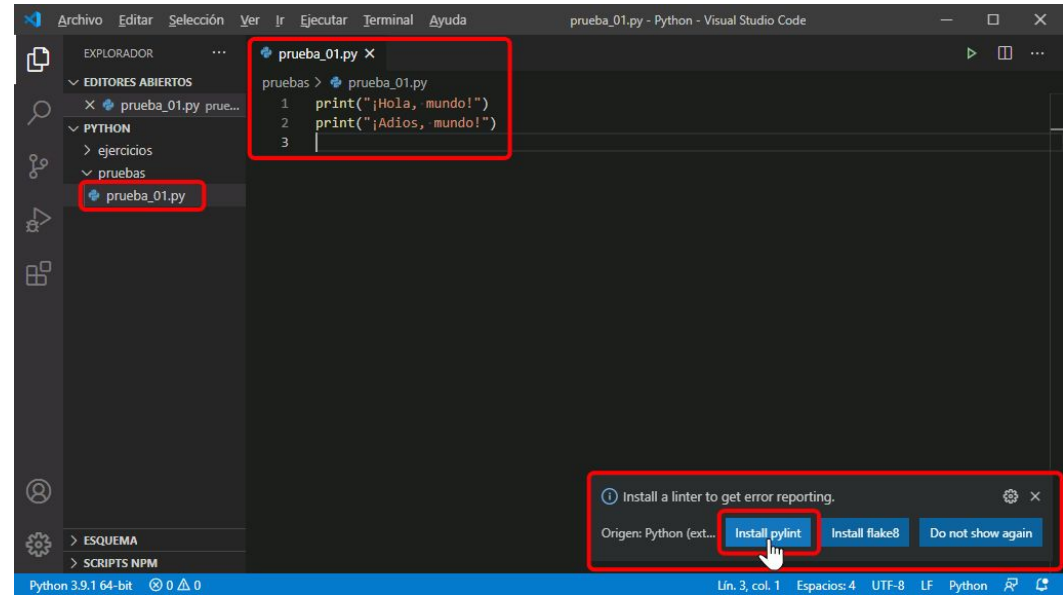
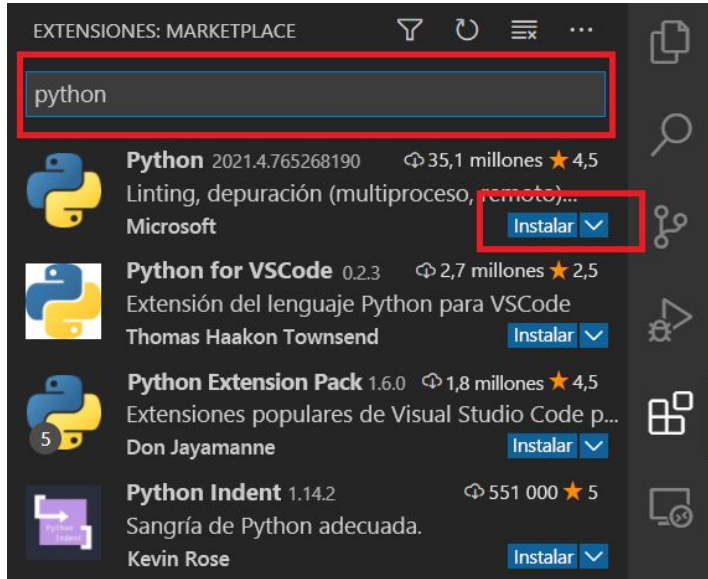
- **MULTIPARADIGMA**
- **MULTIPLATAFORMA**
- **DINÁMICAMENTE TIPADO**
- **FUERTEMENTE TIPADO**
- **INTERPRETADO**



# Visual Studio Code + python™

```
quicksort.py x
20 def partition_random(array, left, right, compare):
21     pivot = left + math.floor(random.random() * (right - left))
22
23     if pivot != right:
24         array[right], array[pivot] = array[pivot], array[right]
25
26     return partition_right(array, left, right, compare)
27
28 def partition_right(array, left, right, compare):
29     pivot = array[right]
30     mid = left
31
```

# Configuración



# Convención de Nombres (PEP8)

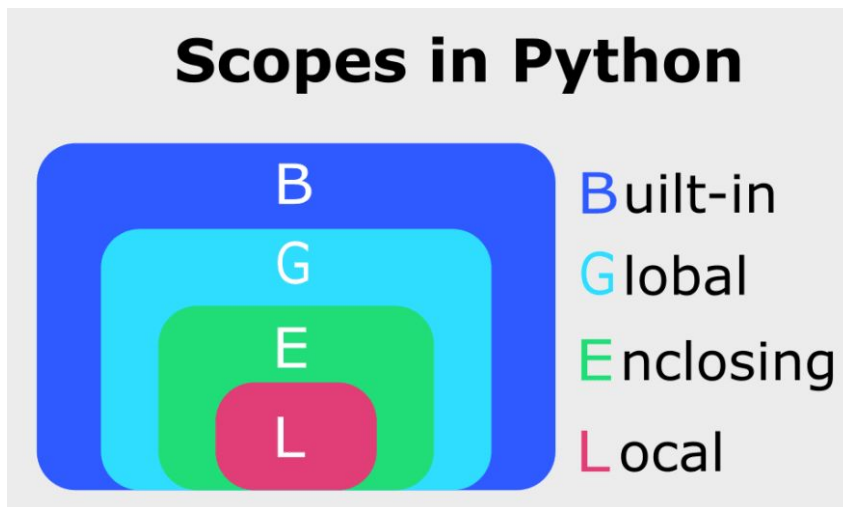
Type	Public	Internal
Packages	lower_with_under	
Modules	lower_with_under	_lower_with_under
Classes	CapWords	_CapWords
Exceptions	CapWords	
Functions	lower_with_under()	_lower_with_under()
Global/Class Constants	CAPS_WITH_UNDER	_CAPS_WITH_UNDER
Global/Class Variables	lower_with_under	_lower_with_under
Instance Variables	lower_with_under	_lower_with_under (protected) or __lower_with_under (private)
Method Names	lower_with_under()	_lower_with_under() (protected) or __lower_with_under() (private)
Function/Method Parameters	lower_with_under	
Local Variables	lower_with_under	

# Palabras reservadas

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield



# Variables y Scope



```
#Global Scope
x = 0
def function():
    #Enclosed scope
    x = 1
    def funcion_interna():
        #Local scope
        x= 2
        print(f"Local scope x={x}")

    funcion_interna()
    print(f"Enclosed scope x={x}")

function()
print(f"Global scope x={x}")
```

# Tipos Primitivos

Ejemplo	Tipo de Dato
<code>x = "Hello World"</code>	str
<code>x = 20</code>	int
<code>x = 20.5</code>	float
<code>x = 1j</code>	complex
<code>x = ["apple", "banana", "cherry"]</code>	list
<code>x = ("apple", "banana", "cherry")</code>	tuple
<code>x = range(6)</code>	range
<code>x = {"name" : "John", "age" : 36}</code>	dict
<code>x = {"apple", "banana", "cherry"}</code>	set
<code>x = frozenset({"apple", "banana", "cherry"})</code>	frozenset
<code>x = True</code>	bool
<code>x = b"Hello"</code>	bytes
<code>x = bytearray(5)</code>	bytearray
<code>x = memoryview(bytes(5))</code>	memoryview
<code>x = None</code>	NoneType

La definición de la función comienza con: "def."

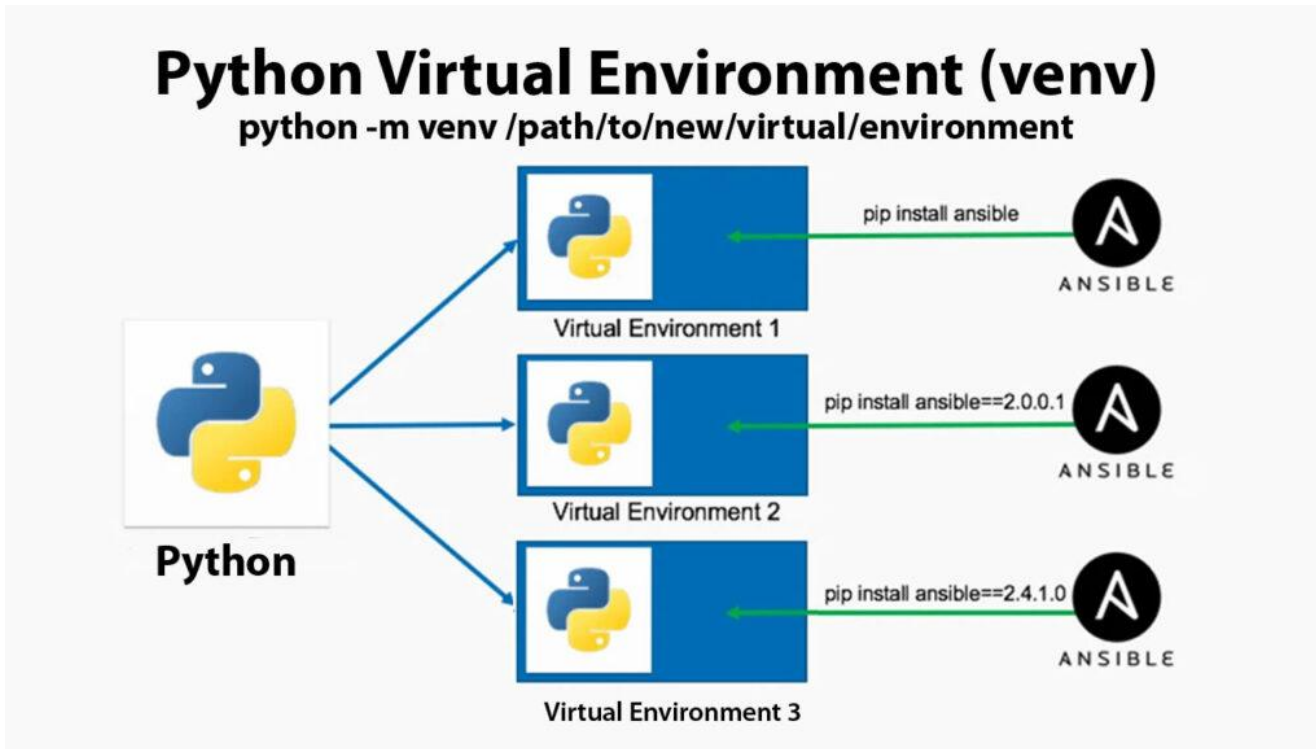
Nombre de la función y sus parámetros

```
def obtener_cantidad(nombre_archivo):  
    """String de documentación"""  
    line1  
    line2  
    return contador
```

La indentación importa...  
La primer línea sin indentación es considerada fuera de la función

La palabra clave 'return' indica que el valor será devuelto a quien llamó a la función.

# Ambientes virtuales



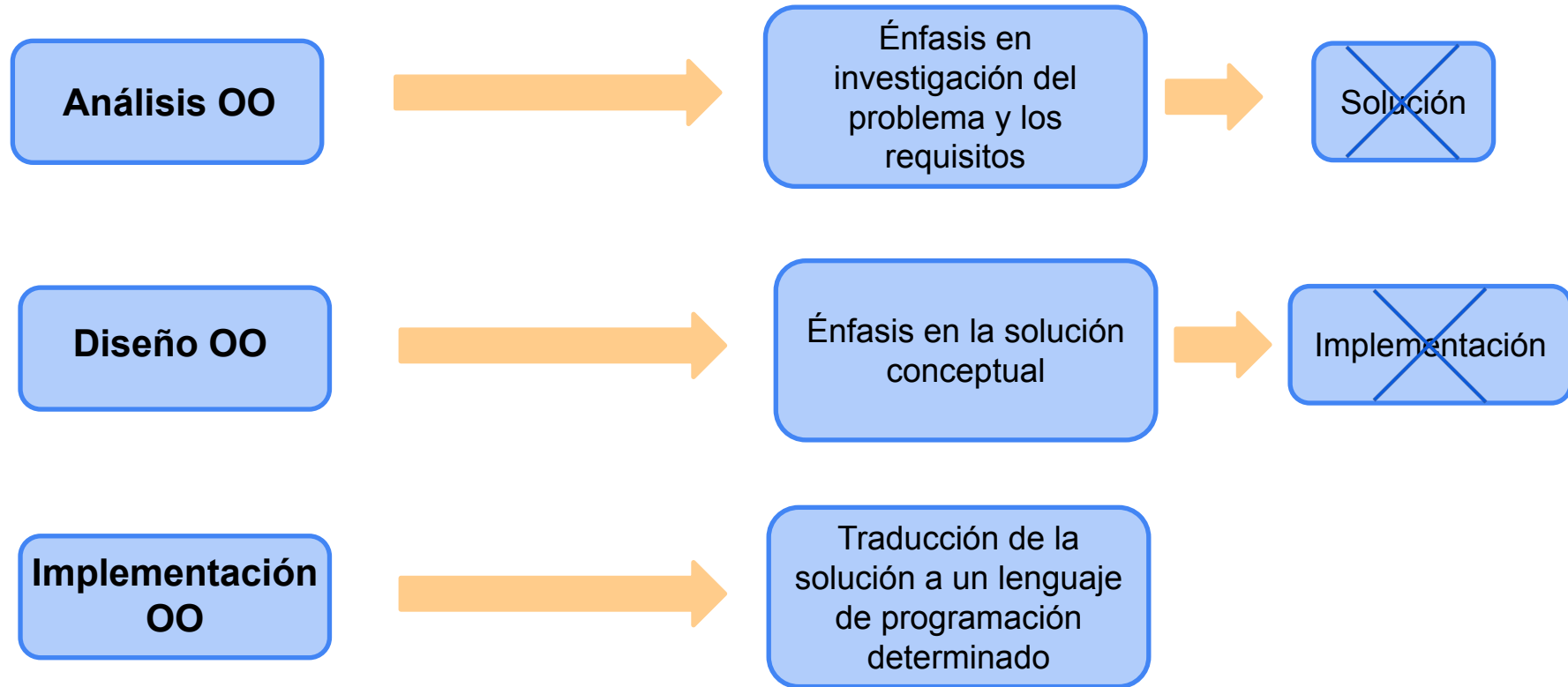
# ¿Qué es POO?

El Paradigma orientado a objetos, define los programas en término de comunidades de objetos. Los objetos con características comunes se agrupan en clases.

## ES UNA FORMA DE VER EL MUNDO

Se impuso por:

- Reduce la brecha entre el mundo de los problemas y el mundo de los modelos.
- Conceptos comunes a lo largo de todo el ciclo de vida
- Uso de patrones
- Aumento complejidad de los sistemas
- Aumento de necesidad de reutilización

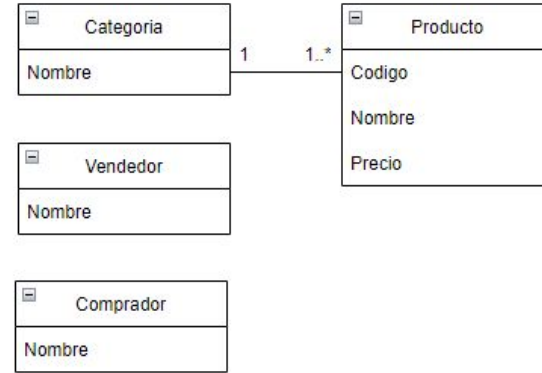


## Análisis OO

Se presta especial atención a encontrar y describir los conceptos del dominio del problema



## Modelo de Dominio

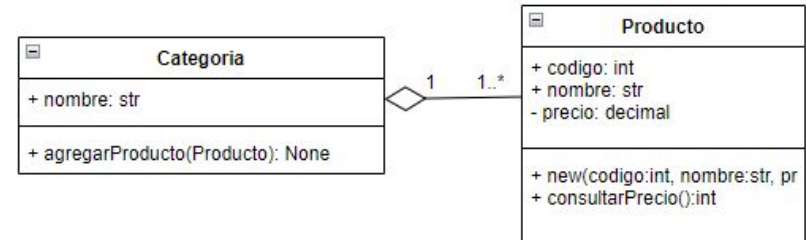


## Diseño OO

Se presta atención a la definición de los objetos software y en como colaboran para satisfacer los requisitos



## Diagrama de Clases



# Estado, comportamiento e identidad

“El **estado** de un objeto abarca todas las propiedades (normalmente estáticas) del mismo, más los valores actuales (normalmente dinámicos) de cada una de esas propiedades”

“El **comportamiento** nos muestra como actúa y reacciona un objeto, en términos de sus cambios de estado y paso de mensajes”

“La **identidad** es aquella propiedad de un objeto que lo distingue de todos los demás objetos”



# Relaciones entre clases

- ❑ Las clases generalmente no se encuentran aisladas, existen tres tipos principales de relaciones:
  - **Dependencias:** relaciones de uso entre clases
  - **Asociaciones:** relaciones estructurales entre clases
  - **Generalizaciones:** conectan clases generales con sus especializaciones (se implementa a través de la herencia)

# Relaciones entre clases

Asociación



Agregación



Composición



Generalización



Dependencia



Realización



# Relaciones entre clases

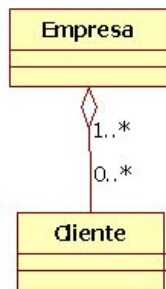
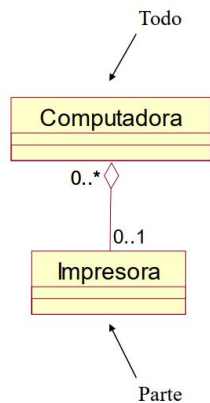
## Dependencia



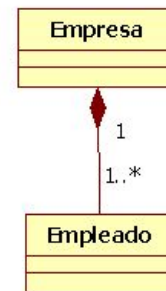
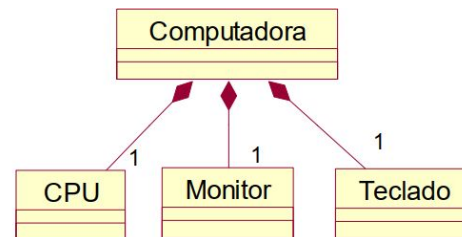
Por ejemplo para resolver una ecuación de segundo grado, tenemos que recurrir a la función `sqrt` de la clase **Math** para calcular la raíz cuadrada.

# Relaciones entre clases

## Agregación

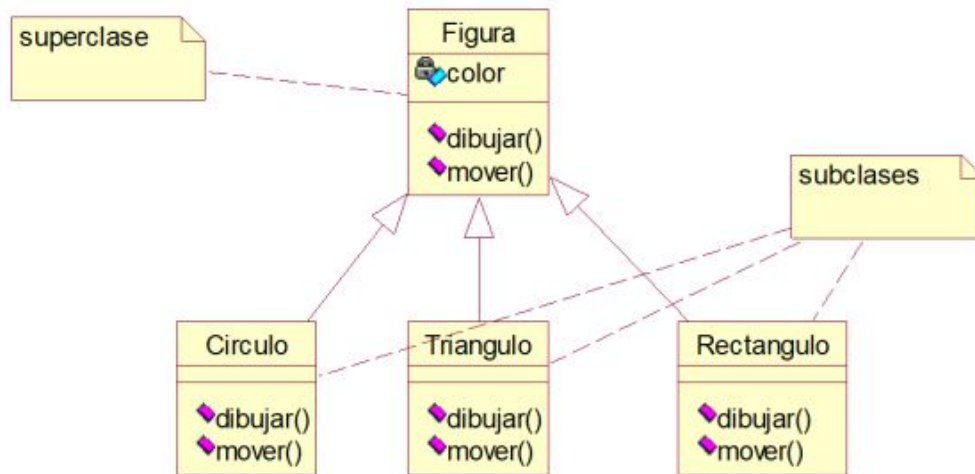


## Composición

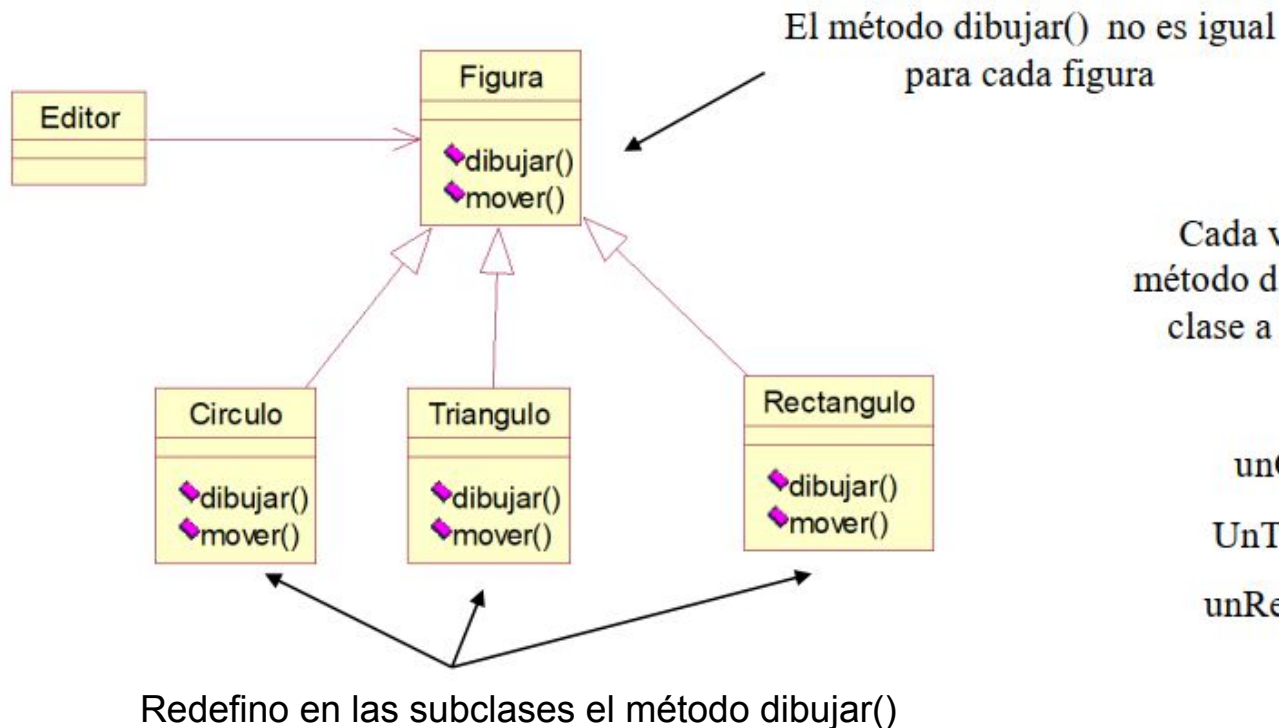


# Relaciones entre clases

## Generalización



# Polimorfismo



Cada vez que se invoque el método dibujar() dependerá de la clase a la pertenece el objeto

```
unCirculo.dibujar();
```

```
UnTriangulo.dibujar();
```

```
unRectangulo.dibujar();
```

**No te olvides de completar la  
asistencia y consultar dudas**

## **Recordá:**

- **Revisar la Cartelera de Novedades.**
- **Hacer tus consultas en el Foro.**

**TODO EN EL AULA VIRTUAL**