

# Curso de introducción a Python

## Introducción a POO y Pandas



# Agenda

- Programación Orientada a Objetos.
  - a. Definición - ¿Qué es?
  - b. Historia - ¿Cómo nace?
- Conceptos y ejemplos.
  - a. Clase e instancias.
  - b. Constructor y Destructor.
  - c. Herencia.
  - d. Polimorfismo.
  - e. Encapsulamiento.
- Pandas.
  - a. ¿Qué es Pandas?
  - b. Series.
  - c. Dataframes.
  - d. Ejemplos básicos.

# **Programación Orientada a Objetos**

# ¿Qué es la Programación Orientada a Objetos?

“La **programación orientada a objetos** es un paradigma de programación que parte del concepto de "objetos" como base, los cuales contienen **información** en forma de campos (a veces también referidos como **atributos o propiedades**) y **código** en forma de **métodos**.” (colaboradores de Wikipedia, 2023)



La POO nace de la necesidad de representar el mundo que nos rodea en código.

# ¿Cómo nace la Orientación a Objetos? 🚀



Ole-Johan Dahl



Kristen Nygaard

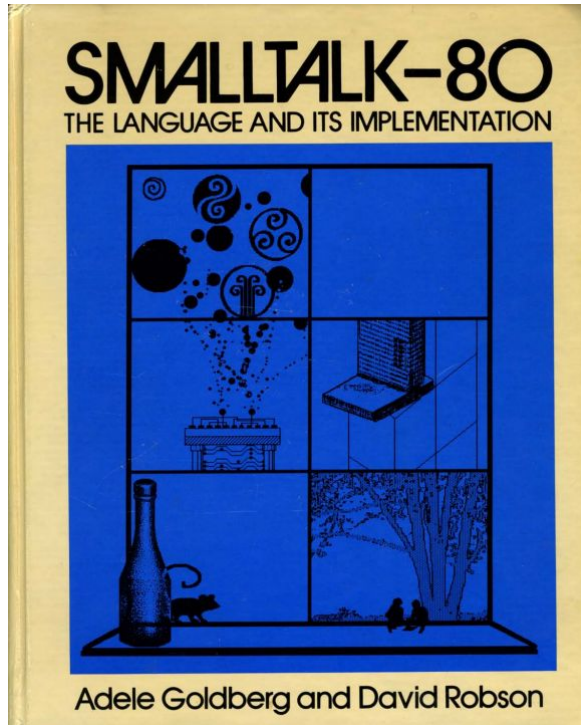


```
Begin  
  OutText ("Hello, World!");  
  Outimage;  
End;
```

## 1960's - 1970's



# ¿Cómo nace la Orientación a Objetos? 🚀



[blue book.pdf](#)



**Adele Goldberg**

**1980's - 1990's**



# Conceptos y Ejemplos



# Clase e instancias

**Clase:** << Receta >>



Atributos	Métodos
<ul style="list-style-type: none"><li>+ Nombre.</li><li>+ Autores.</li><li>+ Ingredientes.</li><li>+ País de origen.</li><li>+ Pasos de preparación.</li></ul>	<ul style="list-style-type: none"><li>- Listar ingredientes.</li><li>- Listar instrucciones.</li><li>- Listar autores.</li></ul>



# Clase e instancias

*Instancia: << Arepas Venezolanas >>*



Atributos	Métodos
<ul style="list-style-type: none"><li>+ Arepas.</li><li>+ María y Andrés.</li><li>+ “Masa”, “agua”, “queso”, ...</li><li>+ Venezuela.</li><li>+ “1. Reunir en un bol...”, ...</li></ul>	<ul style="list-style-type: none"><li>- Listar ingredientes.</li><li>- Listar instrucciones.</li><li>- Listar autores.</li></ul>

# Constructor y Destructor

“En Programación Orientada a Objetos, un **constructor** es una **subrutina** cuya misión es **inicializar un objeto** de una clase. En el constructor se **asignan los valores iniciales** del nuevo objeto.” (colaboradores de Wikipedia, 2022)



# Constructor y Destructor

“Un **destructor** en programación orientada a objetos es una función miembro especial **llamadas automáticamente** en la **ejecución** del programa, y por tanto **no tienen por qué ser llamadas explícitamente por el programador.**”  
(colaboradores de Wikipedia, 2022)

Sus principales cometidos son:

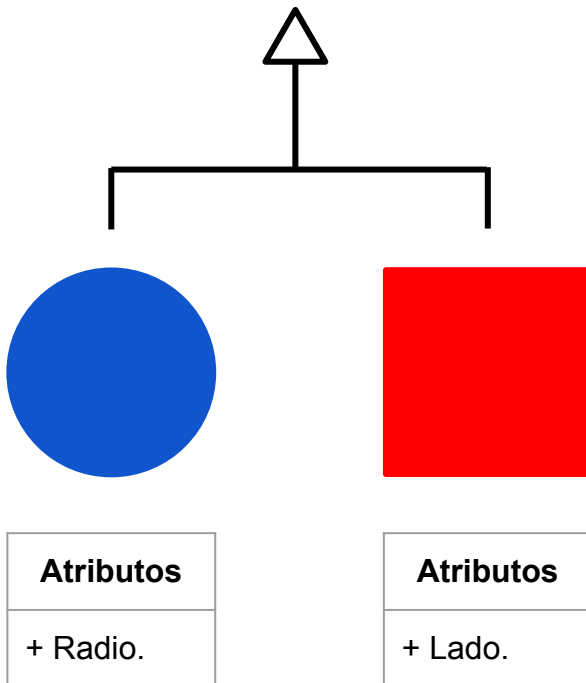
- **liberar los recursos** computacionales que el objeto de dicha clase haya adquirido en su tiempo de ejecución al expirar este.
- **quitar los vínculos** que pudiesen tener otros recursos u objetos con este.

Los ejemplos están en el cuaderno de collab: [aquí](#)

# Herencia

*Clase: << Figura >>*

Atributos	Métodos
+ Color.	- calcular_area( ) - calcular_perimetro( )



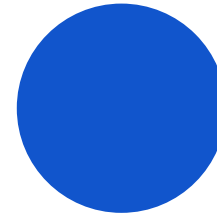
“La **herencia** es uno de los mecanismos más utilizado para alcanzar la **reutilización y la extensibilidad**. A través de ella, se pueden **crear nuevas clases partiendo de una** clase o de una jerarquía de clases preexistente evitando con ello el rediseño, la modificación y verificación de la parte ya implementada.

La herencia **facilita la creación** de objetos a partir de otros ya existentes e implica que una **subclase obtiene** todo el **comportamiento** y finalmente los **atributos** de su superclase.” (colaboradores de Wikipedia, 2021)

# Polimorfismo

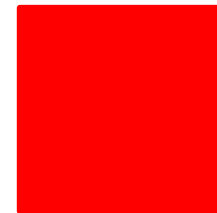
“En programación orientada a objetos, el **polimorfismo** se refiere a la propiedad por la que es posible enviar **mensajes** sintácticamente **iguales** a **objetos** de tipos **distintos**. El único requisito que deben cumplir los objetos que se utilizan de manera polimórfica es **saber responder al mensaje que se les envía.**”  
(colaboradores de Wikipedia, 2023)

¿Saben las clases hijas  
calcular áreas y perímetros?



Área:  $\pi \cdot r^2$

Perímetro =  $2 \cdot \pi \cdot r$



Área:  $l^2$

Perímetro =  $4 \cdot l$

# Encapsulamiento

“Se refiere a la **agrupación de datos** con los métodos que operan en esos datos, o la **restricción del acceso directo** a algunos de los componentes de un objeto.

La encapsulación se utiliza para **ocultar los valores** o el estado de un objeto de datos estructurados dentro de una clase, **evitando el acceso directo** a ellos por parte de los clientes de una manera que **podría exponer detalles** de Implementación **ocultos** o violar la invariancia de estado mantenida por los métodos.” (colaboradores de Wikipedia, 2022)

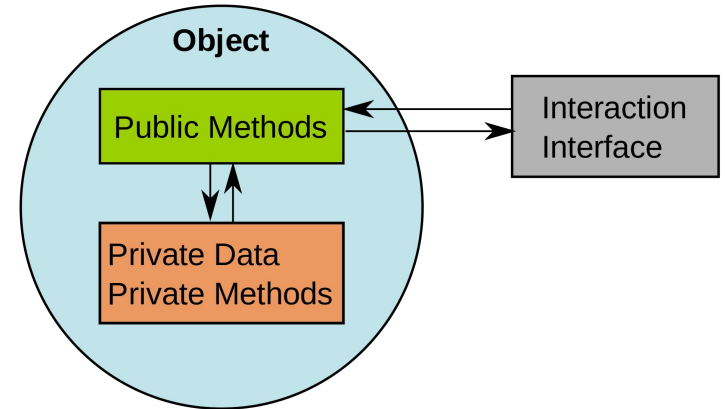


Diagrama de un objeto y su interacción mediante los métodos, no directamente sobre los datos.

Los ejemplos están en el cuaderno de collab: [aquí](#)





# ¿Qué es Pandas?

“Pandas es una biblioteca de Python especializada en la manipulación y el análisis de datos. Ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales, es como el Excel de Python. Es un software libre distribuido bajo la licencia BSD. El nombre deriva del término "datos de panel", término de econometría que designa datos que combinan una dimensión temporal con otra dimensión transversal.” (colaboradores de Wikipedia, 2023)

# Series

Mango		
0		4
1		5
2		6
3		3
4		1

Apple		
0		5
1		4
2		3
3		0
4		2

Banana		
0		2
1		3
2		5
3		2
4		7

# Cantidades de frutas disponible.

mangos = [4, 5, 6, 3, 1]

manzanas = [5, 4, 3, 0, 2]

bananos = [2, 3, 5, 2, 7]

# Por índice.

mangos[0] = 4

manzanas[2] = 3

bananos[4] = 7

# DataFrame

**Series 1**

Mango	
0	4
1	5
2	6
3	3
4	1

+

**Series 2**

Apple	
0	5
1	4
2	3
3	0
4	2

+

**Series 3**

Banana	
0	2
1	3
2	5
3	2
4	7

=

**DataFrame**

	Mango	Apple	Banana
0	4	5	2
1	5	4	3
2	6	3	5
3	3	0	2
4	1	2	7

Diagram illustrating the structure of a Pandas DataFrame, showing the relationship between Columns and Rows.

**Columns:** Name, Team, Number, Position, Age

**Rows:** 0 to 6

**Data:** The table contains data for 7 rows (0 to 6) and 5 columns (Name, Team, Number, Position, Age). The data is as follows:

	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0



# Ejemplos Básicos

*Los ejemplos están en el cuaderno de collab: [aquí](#)*

# Bibliografía

- colaboradores de Wikipedia. (2023). Programación orientada a objetos. Wikipedia, La Enciclopedia Libre [https://es.wikipedia.org/wiki/Programaci%C3%B3n\\_orientada\\_a\\_objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos).
- Wikipedia contributors. (2023). Adele Goldberg (computer scientist). Wikipedia. [https://en.wikipedia.org/wiki/Adele\\_Goldberg\\_\(computer\\_scientist\)](https://en.wikipedia.org/wiki/Adele_Goldberg_(computer_scientist)).
- colaboradores de Wikipedia. (2023). Constructor (informática). Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/Constructor\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Constructor_(inform%C3%A1tica))
- colaboradores de Wikipedia. (2022). Destructor (informática). Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/Destructor\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Destructor_(inform%C3%A1tica))
- colaboradores de Wikipedia. (2023a). Polimorfismo (informática). Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/Polimorfismo\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Polimorfismo_(inform%C3%A1tica))



# Bibliografía

- colaboradores de Wikipedia. (2022). Encapsulamiento (informática). Wikipedia, la enciclopedia libre. [https://es.wikipedia.org/wiki/Encapsulamiento\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Encapsulamiento_(inform%C3%A1tica))
- colaboradores de Wikipedia. (2023a). Polimorfismo (informática). Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/Polimorfismo\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Polimorfismo_(inform%C3%A1tica))
- colaboradores de Wikipedia. (2022). Encapsulamiento (informática). Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/Encapsulamiento\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Encapsulamiento_(inform%C3%A1tica))
- colaboradores de Wikipedia. (2023). Pandas (software). Wikipedia, La Enciclopedia Libre. [https://es.wikipedia.org/wiki/Pandas\\_\(software\)](https://es.wikipedia.org/wiki/Pandas_(software))
- GeeksforGeeks. (2023). Python Pandas DataFrame. GeeksforGeeks. <https://www.geeksforgeeks.org/python-pandas-dataframe/>

