

# Clases Abstractas

August 30, 2023

## 1 Clases Abstractas

Una clase abstracta es una clase que no puede ser instanciada directamente, sino que se crea para ser subclasificada por otras clases. Su principal propósito es proporcionar una estructura base y definir métodos comunes que deben ser implementados por las clases derivadas. Las clases abstractas actúan como plantillas o esqueletos para las clases concretas que las heredan.

En Python, las clases abstractas se definen utilizando el módulo `'abc'` (*Abstract Base Classes*), que proporciona una funcionalidad para crear clases abstractas y métodos abstractos. Para definir una clase abstracta, necesitas heredar de la clase `'ABC'` (*Abstract Base Class*) y utilizar el decorador `__@abstractmethod__` en los métodos que deseas que sean obligatorios para las clases derivadas.

Aquí tienes un ejemplo de cómo se crean y utilizan clases abstractas en Python:

```
[ ]: from abc import ABC, abstractmethod

class Forma(ABC): # Clase abstracta

    @abstractmethod
    def calcular_area(self): # Método abstracto
        pass

# Clases Heredadas:
class Cuadrado(Forma):
    def __init__(self, lado):
        self.lado = lado

    def calcular_area(self):
        return self.lado ** 2

class Circulo(Forma):
    def __init__(self, radio):
        self.radio = radio

    def calcular_area(self):
        return 3.14159 * self.radio ** 2
```

En este ejemplo, la clase Forma es una clase abstracta que contiene un método abstracto llamado `calcular_area()`. Las clases Cuadrado y Circulo son clases derivadas que heredan de Forma y proporcionan implementaciones concretas del método `calcular_area()`.

Si intentas crear una instancia directa de la clase abstracta Forma, obtendrás un error:

```
[ ]: # Esto generará un error, ya que no puedes instanciar una clase abstracta
forma = Forma()
```

```
-----
TypeError                                Traceback (most recent call last)
Cell In[2], line 2
      1 # Esto generará un error, ya que no puedes instanciar una clase abstracta
----> 2 forma = Forma()

TypeError: Can't instantiate abstract class Forma with abstract method
↳ calcular_area
```

Sin embargo, puedes crear instancias de las clases derivadas y utilizar sus métodos implementados:

```
[ ]: cuadrado = Cuadrado(5)
circulo = Circulo(3)

print(f"Área Cuadrado: {cuadrado.calcular_area():.2f}") # Imprimirá el área
↳ del cuadrado
print(f"Área Círculo: {circulo.calcular_area():.2f}")   # Imprimirá el área del
↳ círculo
```

Área Cuadrado: 25.00

Área Círculo: 28.27

Las clases abstractas son útiles cuando deseas definir una estructura común para varias clases relacionadas, pero no quieres permitir la instanciación directa de la clase base. Obligan a las clases derivadas a proporcionar implementaciones específicas para los métodos abstractos, lo que ayuda a garantizar una estructura coherente en tu programa.