

Para saber mais: Criando ABC

Verificamos que é possível usar classes abstratas do sistema, e conseguimos alguns benefícios de bandeja, por exemplo:

- Tenho um erro que me diz, em tempo de instanciamento, se eu esqueci de implementar algum método da superclasse
- E também sou impedido de instanciar um objeto do tipo da superclasse, pra não ter problema com os métodos abstratos.
- Ainda posso aproveitar código dos meus métodos abstratos (que podem ter implementação na classe mãe)

Se você ficou com curiosidade sobre como criar uma classe abstrata, vamos pensar no seguinte caso: imagine que não queremos permitir que ninguém instancie um objeto da classe `Programa`, e queremos garantir que todo mundo implemente o `__str__` nas suas subclasses. Parece uma boa ideia.

Para fazer isso, o nosso código ficaria assim:

```
from abc import ABCMeta, abstractmethod
class Programa(metaclass = ABCMeta):
    @abstractmethod
    def __str__(self):
        pass
```

Apenas com isso, podemos garantir que o `__str__` será implementado nas nossas subclasses, se não for implementado em alguma, será avisado em tempo de instanciamento (não vai conseguir criar instâncias).