

Faça o que eu fiz na aula

Vamos refazer o método que indica há quanto tempo o cliente está cadastrado e vamos utilizar a classe `timedelta` para ajudar nesse processo.

1. Vamos construir uma classe que possua um atributo configurado no momento em que o objeto for criado. Esse atributo será a data de cadastro:

```
class Cadastro:
    def __init__(self):
        data_cadastro = datetime.today()
```

Dessa forma, o momento de cadastro é salvo quando a classe de cadastro for instanciada.

1. Vamos criar um método que vai retornar o dia atual menos o dia de cadastro:

```
def tempo_cadastro(self):
    agora = datetime.today()
    return agora - self.data_cadastro
```

Esse método funciona normalmente, o problema que em tempo de execução o valor retornado sempre será 0 e aí que o `timedelta` entra em ação.

1. Vamos usar o `timedelta()` para modificar o valor da variável `agora`.

```
agora = datetime.today() + timedelta(days=15, minutes=20, seconds=30)
```

1. Juntando tudo, temos a seguinte classe:

```
class Cadastro:
    def __init__(self):
        data_cadastro = datetime.today()

    def tempo_cadastro(self):
        agora = datetime.today() + timedelta(days=15, minutes=20, seconds=30)
        return agora - self.data_cadastro
```

A classe `timedelta` possui mais argumentos além dos que usamos: `timedelta([days[, seconds[, microseconds[, milliseconds[, minutes[, hours[, weeks]]]]]])`. São todos opcionais e podem ser usados como fizemos acima.