07

Do as I did in class

Let's build the base class that implemented the design pattern `Método Template` or `Template Method`

1. First let's insert the class and its attributes

```
import abc

class FilaBase(metaclass=abc.ABCMeta):
    codigo = 0
    fila = []
    clientes_atendidos = []
    senha_atual = None
```

COPY CODE

1. Let's build the abstract methods, which are the methods that must be created in the child classes.

```
    @abc.abstractmethod
    def chama_cliente(self, caixa):
        ...

    @abc.abstractmethod
    def estatistica(self, dia, agencia, flag_detail):
        ...

    @abc.abstractmethod
    def atualiza_fila(self):
        ...
```

COPY CODE

1. We will build a method that will be common to all child classes, this method will be responsible for inserting a customer into the queue.

```
    def inseri_cliente(self) -> None:
        self.fila.append(self.senha_atual)
```

COPY CODE

1. Finally, we will create the template method

```
    def genha_senhas(self):
        self.busca_posicao_fila()
        self.gera_senha_atual()
        self.inseri_cliente()
```

COPY CODE

And so you build your class using a super important design pattern!

Instructor's opinion

If something goes wrong, don't panic, check the error message and try to solve the problem!