

Faça o que eu fiz na aula

Nesta aula criamos uma classe utilizando Type-hints e PEP-8, vamos recriar passo a passo a classe `FilaPrioritaria`.

1. Vamos criar a classe e definir seus atributos.

```
class FilaPrioritaria:
    Codigo : int = 0
    fila = []
    clientes_atendidos = []
    Senha_atual : str = ''
```

2. Precisamos definir o método responsável por gerar as senhas, lembrando que o código da fila é inserido nesse método.

```
def gera_senha_atual(self) -> None:
    self.senha_atual = f'PR{self.codigo}'
```

3. Agora vamos definir o método que verifica qual a posição atual da fila e caso ela seja maior ou igual a 100 a posição da fila deve ser resetada.

```
def reseta_fila(self) -> None:
    if self.codigo >= 100:
        self.codigo = 0
    else:
        self.codigo += 1
```

4. Agora precisamos chamar esses métodos e inserir a senha atual na fila.

```
def atualiza_fila(self) -> None:
    self.reseta_fila()
    self.gera_senha_atual()
    self.fila.append(self.senha_atual)
```

5. Com as senhas geradas e a fila sendo alimentada já podemos nos preocupar em chamar os clientes que devem ser atendidos!

```
def chama_cliente(self, caixa:int) -> None:
    cliente_atual = self.fila.pop(0)
    self.clientes_atendidos.append(cliente_atual)

    return (f' Cliente atual: {cliente_atual }, dirija-se ao caixa: {caixa}')
```

6. Por fim vamos criar o método responsável por gerar as estatísticas, esse método deve retornar uma estatística detalhada ou resumida, de acordo com um argumento recebido.

```
def estatistica(self, dia:str, agencia:str, flag:str) -> dict:
    if flag != 'detail':
        estatistica = {f'{agencia} - {dia}': f'{len(self.clientes_atendidos)}'}
    else:
        estatistica = {}
        estatistica['dia'] = dia
        estatistica['agencia'] = agencia
        estatistica['clientes atendidos'] = self.clientes_atendidos
        estatistica['quantidade de clientes atendidos'] = len(self.clientes_atendidos)

    return estatistica
```