

Classe Abstract Factory

```
class AbstractFactory:
    def create_company(self):
        pass

    def create_person(self):
        pass
```

Classe Concrete Factory

```
class ConcreteFactory(AbstractFactory):
    def create_company(self):
        return CadastroEmpresa()

    def create_person(self):
        return CadastroPessoa()
```

Classe CadastroEmpresa

```
class CadastroEmpresa:
    def cadastrar(self):
        print("Cadastro de empresa realizado.")
```

Classe CadastroPessoa

```
class CadastroPessoa:
    def cadastrar(self):
        print("Cadastro de pessoa realizado.")
```

Uso do padrão Abstract Factory

```
factory = ConcreteFactory()

company_cadastro = factory.create_company()
company_cadastro.cadastrar()
```

```
person_cadastro = factory.create_person()

person_cadastro.cadastrar()
```

Neste exemplo, temos a classe **AbstractFactory** que define os métodos **create_company()** e **create_person()**, que serão implementados pelas classes concretas. A classe **ConcreteFactory** é a implementação concreta da fábrica, que retorna instâncias das classes **CadastroEmpresa** e **CadastroPessoa** através dos métodos **create_company()** e **create_person()**, respectivamente.

As classes **CadastroEmpresa** e **CadastroPessoa** representam as implementações concretas dos cadastros de empresa e pessoa, e cada uma delas possui um método **cadastrar()** que realiza o cadastro específico.

A vantagem de usar o padrão Abstract Factory neste caso é que ele nos permite fornecer diferentes implementações de cadastro (empresa e pessoa) através de uma única interface (**AbstractFactory**). Se no futuro desejarmos adicionar novos tipos de cadastro, como por exemplo cadastro de produtos, bastaria adicionar uma nova classe de cadastro e uma nova implementação na fábrica concreta, sem alterar o código existente.

Espero que este exemplo tenha sido útil para entender o uso do padrão Abstract Factory e como aplicá-lo em um cenário específico de uma página de dicas de segurança com cadastro de empresas e pessoas.