

CHAIN OF RESPONSABILITY: é um padrão de design comportamental que permite passar a solicitação ao longo da cadeia de manipuladores em potencial até que um deles lide com a solicitação. Permite dar a mais de um objeto a oportunidade de processar uma solicitação onde uma solicitação é passada por uma cadeia de manipuladores (handler) e cada manipulador decide se processa a solicitação ou se passa para o próximo manipulador da cadeia.

Ele é indicado quando se deseja que uma solicitação seja processada em partes, uma aplicação desse padrão poderia ser ao autenticar um usuário, onde um manipulador verifica se as credenciais são válidas e um segundo manipulador verifica as permissões do usuário, e se tudo estiver certo autentica o usuário.

Outro exemplo seria no processo de criar conta de usuário, que é o caso desse exemplo.

O contexto escolhido foi sobre um cadastro de usuário, como por exemplo ao criar conta em um sistema web

O padrão CHAIN OF RESPONSABILITY pode ser usado pra realizar a validação dos dados de formulário enviados, a validação do email pra verificar se existe na base de dados e a criação do usuário dividindo cada parte em um manipulador separado

- Handler é a interface que define os métodos a serem implementados pelos manipuladores concretos

- BaseHandler é a classe onde tem o código padrão que é comum a todas as classes de manipuladores.

Essa classe tem um campo para armazenar um ponteiro para o próximo manipulador e um método setHandler() onde as classes

filhas irão definir o seu próximo manipulador da cadeia.

Os clientes podem construir uma cadeia passando um manipulador para o construtor ou através do setHandler()

- CriarUsuarioService é o service que recebe a solicitação do usuário e passa a solicitação para a cadeia de manipuladores.

- BancoDadosService é uma classe que simula um banco de dados.

- Usuario é a classe que representa um usuário no banco de dados.

- CriarContaController é uma classe que representa um controller que recebe requisições dos usuários para criar uma conta.

Concrete Handlers: são os manipuladores concretos que processam cada parte da solicitação, nesse exemplo são 3:

- ValidaDadosFormularioHandler: valida os dados enviados pelo formulário e passa a solicitação para o próximo manipulador

se tudo estiver certo, ou interrompe o processamento em cadeia e retorna falso caso os dados estejam inválidos

- VerificaUsuarioExisteHandler: verifica se o usuário já está cadastrado no banco de dados através do email.

Se o usuário não existe então passa a solicitação para o próximo manipulador da cadeia, ou interrompe o processamento em cadeia e retorna falso caso o usuário já exista

- CriaUsuarioHandler: último manipulador da cadeia, se a solicitação passar pelos 2 manipuladores anteriores insere o usuário no banco de dados e retorna true informando que o usuário foi cadastrado com sucesso.