Curso Softex

Aluna: Vandilma Cândido

**Builder**

O Builder é amplamente usado nas bibliotecas principais do Java: Identificação: O padrão Builder pode ser reconhecido na classe que possui um único método de criação e vários métodos para configurar o objeto resultante. Os métodos do Builder geralmente suportam encadeamento (por exemplo, algumBuilder->configValorA (1)->configValorB (2)->criar () ).

Como funciona esse padrão?

O padrão Builder sugere que você extraia o código de construção do objeto para fora de sua própria classe e mova ele para objetos separados chamados builders. “Builder” significa “construtor”, mas não usaremos essa palavra para evitar confusão com os construtores de classe. O padrão Builder permite que você construa objetos complexos passo a passo.

Segue abaixo um exemplo:

public class UsuarioBuilder {

private String nome;

private String sobrenome;

private Integer idade;

private String telefone;

private String endereco;

public UsuarioBuilder() {

}

public UsuarioBuilder comNome(String nome) {

this.nome = nome;

return this;

}

public UsuarioBuilder comSobrenome(String sobrenome) {

this.sobrenome = sobrenome;

return this;

}

public UsuarioBuilder comIdade(Integer idade) {

this.idade = idade;

return this;

}

public UsuarioBuilder comTelefone(String telefone) {

this.telefone = telefone;

return this;

}

public UsuarioBuilder comEndereco(String endereco) {

this.endereco = endereco;

return this;

}

public Usuario toUsuario() throws UsuarioBuilderException {

validarUsuario();

return new Usuario(nome, sobrenome, idade, telefone, endereco);

}

private void validarUsuario() throws UsuarioBuilderException {

if(nome == null || nome.isEmpty()) {

throw new UsuarioBuilderException();

}

if(sobrenome == null || sobrenome.isEmpty()) {

throw new UsuarioBuilderException();

}

if(idade < 18)

throw new UsuarioBuilderException();

}

}

Ao finalizar o **processo de construção**, o método toUsuario() é chamado. Aqui, ainda acrescentamos uma validação para saber se o processo de construção foi bem executado, mas o mesmo poderia ter sido distribuido a medida que os métodos das etapas fossem chamados.

Assim, nosso processo de construção será semelhante ao exemplo abaixo.

UsuarioBuilder builder = new UsuarioBuilder();

builder.comNome("Diogo").comSobrenome("Moreira");

Usuario novoUsuario = builder.toUsuario();