

Padrões de Projeto

Padrão Builder (2ª Instanciação)

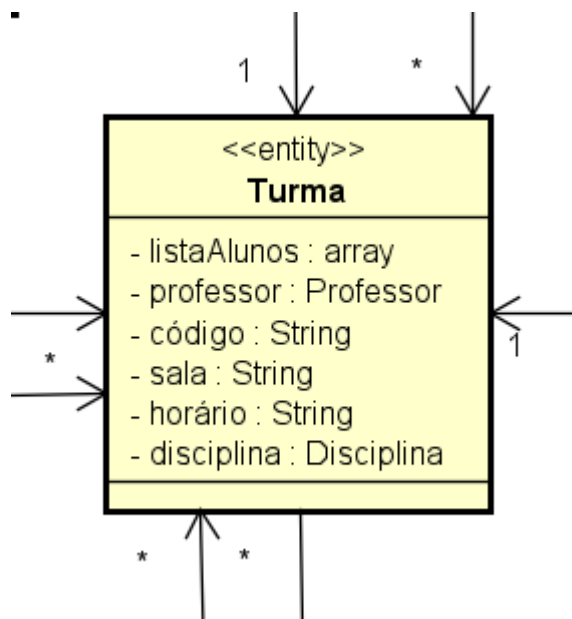
Aluno: Joel Lopes Cunha de Souza

Sobre o padrão

O objetivo do padrão Builder é permitir a criação de objetos passo a passo. Ele também permite a produção de representações e tipos diferentes de um objeto utilizando o mesmo trecho de código.

Problema:

Deseja-se representar a classe Turma de forma que alguns de seus atributos sejam opcionais. Por exemplo, uma turma de TCC não necessita do atributo horário e sala pois não possui uma sala ou horário de aula; também uma turma durante o período de inscrições pode não ter uma sala definida dentro do campus. Se esse problema fosse resolvido por meio de herança ou sobrescrita de construtores, criaríamos uma potencial propagação de subclasses ou de combinações de métodos construtores dentro de Turma.



Solução:

Através do uso do padrão Builder, moveremos o código de criação de uma *Turma* no sistema para uma classe *TurmaBuilderConcreto*. Dessa forma, será possível instanciar diferentes representações de Turma através da chamada ordenada aos passos de construção providos pelo Builder. Adicionalmente, utilizamos um Diretor para extrair as chamadas aos passos de construção presentes no construtor, permitindo que certas formas de construção de uma turma sejam reutilizadas. O construtor se relaciona com uma interface *TurmaBuilder* implementada, atualmente, somente pela classe *TurmaBuilderConcreto*. Isso foi feito para permitir que, futuramente, construtores relacionados ao conceito de turma possam ser adicionados ao sistema implementando a interface *TurmaBuilder* e que esses

construtores sejam utilizados da mesma forma pelo diretor. Isso respeita os princípios aberto/fechado e inversão de dependências, pois o Diretor se relaciona com uma interface de Builder ao invés de um Builder concreto e novas funções podem ser adicionadas sem mudar o código do Diretor.

***(Imagem na página seguinte)**

