

MVC e C#

Aula 1

Etec
Pedro Ferreira
Alves
Mogi Mirim



Introdução aos conceitos

Os conceitos apresentados de maneira teórica neste capítulo não se prendem C#.

São pontos que balizam o modelo orientado a objetos e de que todo programador deve ter conhecimento.

Alguns deles já foram apresentados durante disciplinas anteriores mas, por objetivos didáticos, se concentram novamente aqui.



Objeto

Um objeto pode ser interpretado como qualquer estrutura modular que faça parte de algo, seja um carro, uma pessoa ou um acontecimento etc.

Tendo uma janela como objeto, pode-se afirmar que ela faz parte de uma casa, de um carro ou ainda de uma aplicação (software).

Tanto a casa como o carro ou a aplicação podem também ser vistos como objetos, compondo objetos maiores, ou ainda, se associando a objetos maiores.



Propriedade

Uma propriedade pode ser conceituada como a interface de um objeto com o mundo externo.

Esse mundo externo pode ser visto como outros objetos relacionando-se entre si, que também podem ser chamados de objetos clientes do serviço oferecido.

Retornando ao exemplo do carro como objeto, a Cor do carro pode ser visualizada como uma propriedade. Com base nesse exemplo, identifica-se que as propriedades caracterizam um objeto, ou ainda, o estado dele.



Comportamento

Em relação aos comportamentos possíveis (ou esperados) por um objeto, é possível defini-los como reações de um objeto em decorrência de alguma interação com ele.

Tendo um objeto mais comum com o objetivo desta disciplina, pegaremos como exemplo uma janela de uma aplicação. Nesta há um botão localizado na barra de título, que tem a responsabilidade de fechá-la.

A ocorrência dessa responsabilidade é uma reação a um evento sofrido, que seria o clique nesse botão.



Métodos

Como comentado na explanação sobre comportamentos, para cada evento ocorre uma reação e estes dois itens, juntos com os métodos, caracterizam a realização dos comportamentos.

Os métodos, na realidade, realizam a reação dos eventos capturados pelos objetos.

No exemplo sobre o fechamento de uma janela de aplicação, a ação de fechar a janela precisa ocorrer, pois ela não se fecha sem uma instrução que realize este processo. Sempre que ocorre um evento, este delega a um método a responsabilidade de executar os procedimentos necessários para a realização do evento.

Classe

Uma classe é uma descrição que abstrai um conjunto de objetos com características similares.

Mais formalmente, é um conceito que encapsula abstrações de dados e procedimentos que descrevem o conteúdo e o comportamento de entidades do mundo real, representadas por objetos.

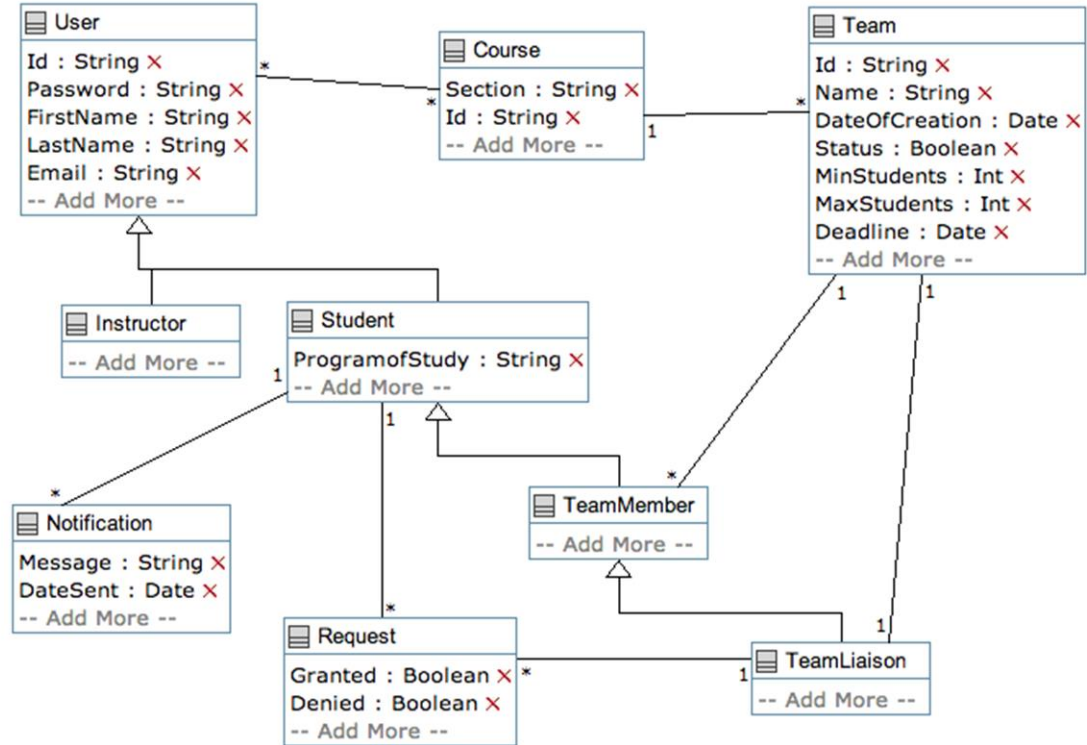


Diagrama de Classes: Domínio Escola

Abstração

► O conceito de abstração para a área computacional é considerada como a habilidade de modelar características do mundo real, de um denominado problema em questão, que o programador busca resolver.





Implementação

Compras

Problema

- Supondo que uma empresa tenha a necessidade de uma aplicação que possibilite o **registro de compras realizadas por ela**, crie um modelo para o formulário que representará essa aplicação.





Análise do Problema

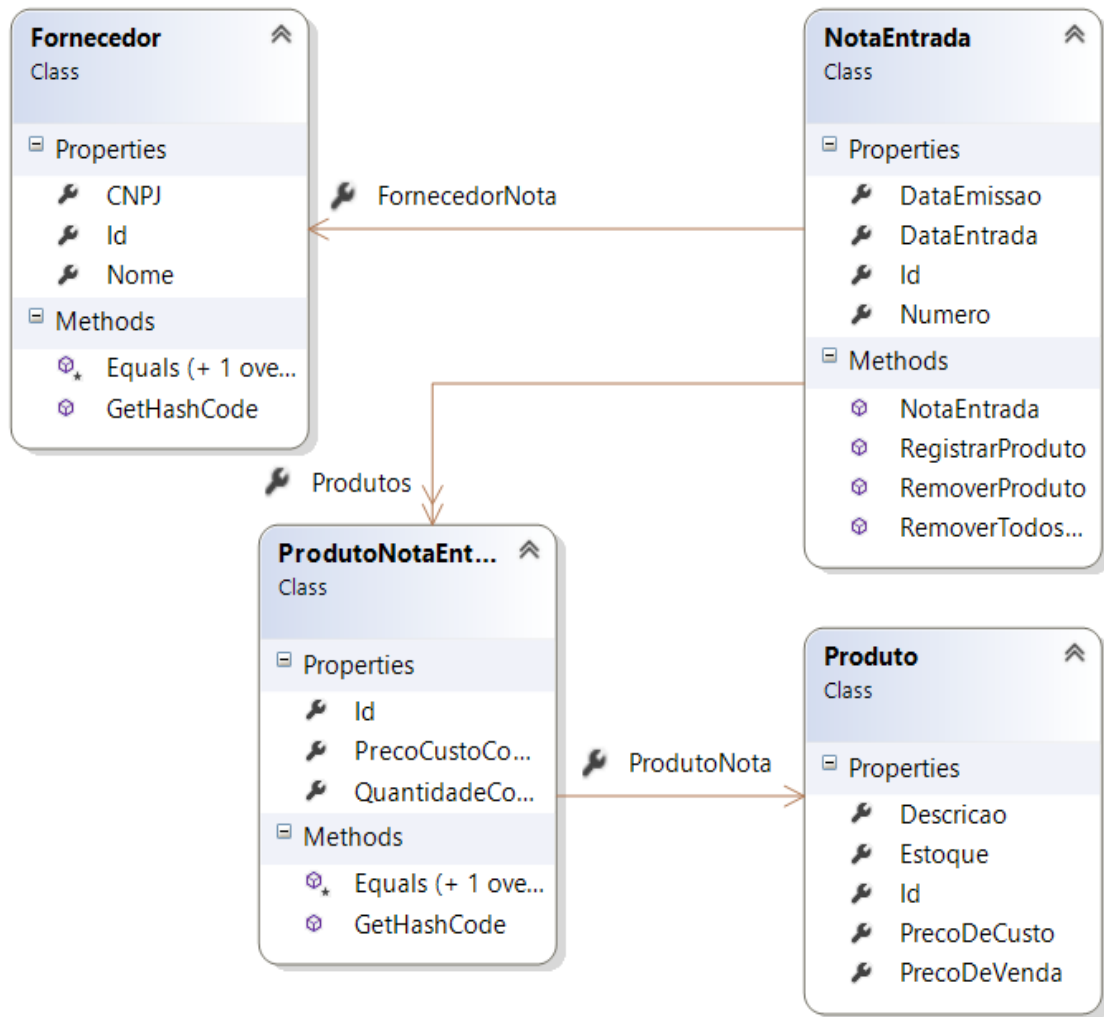
- Com base no proposto e uma análise realizada, abstrai-se que o formulário para o modelo precisa contemplar:
 - Os dados relacionados **ao fornecedor dos produtos, como nome e CNPJ;**
 - Em relação aos dados pertencentes ao processo de registro, o **formulário precisa das datas de emissão da nota pelo fornecedor e a data em que o produto tem a entrada registrada na empresa, além do número da nota fiscal**
 - Para os produtos adquiridos, é preciso informar a **descrição, o preço de custo e a quantidade comprada para cada um, além do código que o identifica.**



Análise do Problema

- Com a análise e a abstração sobre o enunciado realizadas e as propriedades básicas identificadas, é possível constatar que existem dois grupos de informações: um que se refere à nota de entrada (número, fornecedor e datas) e outro que se refere aos produtos adquiridos na compra, classes: **NotaEntrada** e **ProdutoNotaEntrada**.
- Todavia os produtos que compõe as notas de entrada são um ou mais, neste caso, precisamos de uma classe para representar os produtos, classe: **Produto**.
- Seguindo o raciocínio, cada produto é fornecido / distribuído por um fornecedor, precisamos de uma classe que represente este objeto, classe **Fornecedor**.

Modelo


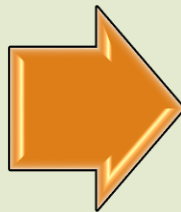




Fim



Obrigado



No próximo
material
apresentamos
uma solução
possível.