

Arquitetura de Sistemas

Aula 3 - Aplicando UML

INTRODUÇÃO



Uma metodologia de desenvolvimento de sistemas gera um conjunto de entregas que começam em modelos conceituais de negócio e vão evoluindo para modelos de sistemas até chegarem à implementação e construção do sistema. Seguir esse conjunto de etapas ajuda muito para que os sistemas sejam mais eficazes e eficientes.

Como visto na aula anterior, o uso de metodologias nos garante o aumento na maturidade na condução dos projetos, mesmo quando os profissionais envolvidos não têm muita experiência. Isso se deve ao fato de as metodologias servirem de referência e de guia na condução dos trabalhos e nelas estarem contidas toda a boa prática da área de referência.

Nesta aula, faremos contato com a metodologia Unified Modeling Language (UML) para desenvolvimento de sistemas e veremos como seus modelos podem ser extremamente úteis no contexto da arquitetura de sistemas.

OBJETIVOS



Descrever como os modelos UML ajudam no desenvolvimento de software por componentes;

Explicar a utilização de UML na arquitetura de sistemas modelados por componentes;

Reconhecer como os modelos UML ajudam no processo de desenvolvimento e garantem mais confiabilidade ao conceito de arquitetura de sistemas.

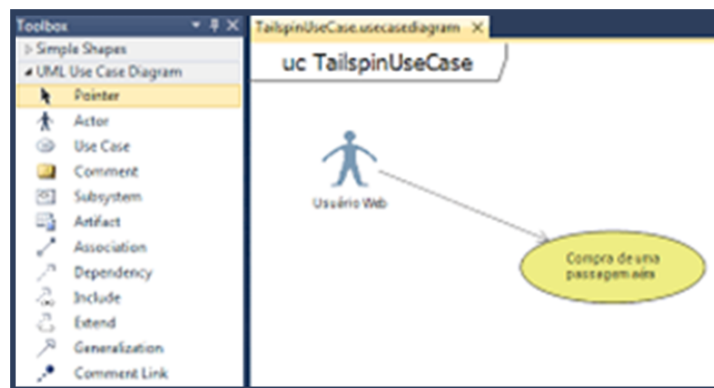
PORQUE PRECISAMOS DE UML NA ARQUITETURA DE SISTEMAS?

UML foi concebida para modelagem de projeto de sistemas orientado a objetos.

Esse conceito tem uma relação direta com o desenvolvimento orientado a componentes, do conceito de arquitetura de sistemas.

Atenção

Mas lembre-se que **o foco desta disciplina está nos aspectos externos de componentes de sistemas** e não no desenvolvimento dos módulos internos, mesmo que eles também possam ser modelados utilizando UML.



Fonte da Imagem: Autor

Neste contexto, um dos itens que usaremos em nossa disciplina é a modelagem de pacotes de interface, em que os componentes são modelados sem a preocupação com a codificação, mas, sim, com a junção dos elementos do sistema para definir sua lógica de funcionamento.

Veremos nesta aula que UML pode nos ajudar a definir e projetar a arquitetura do sistema, servindo como ferramenta de suporte à técnica de modelagem por componentes da arquitetura de sistemas.

Não vamos aqui definir ou indicar uma ferramenta de modelagem UML específica, já que todas as ferramentas que se propõe a isso atendem aos objetivos propostos nesta disciplina.

UML E SUAS GENERALIZAÇÕES



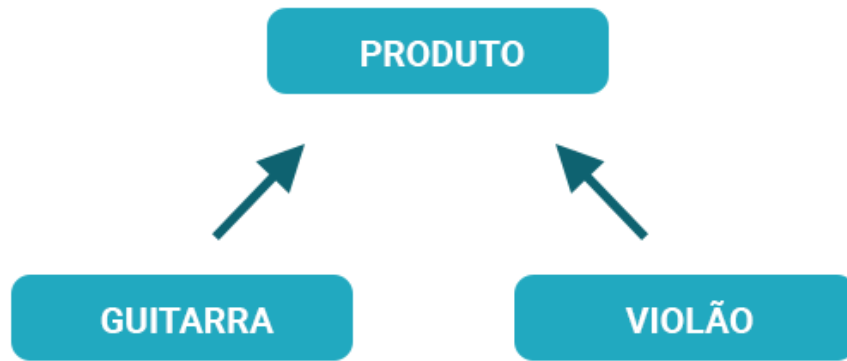
Na UML, temos um grande número de mecanismos de derivação de classes. Porém, com certeza, o mais utilizado na criação e implementação dos componentes é a generalização.



Qualquer classe UML pode sofrer generalização para implementar novas funcionalidades e/ou novos componentes.



Em um conceito mais amplo, sempre que você generalizar uma classe utilizando UML, pode também criar restrições específicas para essa generalização, independente da classe da qual este novo conceito foi estendido.



COMO ATINGIR CONFORMIDADE EM REQUISITOS COM UML?

1

2

3

4

TÉCNICAS DE MODELAGEM UML



Fonte da Imagem: Autor

Veremos aqui como usar UML para modelar os vários artefatos necessários na modelagem de sistemas por componentes.

Esses artefatos nos servirão neste contexto para modelar o negócio e o sistema, na forma dos componentes que formam a arquitetura do sistema.

TÉCNICAS DE MODELAGEM UML

Os diagramas que vamos utilizar usam os mesmos nomes da UML, mas são aqui apresentados com uma funcionalidade específica.

Exemplo

Para modelarmos os conceitos de negócios, usaremos o **diagrama de classes**, pois ele descreve o modelo de conceito do negócio., , Outro exemplo é o **diagrama de especificação de interface** que apresenta a especificação das interfaces. Nesse mesmo caminho temos os seguintes modelos:, , **diagrama de modelo de negócio**, **diagrama de componentes** e **diagrama de implantação**.

Temos também as seguintes variações:

Diagrama de funcionalidades de interface

Descreve o modelo de negócio, suas interfaces e as regras de funcionalidades para essas interfaces.

Diagrama de interação de componentes

É um diagrama de colaboração utilizado para interação entre componentes.

Veremos mais detalhadamente cada um deles nos próximos elementos desta aula.

MODELO CONCEITUAL DE NEGÓCIO

O modelo conceitual de negócio não é um modelo de software, mas sim um modelo de informação que define o domínio do problema.

Sua principal finalidade é identificar e definir as relações existentes dentro do negócio.

Usaremos o diagrama de classe UML para representá-lo, mas é importante ficar claro que a ferramenta é independente do modelo, ou seja, poderemos generalizar cada classe do modelo e amarrá-las do ponto de vista de restrições. Depois, quando formos definir os pacotes, poderemos separar o negócio do sistema.



Os modelos conceituais de negócio, normalmente, definem classes conceituais e suas associações e, como previsto nesse modelo, as associações terão suas multiplicidades.

O modelo pode conter atributos, caso isso seja significativo para explicar o negócio, mas seu uso não é obrigatório. A ênfase desse modelo é definir o domínio e não normalizar suas relações.

MODELO CONCEITUAL DE SISTEMAS

Diagrama de casos de uso

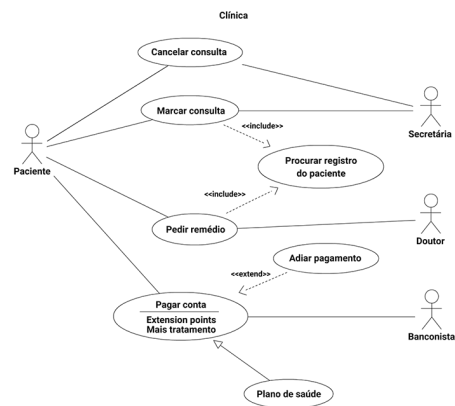
O **diagrama de casos de uso** é modelo que define a funcionalidade do sistema a ser desenvolvido e contém os seguintes elementos:



Relação entre objetos

Quando estamos falando da relação entre objetos, temos o **include**, que informa que o objeto A é parte do objeto B, caso exista entre eles uma relação do tipo include partindo de A.

Outro tipo de relação entre objetos é o **extends**. Esse tipo de relação representa que o objeto D é uma extensão do objeto A, compartilhando suas funcionalidades e implementando algumas novas, que existem somente para ele.



Fonte: O Autor

Descrição de casos de uso

Descrevem, geralmente na forma textual, as funcionalidades existentes no diagrama de casos de uso, no que diz respeito a como seus atores interagem com os objetos do sistema.

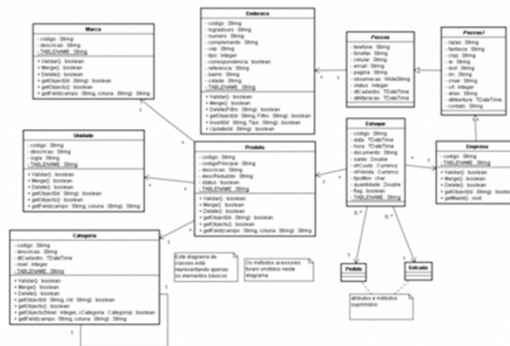
Descrevem também como os objetos se relacionam entre si e, por último, a funcionalidade dos objetos.

Nome do Caso de Uso	Manutenção de Cliente
Atores	Usuários, Administrador
Dados Consumidos	Nome, Endereço, Telefone, Email, CPF
Dados Produzidos	Código do Cliente
Prioridade	Alta
Fluxo Principal	
Ação do Ator	Resposta do Sistema
O usuário, após autenticar-se, poderá dar manutenção (incluir, alterar, excluir, consultar) nos clientes	
Incluir, alterar, excluir ou consultar	Apresentar formulário padrão
Fluxo Alternativo	
Ação do Ator	Resposta do Sistema

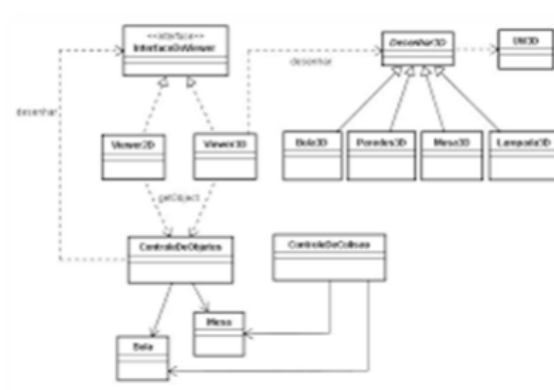
Fonte: O Autor

DIAGRAMA DE DOMÍNIO

O diagrama de domínio é uma variação do diagrama de classes e utiliza quase a mesma notação. Representa o domínio de dados a serem tratados e armazenados pelo sistema, ou seja, os elementos DAO do sistema. Garante uma visão sistêmica entre os componentes de interface e os componentes dedados do sistema.



ESPECIFICAÇÃO DE INTERFACE



Fonte da Imagem: Autor

A especificação de interfaces de componentes é a forma de descrever como os elementos que quiserem utilizar esses componentes:

Devem acessá-lo (sua assinatura, com os parâmetros e retornos oferecidos);

Quais os atributos que ele manipula;

Quais as funções disponíveis para uso.

Os componentes podem ser agrupados em pacotes, que reúnem componentes com funcionalidades e atributos complementares, dentro do conceito de família de componentes.

Esses pacotes vão apresentar as informações necessárias para que possam ser utilizados no conceito de arquitetura de componentes, através de suas funcionalidades e operações.

ATIVIDADE PROPOSTA



Fonte da Imagem: [//www.inmetro.com.br](http://www.inmetro.com.br)

Os modelos conceituais utilizados na UML garantem que os elementos de negócio, referentes ao sistema que está sendo desenvolvido, sejam completamente mapeados. Garantem também que haja uma migração consistente dos modelos conceituais de negócio para os modelos conceituais de sistemas.

Dessa maneira, conseguimos que nossos sistemas atendam a seus objetivos e sejam desenvolvidos, segundo o melhor conjunto de boas práticas.

Você acha que podemos fazer um paralelo entre esse conceito e a construção de um equipamento, atendendo às normas da ABNT e definições do INMETRO?

Questão 1:

UML é uma linguagem de modelagem de uso geral que pode ser usada com os principais métodos de objetos e de componentes, podendo ainda ser aplicada a todos os domínios de aplicação e plataformas de implementação;

Um relacionamento estendido entre casos de uso significa que o caso de uso base incorpora implicitamente, sob alguma condição, o comportamento de outro caso de uso.

Um relacionamento de inclusão entre casos de uso significa que o caso de uso base incorpora explicitamente o comportamento de outro caso de uso.

Relacionamentos de inclusão e extensão são representados pela mesma notação do relacionamento de dependência, com a seta apontada para o caso de uso base.

A alternativa completamente CORRETA é:

☐

A) I – II – III

☐

B) I – III – IV

☐

C) II – III – IV

☐

D) I – III – IV

☐

E) I – II – III -IV

☐☐☐

- ☐
- ☐

Justificativa

Questão 2:

Diagramas de casos de uso são compostos por atores, casos de uso e seus relacionamentos;

Diagramas de classes são compostos por classes e seus relacionamentos;

Diagramas de sequência são compostos por objetos e suas trocas de mensagens, ou seja, chamadas de métodos para outros objetos;

Considerando as sentenças acima, marque a opção correta:

☐

A) Somente as afirmativas I e III são verdadeiras.

☐

B) Somente as afirmativas II e III são verdadeiras.

☐

C) Somente as afirmativas I e II são verdadeiras.

☐

D) As afirmativas I, II e III são verdadeiras.

☐

E) Somente a alternativa I é verdadeira.

- ☐
- ☐
- ☐
- ☐
- ☐

Justificativa

Questão 3:

O diagrama UML, que apresenta um conjunto de objetos envolvidos em um cenário e a especificação das mensagens trocadas entre esses objetos ao longo de uma linha de tempo, é chamado de:

☐

A) Diagrama de mensagens

☐

B) Diagrama de sequência

☐

C) Diagrama de execução em linha de tempo

☐

D) Diagrama de estado

☐

E) Nenhuma das respostas anteriores

☐☐☐☐☐

Justificativa

Questão 4:

Em UML, são exemplos de diagramas responsáveis por representar o aspecto estrutural de um software:

☐

A) Diagrama de classes e diagrama de pacotes

☐

B) Diagrama de entidade relacionamento e diagrama de pacotes

☐

C) Diagramas de classes, diagrama de pacotes e use case

☐

D) Use case e diagrama de classes

☐

E) Nenhuma das respostas anteriores

☐☐☐☐☐

Justificativa

Questão 5:

A atividade de análise de requisitos procura descobrir o que os stakeholders de um projeto de sistema de software querem que o sistema faça. Para ajudar na comunicação com os usuários e clientes vários diagramas da UML podem ser utilizados.

Com relação à utilização dos diagramas da UML na atividade de análise de requisitos, analise se as sentenças a seguir são falsas ou verdadeiras e depois assinale a opção correta:

() Diagrama de classes desenhado a partir da perspectiva conceitual é uma boa maneira de construir um vocabulário rigoroso do domínio.

() Um diagrama de atividades é recomendado para exibir o fluxo de trabalho da organização, mostrando como o software e as atividades humanas interagem.

() Um diagrama de objetos é indicado para representar um conceito que tenha um ciclo de vida com vários estados e os eventos que mudam esses estados.

☐

A) F, V e F

☐

B) F, V e V

☐

C) V, F e F

☐

D) V, V e F

☐

E) F, F e V

☐☐☐☐☐

Justificativa

Questão 6:

A UML é uma famosa linguagem usada para análise e projeto orientado a objetos. O seu diagrama de sequência tem como características, EXCETO:

☐

A) Geralmente é construído a partir dos Casos de Uso.

☐

B) Usado para representar interações entre objetos de um cenário, realizadas através de operações ou métodos.

☐

C) Dá ênfase ao fluxo de controle de uma atividade para outra.

☐

D) Dá ênfase à ordenação temporal em que as mensagens são trocadas entre os objetos de um sistema.

☐

E) Pode ser facilmente transformado em um Diagrama de Colaboração.

☐☐☐☐☐

Justificativa

Glossário