Macoratti.net UML - Principais diagramas da linguagem

Uma grande parte dos desenvolvedores de software geralmente torcem o nariz para o trabalho de análise do projeto de software partindo diretamente para o trabalho de codificar, pois afinal, como dizem , isto é o que realmente importa.

Esta visão romântica e suicida, hoje em dia não tem futuro. Devido ao aumento da complexidade dos projetos de software, realizar um planejamento profundo do projeto é crucial. O cliente precisa entender o que o desenvolvedor esta fazendo e precisa ter condições de indicar alterações nas funcionalidades do projeto. Para isso é necessário um canal de comunicação onde a linguagem usada seja compreensível pela equipe de desenvolvimento e pelo cliente. A chave para este processo ser bem sucedido é organizar o processo de desenvolvimento de forma a envolver programadores, analistas e clientes no desenvolvimento do sistema usando uma linguagem que seja de fácil entendimento a todos.

Para isto é preciso adotar uma linguagem padrão que seja aceita e compreendida por toda a equipe de desenvolvimento e pelo cliente. A UML tem sido adotada como esta linguagem padrão.

A UML consiste de um certo número de elementos gráficos que se combinam para formar diagramas. Como a UML é uma linguagem, ela possui regras para combinar estes elementos nos diversos diagramas.

Nota: Para saber mais sobre UML acompanhe os artigos : UML - Unified Modeling Language e UML - Modelando sistemas - Casos de Uso

O objetivo dos diagramas é apresentar múltiplas visões do sistema sendo que este conjunto de múltiplas visões é chamado de modelo. Podemos dizer que um modelo UML pode ser visto como um conjunto de diagramas que podem ser examinados e modificados a fim de compreender e desenvolver um sistema de software.

Um modelo UML descreve o que o sistema fará mas não diz nada como implementar o sistema.

Vejamos a seguir os diagramas da UML:

Uma classe é uma categoria ou grupo de elementos(coisas) que possuem o mesmo atributo e comportamento. Nesta linha de raciocínio a classe máquina de lavar roupa conterá elementos que possuem atributos como nome, modelo, número de série e

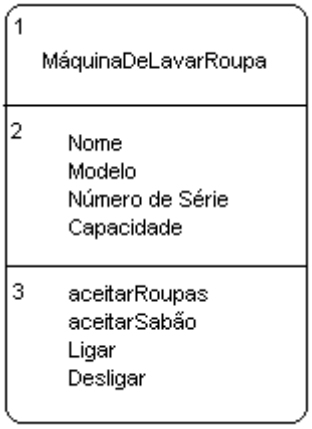
capacidade e deverá apresentar comportamento que inclui as operações como: "aceitar roupas", "aceitar sabão" , "ligar", "desligar".

Vamos aplicar a notação UML através dos seus diagramas a uma máquina de lavar roupa que é um objeto da classe máquina de lavar roupa.

1- Diagrama de Classes:

A notação UML que captura os atributos e comportamentos de um máquina de lavar roupa é mostrada a seguir e chama-se diagrama de classe. O retângulo é o desenho que representa a classe e ele esta dividido em três áreas :

1- A parte superior contém o nome da classe
2- A parte do meio contém os atributos da classe
3- A parte de baixo contém as operações(métodos) da classe



O diagrama de classes permite aos analistas usarem uma notação de fácil compreensão
pelo cliente estimulando-os desta a forma a revelar detalhes importantes sobre o problema
que necessita ser resolvido.

2- Diagrama de Objeto

Um objeto a é uma instância de uma classe - um elemento específico que possui valores dos atributos da classe. No caso da lavadora poderíamos ter os seguintes valores:

Nome	**Westhinghouse**
Modelo	**Master**
Número de Série	**MLV05245**
Capacidade	**8 Kg**

A UML representa um objeto usando o diagrama da figura abaixo. Um retângulo representa o objeto, e o nome é sublinhado. Podemos ainda ter duas variações:

a- O nome da instância específica do objeto, dois pontos seguido do nome da classe

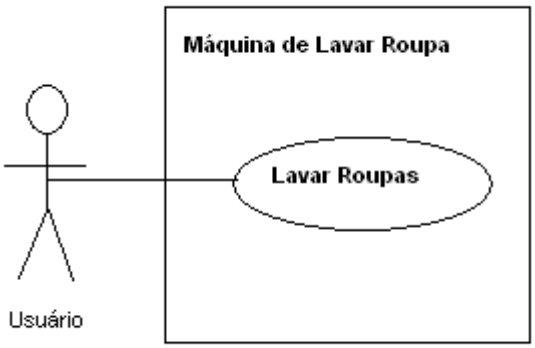
b- dois pontos seguido do nome da classe. Neste caso temos um objeto anônimo, i.e, não é fornecido um nome para o objeto mas mostramos a classe a qual ele pertence.

1- nome do objeto: nome da classe 2- Objeto Anônimo

3- Diagrama de caso de uso

Um caso de uso é a descrição do comportamento do sistema do ponto de vista do usuário. Para os desenvolvedores os casos de uso são uma ferramenta muito útil pois ele pode ser considerado uma técnica do tipo tentativa e erro para obter os requisitos do sistema a partir do visão do cliente.

A seguir temos a representação UML para o caso de uso **Lavar roupas**:



- A figura que representa o usuário é chamado ator. O ator é a entidade que inicia o caso de uso e pode ser uma pessoa ou outro sistema.

- A elipse representa o caso de uso , no exemplo: **Lavar Roupas**.

Observe que o caso de uso esta no interior de um retângulo que representa o sistema e o ator esta fora do retângulo.

Caso de uso: Lavar Roupas

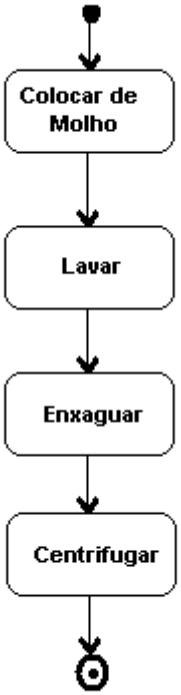
4- Diagrama de estado

Em um determinado momento um objeto possui um estado particular. No caso de uma pessoa ela pode ser: recém-nascida, criança, adolescente ou adulto. Uma máquina de lavar pode possuir os seguintes estados: colocar em molho, lavar , enxaguar e centrifugar ou desligar.

A representação UML que captura este comportamento chama-se diagrama de estado e esta representado abaixo:

O símbolo no topo da figura representa o inicio do estado e o símbolo na base da figura representa
o fim do estado.

As transições de um estado para outro nem sempre são lineares como representado para este exemplo.



4- Diagrama de seqüência

O diagrama de classe e o diagrama de objeto representam uma informação estática. Em um sistema funcional , no entanto, os objetos interagem uns com os outros, e , estas interações ocorrem a todo momento. O diagrama de seqüência UML é usado para representar estas interações e é composto basicamente por objetos e mensagens.

Um diagrama de seqüência descreve a maneira como os objetos colaboram em algum comportamento ao longo do tempo e registra o comportamento de um único caso de uso. Esse diagrama é simples e lógico, com o objetivo de óbvios a seqüência e o fluxo de controle.

Continuando com o exemplo da máquina de lavar roupa podemos identificar os seguintes componentes na máquina: *(Podemos considerar estes componentes como objetos)*

- **O Timer**
- **A Bomba d'agua (que introduz a água na máquina)**
- **O Tambor (onde são colocadas as roupas)**

Vejamos então o que acontece quando invocamos o caso de uso "**Lavar Roupas**", assumindo que já tenhamos incluído as roupas na máquina, o sabão e a mesma tenha sido ligada. Podemos descrever a seguinte seqüência de operações: (São apenas considerações)

1. No ínicio da operação "Colocar de Molho", a água entra no Tambor pela Bomba d'agua;
2. O Tambor permanece estacionário por 5 minutos aproximadamente;

3. No final da operação "Colocar de Molho" a água para de entrar no Tambor;
4. **No início da operação "Lavar" o Tambor inicia a rotação alternada por 15 minutos;**
5. **No final da operação "Lavar" o Tambor joga a água com sabão para fora;**
6. **O Tambor para a sua rotação;**
7. No início da operação "Enxaguar" a água começa a entrar novamente no Tambor
8. O tambor inicia a rotação alternada;
9. Depois de 15 minutos a água para de entrar no Tambor;
10. No final da operação o Tambor joga a água para fora;
11. O Tambor pára de efetuar a rotação alternada;
12. No início da operação "Centrifugar" o Tambor inicia a rotação continua no sentido horário por 15 minutos;
13. No final da operação "Centrifugar" o Tambor para de efetuar a rotação;
14. A lavagem de roupas esta completa

Assumindo que o timer, a bomba d'agua e o tambor são objetos e que cada objeto possui uma ou mais operações podemos perceber que estes objetos trabalham em conjunto enviando mensagens uns para os outros para que a tarefa seja realizada com êxito.

A seguir vamos definir as operações nas quais atuam cada objeto :

a- O Timer

- tempo de molho (Colocar de Molho);
- tempo de lavagem (Lavar);
- tempo de enxugamento (Enxugar);
- tempo de centrífuga (Centrifugar);

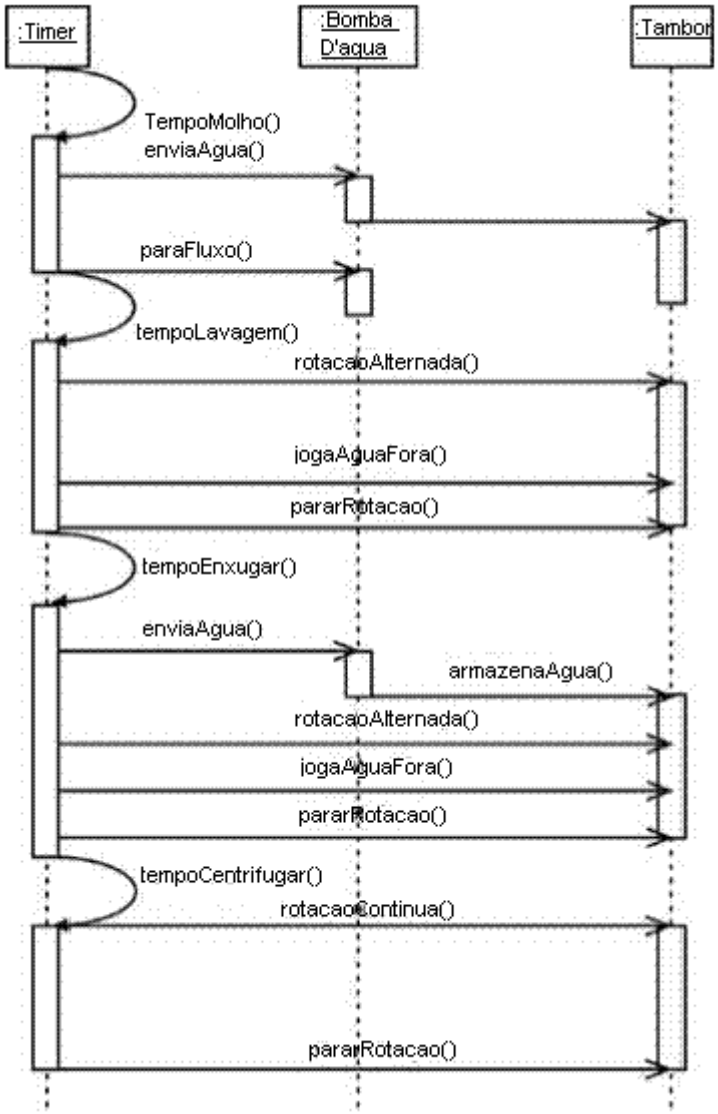
b- A bomba d'agua

- Iniciar o fluxo de água;
- Parar o fluxo de água;

c- O Tambor

- Armazenar água;
- Rotacionar alternadamente;
- Rotacionar no sentido horário;
- Parar de rotacionar;
- Jogar água para fora;

A seguir temos o diagrama de seqüência que captura as mensagens entre os objetos : **Timer, Bomba d'gua e o Tambor**. Cada seta representa uma mensagem que é enviada de um objeto para outro. O tempo transcorre do topo para a base do diagrama, sendo exibida em seqüência as mensagens trocadas ao longo do tempo.



- O objeto **Timer** envia mensagens para ele mesmo

5- Diagrama de Atividades

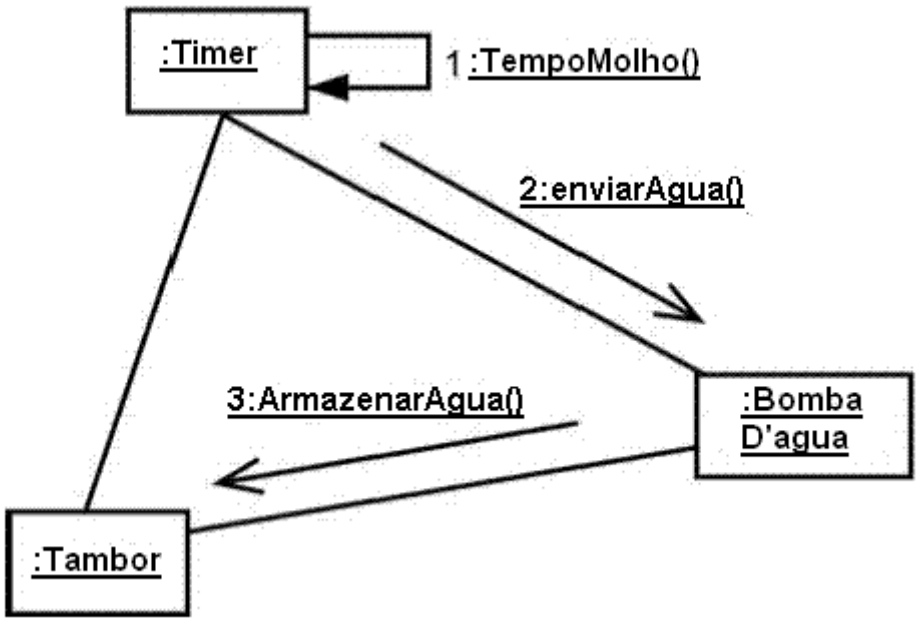
O objetivo do diagrama de atividades é mostrar o fluxo de atividades em um único processo. O diagrama mostra como um atividade depende uma da outra.

A seguir temos o diagrama de atividades representando as atividades do objeto **Tambor** da máquina de lavar descrita nos passos **4 a 6**:

As atividades que ocorrem dentro do caso de uso ou no comportamento do objeto ocorrem em uma seqüência conforme descritos no item anterior.

6- Diagrama de Comunicação

Os elementos do sistema trabalham em conjunto para cumprir os objetos do sistema e um linguagem de modelagem precisa poder representar esta característica. O diagrama de comunicação procura capturar este comportamento. O diagrama mostrado a seguir procura mostrar as mensagens trocadas entre o objeto timer, a bomba d'agua e o tambor. O digrama mostra a ordem das mensagens usando uma numeração para indicar a sua ordem.



Os diagramas de seqüência e de comunicação mostram as interações entre os objetos, por este motivo a UML se refere as estes diagramas como diagramas de interação.

7- Diagrama de Componentes

"Os diagramas de componentes mostram os elementos reutilizáveis de software e sua interdependência. Um componente é formado por um conjunto de classes que se encontram nele implementadas. Um componente, assim como as classes que ele possui, dependem funcionalmente das classes de outro componente. O diagrama de componentes mostra esta dependência. No diagrama de componentes também é

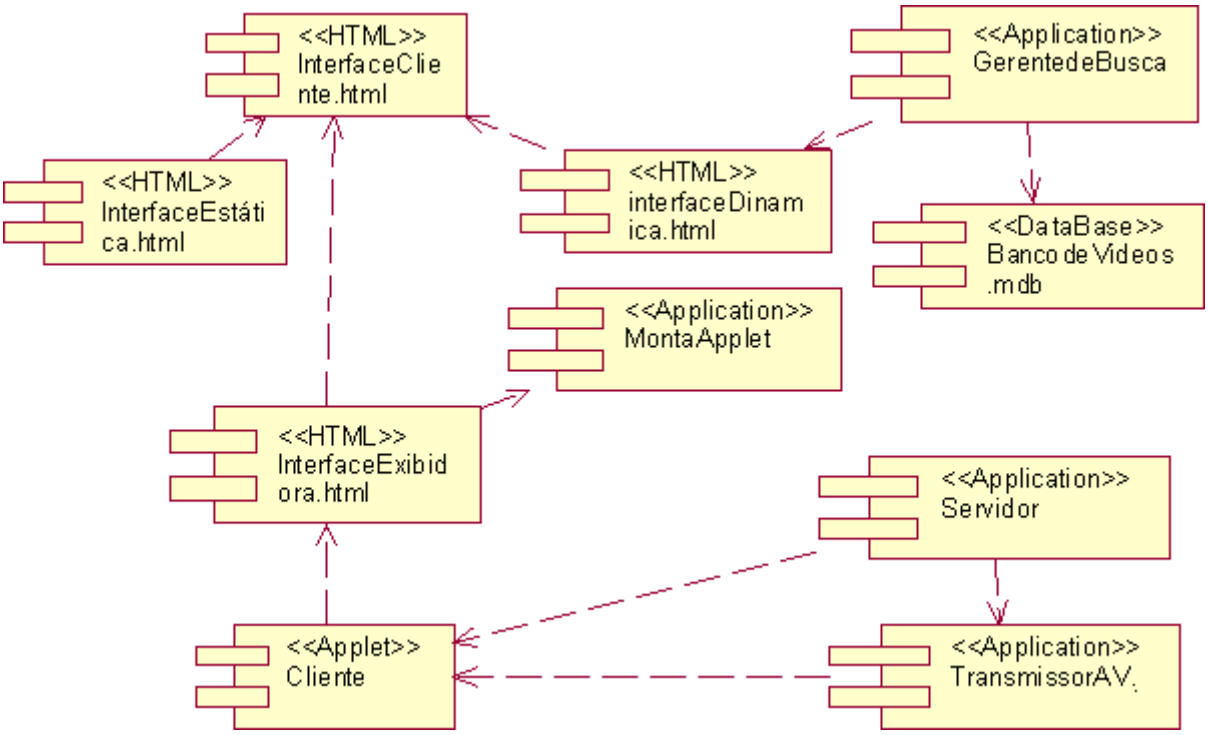
possível mostrar a configuração de um sistema de software mostrando, graficamente, a dependência entre os diversos arquivos que compõem o sistema." (VOXXEL).

As relações de dependência são usadas nos diagramas de componentes para indicar os diversos arquivos que compõe o sistema.

A UML reconhece cinco estereótipos de componentes:

- **Um executável:** Um componente que pode ser executado (um programa).
- **Uma biblioteca:** Uma biblioteca de classes ou funções, dinâmica ou estática.
- **Um tabela:** Uma tabela de um banco de dados.
- **Um documento:** Uma parte da documentação (texto livre, diagramas, documentos de ajuda, etc.)
- **Um arquivo:** Outros arquivos, geralmente, se trata de um arquivo de código fonte, mas pode ser também um arquivo de dados, um ``script'' ou outros arquivos.

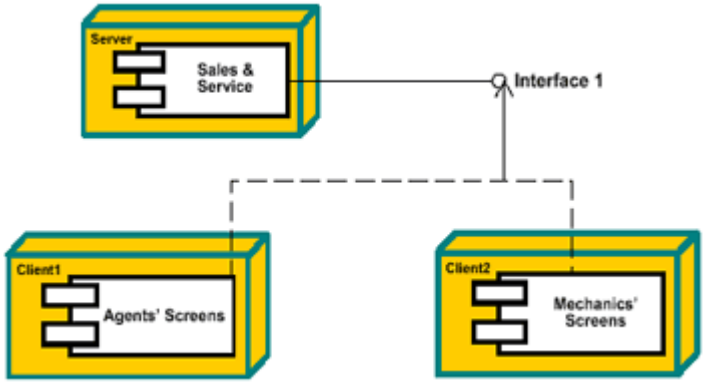
Exemplo de um diagrama de componente exibindo os componentes para um sistema de locadora na web :



8- Diagrama de Distribuição

"Os diagramas de distribuição mostram a distribuição de hardware do sistema, identificando os servidores como nós do diagrama e a rede que relaciona os nós. Os componentes de software vão estar mapeados nestes nós". (VOXXEL)

Ele permite apresentar a topologia de uma rede de máquinas'' e qual processo (um componente executável) cada máquina'' vai rodar. As maquinas'' são chamadas de *nos*. Um nó apresenta uma fonte computacional, sendo normalmente um processador com alguma memória.



A UML é uma importante ferramenta de modelagem , através dela podemos obter diversas visões de um sistema de software.

Neste artigo procurei apresentar os principais diagramas usados na UML de forma a que você tenha uma visão geral de sua utilização.

Até o próximo artigo .NET...

Referências:

- Seção VB .NET do Site Macoratti.net
- Super DVD .NET - A sua porta de entrada na plataforma .NET
- Super DVD Vídeo Aulas - Vídeo Aula sobre VB .NET, ASP .NET e C#
- Seção C# do site Macoratti.net
- Super DVD C#
- Super DVD Visual Basic
- Curso Básico VB .NET - Vídeo Aulas
- Curso C# Básico - Vídeo Aulas
- UML - Macoratti .NET
- UML - Diagrama de classes - Macoratti.net
- UML - Modelando sistemas - Casos de Uso
- UML - Unified Modeling Language e Visual Modeler.
- VB.NET - Revisitando conceitos OOP
- VB .NET - Conceitos OOP.
- VB .NET - Primeiros passos - Conceitos - VI.
- VB .NET - Finalmente uma linguagem orientada a objetos.
- OOP noções para iniciantes
- O que significa "orientação a objetos" ?
- Conceitos sobre Projetos - Decomposição de software.

- UML - Unified Modeling Language - Macoratti.net

José Carlos Macoratti