**Texto base**

**12**

**Diagrama de Classe, Diagrama de Sequência e Diagrama de Atividades**

Prof. João de Deus Freire Junior

***Resumo***

*A UML é a linguagem de modelagem unificada. Trata-se de uma linguagem de modelagem de soluções e sistemas. Ela auxilia na compreensão da solução sistêmica na perspectiva interna estática e iterativa do sistema e na perspectiva do cliente. Ela é formada por diagramas que possibilitam essa visão completa de solução. Nesta aula, será exibido os Diagrama de Classe, Diagrama de Sequência e Diagrama de Atividades que são três dos principais diagramas da UML .*

**1.1. Introdução**

Como posso projetar uma solução sistêmica na perspectiva interna do sistema? Como posso construir uma representação gráfica das estruturas do sistema? O que são os diagramas de classe, atividades e sequência? Quais são os principais elementos desses diagramas? Todas essas perguntas serão respondidas nesta aula. O entendimento dos elementos, usos e formato desses diagramas da UML possibilitará aos analistas uma poderosa ferramenta de modelagem para expressar a visão interna do sistema. Desta forma, podemos mitigar e diminuir significativamente as falhas e fracassos no desenvolvimentos de projetos de soluções tecnológicas.

**1.2. Principais Diagramas da UML**

A UML 2.0 fornece 13 diferentes diagramas para uso na modelagem de software. Os principais diagramas são:

* De classe;
* Distribuição;
* Caso de uso;
* Sequência;
* Comunicação;
* Atividade;
* Estado.

De acordo com Sommerville (2011), através desses diagramas podemos modelar os sistemas em perspectivas diferentes. Por exemplo:

1. Uma perspectiva externa, em que você modela o contexto ou o ambiente do sistema.
2. Uma perspectiva de interação, em que você modela as interações entre um sistema e seu ambiente, ou entre os componentes de um sistema.
3. Uma perspectiva estrutural, em que você modela a organização de um sistema ou a estrutura dos dados processados pelo sistema.
4. Uma perspectiva comportamental, em que você modela o comportamento dinâmico do sistema e como ele reage aos eventos.

Sommerville (2011) afirma também que os cincos principais diagramas da UML podem representar a essência de um sistema. Eles os fazem da seguinte forma:

1. Diagramas de atividades, que mostram as atividades envolvidas em um processo ou no processamento de dados.
2. Diagramas de casos de uso, que mostram as interações entre um sistema e seu ambiente.
3. Diagramas de sequência, que mostram as interações entre os atores e o sistema, e entre os componentes do sistema.
4. Diagramas de classe, que mostram as classes de objeto no sistema e as associações entre elas.
5. Diagramas de estado, que mostram como o sistema reage aos eventos internos e externos.

**1.3. Diagrama de Classe**

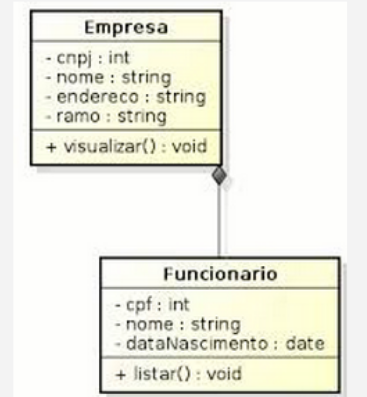
**1.3.1. Definições Gerais**

O diagrama de classe é o mais utilizado da UML. Ele permite a visualização das classes utilizadas pelo sistema e como elas se relacionam. Ele apresenta uma visão estática da organização das classes, definindo sua estrutura lógica. Ele pode ser usado para definir um modelo lógico de um banco de dados (mas, uma classe não corresponde a uma tabela no BD) (Pressman, 2011).

Ele é formado por classes que são compostas por: nome da classe, lista de atributos e lista de operações (métodos).

**1.3.1. As classes**

As classes no diagrama de classes são compostas por: nome da classe, lista de atributos e lista de operações (métodos).



**Figura 1.1. Classes. Fonte: PRESSMAN, R. S. (2011)**

**1.3.2. Associações**

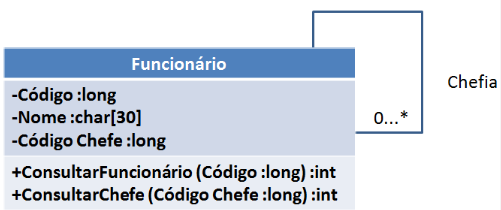
As associações são compartilhamento de informações e colaboração para execução dos processos do sistema; descrevem o vínculo entre os objetos de uma ou mais classes. (Pressman, 2011)

As associações exibem a navegabilidade (caminho da informação) e multiplicidade (números mínimo e máximo de objetos envolvidos);

**1.3.3. Tipos de Associações**

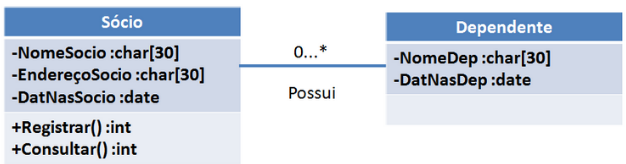
* **Unária:** relacionamento de objeto de uma classe com objetos da mesma

classe.

****

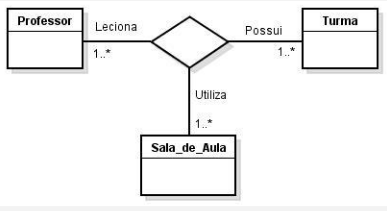
**Figura 1.2. Associação Unária. Fonte: PRESSMAN, R. S. (2011)**

* **Binária:** quando há relacionamento de objetos de duas classes.

****

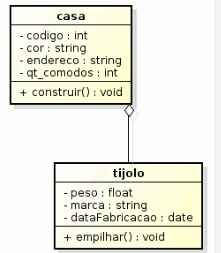
**Figura 1.3. Associação Binária. Fonte: PRESSMAN, R. S. (2011)**

* **Ternária:** conectam objetos de mais de duas classes.



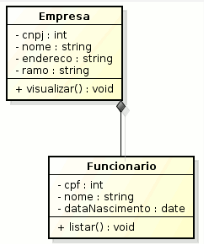
**Figura 1.4. Associação Ternária. Fonte: PRESSMAN, R. S. (2011)**

* **Agregação:** demonstra relação todo/parte entre objetos associados**.**



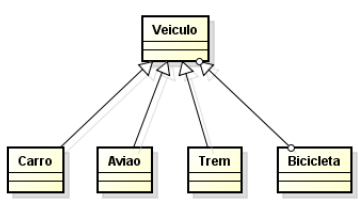
**Figura 1.5. Associação Agregação. Fonte: PRESSMAN, R. S. (2011)**

* **Composição:** variação da agregação, com vínculo mais forte.



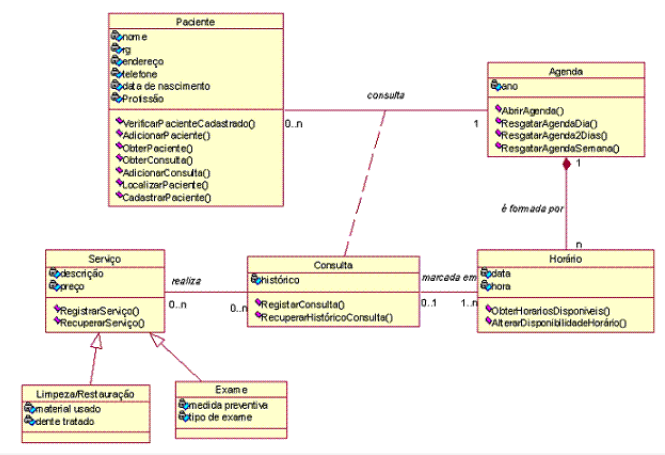
**Figura 1.6. Associação Composição. Fonte: PRESSMAN, R. S. (2011)**

* **Especialização / Generalização:** identifica classes-mãe e classes-filhas, com ocorrência de herança.

****

**Figura 1.7. Associação Especialização / Generalização. Fonte: PRESSMAN, R. S. (2011)**

**1.3.4. Exemplo de Diagrama de Classe**

****

**Figura 1.8. Exemplo de Caso de Uso. Fonte: PRESSMAN, R. S. (2011)**

**1.4. Diagrama de Sequência**

**1.4.1. Definições Gerais**

O diagrama de sequência procura determinar a sequência de eventos que ocorrem em um processo. Ele identifica quais métodos devem ser disparados entre os atores e objetos envolvidos e em qual ordem. (Pressman, 2011)

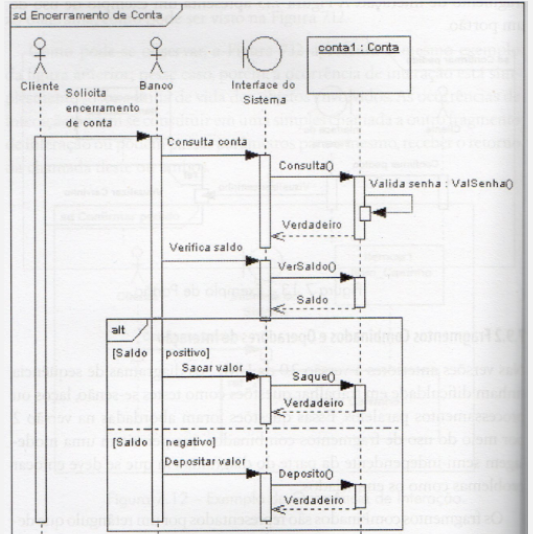
O diagrama de sequência baseia-se no Diagrama de UC: normalmente há um diagrama de sequência para cada UC (cada UC é um processo disparado pelo Ator). Ele também depende do Diagrama de Classes: classes dos objetos declarados, além dos métodos. (Pressman, 2011)

**1.4.2. Elementos**

De acordo com Pressman (2011), o diagrama de sequência é composto pelos seguintes elementos:

* **Atores:** são os mesmos dos Diagramas UC; aqui, possuem “linha de vida”;
* **Objetos:** representam as instâncias das classes do processo ilustrado no Diagrama de Sequência; um objeto pode existir desde o início do processo ou ser criado durante a execução do mesmo;
* **Linha de Vida:** representa o tempo que um objeto existe durante o processo. Representada por uma linha fina e tracejada, é interrompida com um “X” quando o objeto é destruído;
* **Foco de Controle (Ativação):** indica os períodos que um objeto participa ativamente do processo. Na linha da vida, é representado por uma linha grossa.
* **Mensagens (Estímulos):** demonstram a ocorrência de eventos, que forçam a chamada de um método em um dos objetos envolvidos no processo. Representadas por uma seta entre dois componentes (quem envia para quem recebe). O texto contido nelas indicam o evento ocorrido e o método chamado.
* **Mensagens de Retorno:** identifica a resposta a uma mensagem para o objeto que a chamou; pode retornar informações específicas ou flag de sucesso ou não.
* **Autochamadas:** mensagens que partem da linha de vida do objeto e atingem a linha de vida do mesmo objeto.
* **Condições de Guarda:** estabelecem uma regra ou condição para que uma mensagem possa ser disparada.

**1.4.3. Exemplo de Diagrama de Sequência**

****

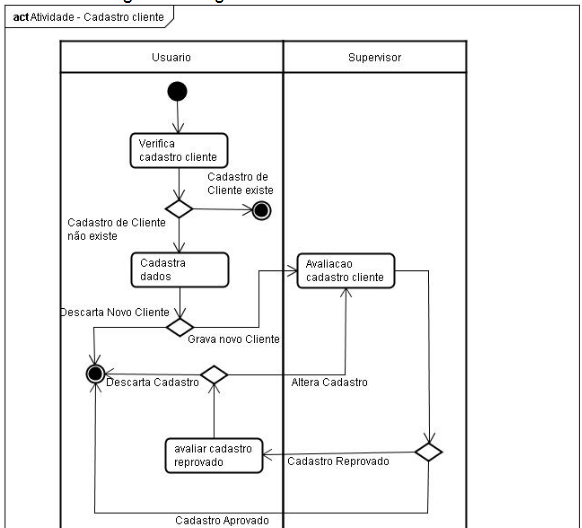
**Figura 1.9. Exemplo de Diagrama de Sequência. Fonte: PRESSMAN, R. S. (2011)**

**1.5. Diagrama de Atividades**

**1.5.1. Definições Gerais**

O diagrama de atividades dá maior ênfase ao nível de algoritmo (um dos mais detalhistas). Ele é utilizado para modelar atividades, que podem ser um método, algoritmo ou um processo completo. Ele é semelhante aos antigos “fluxogramas”. Ele possui partição de atividades para representar o fluxo de um processo que envolve diversos atores (ou departamentos, setores, etc...). (Pressman, 2011)

**1.5.2. Exemplo de Diagrama de Atividades**

****

**Figura 1.10. Exemplo de Diagrama de Atividades. Fonte: PRESSMAN, R. S. (2011)**

**1.6. Você quer ler?**

Segue uma indicação de estudo complementar. Trata-se de um breve guia da linguagem UML.

* DEVMEDIA. Conhecendo os diagramas da UML através de um estudo de caso. 2011. Disponível em: <https://www.devmedia.com.br/conhecendo-os-diagramas-da-uml-atraves-de-um-estudo-de-caso/22550>. Acesso em: 20 dez. 2019.

**1.7. Referências**

* PRESSMAN, R. S.(2011) Engenharia de Software: uma abordagem profissional. 7.ed. Porto Alegre: Bookman, 2016.
* Figuras 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 1.10: PRESSMAN, R. S.(2011)