

Tecnologia em Análise e Desenvolvimento de Sistema

Engenharia de Software

Autores:

Wesley R. Prado

Graduando em Física - UFES

Graduando em Análise e Desenvolvimento de Sistemas - FMU

Laísa Cominotti Rossim Prado

Mestre em Matemática - IFES

Graduada em Matemática - IFES

Graduada em Engenharia Civil - Pitágoras

Introdução

A tarefa de desenvolver um software não é algo simples, exige conhecimentos específicos que ultrapassam as barreiras das linguagens de programação e conhecimentos práticos de bancos de dados. É de suma importância que antes de qualquer tentativa haja um análise de requisitos, que comumente faz-se uso de notas autoadesivas (post-its) que geram um grupo de informações, tornando o entendimento e a organização mais legíveis, vale dizer que essa prática é conhecida como métodos ágeis e que usada em outros ramos que exige planejamento, estratégico, inovação e etc . Após toda essa documentação levantada gera-se diversos gráficos e diagramas.

Feito o levantamento da documentação técnica, dá-se início a elaboração de diversas representações gráficas (diagramas) que auxiliaram ou melhor guiaram toda a fase de desenvolvimento da aplicação. A UML (Unified Modeling Language) é a ferramenta de linguagem mais utilizada para realizar a criação de muitos diagramas. Caro leitor, quando falamos em UML não estamos falando de uma linguagem de programação como Java ou Python, mas sim sobre uma linguagem gráfica composta por uma série de símbolos com significados bem específicos.(Oliveira, 2020)

A partir do conceito mencionado, vamos entender como surgiu a UML que é muito utilizada e vamos ver a importância dessa ferramenta. E ainda, veremos elementos que fazem parte de uma solução que utiliza a orientação a objeto que se relaciona com um tipo de diagrama UML.

UML - Unified Modeling Language

As linguagens com orientação a objetos foram suscitadas a partir de 1970, contudo a formação da UML iniciou-se sua produção em outubro de 1994 e foi lançada em outubro de 1995. A linguagem de modelagem UML é uma linguagem visual que tem por função especificar, construir e documentar dispositivos de sistemas.

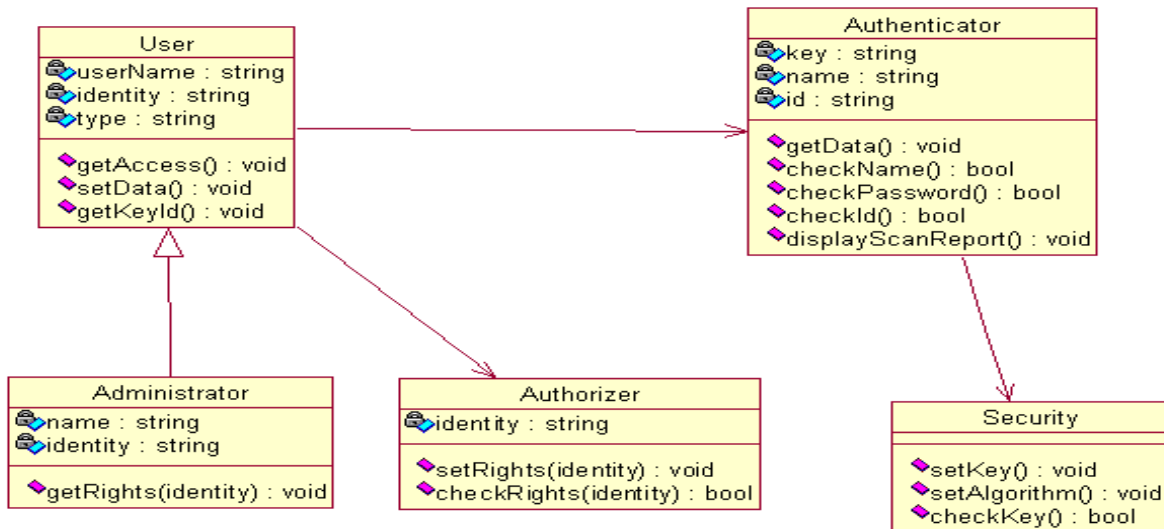
A UML é um padrão diagramático que utiliza de apresentação figuras com possui algum texto que se relaciona a software, principalmente software de orientação ao objeto. Existem três maneiras de utilizar a UML, vamos abordar algumas delas e suas funcionalidades.

A UML pode ser usada como **rascunho** que tem por principal objetivo visualizar a ideia do sistema que tem que ser desenhado. Neste momento não se tem a preocupação de modelar os diagramas de maneira completa e formal. Pelo contrário, a ideia é desenhar ou modelar diagramas incompletos e informais que muitas vezes podem ser desenhados a mão. Essa fase permite explorar partes difíceis do problema ou espaço de soluções, explorando o poder dessa linguagem que é utilizada em todo o mundo.

A outra maneira já é mais formal que é conhecida como planta de software, ou seja, os diagramas do projeto serão mais detalhados do que o anterior e terão um nível de modelagem padronizada e são utilizados para dois pontos os quais englobam engenharia reversa e geração de código. Para a engenharia reversa, quando utilizamos tal ferramenta nos ajuda a visualizar melhor e entender um determinado código, ou melhor, lê um código fonte ou binário gerando assim diagramas UML de pacotes, de classes e de sequência, por sua vez, permite que o leitor entenda os elementos, estruturas e outras colaborações. Quanto geração de códigos, vale pontuar que antes da programação, os diagramas podem fornecer diretrizes para geração de um código(por exemplo, em Java ou C#) e é comum que sejam usados para uma parte do código e outra parte seja preenchida por um desenvolvedor que seja o codificador do sistema, nessa parte pode-se utilizar ou aplicar alguns rascunhos.

Ainda temos a UML como linguagem de programação, mas que nós usemos de discernimento, pois quando falamos de tal maneira, não estamos falando que a UML pode ser uma linguagem de programação como Java, Python, C# e etc, pois o que queremos dizer, é que nesse tópico a UML é utilizada para desenhar um sistema inteiro e a partir desse desenho ou projeto, ele pode ser executável, ou seja, executa a linguagem ou código por trás dele, a parte não visível. Ou seja, nessa etapa, é necessário utilizar uma boa lógica, como exige uma maior complexidade no desenvolvimento. (LARMAN, Craig)

A imagem abaixo exemplifica como seria a UML como linguagem de programação executável.



Fonte da imagem:

https://www.researchgate.net/figure/Class-Diagram-for-MDA-Authentication-using-Executable-UML_fig2_50234963 , consultado às 1h17 (horário de Brasília), 29/05/2021.

Como vimos até aqui, a UML é uma ferramenta que ajuda a visualizar um sistema, tornando o projeto mais preciso, como também ela pode ser um sistema executável. Podemos concluir até o momento, que todos os envolvidos em desenvolvimento de software devem possuir conhecimento sólido nessa ferramenta tão importante.

Perspectiva de Aplicação da UML

A descrição de tipos de esboço de diagramas podem ser feitas pela UML. Tal linguagem não prioriza uma perspectiva apenas o conceito de modelagem, ou seja, ela pode pontuar conceitos do mundo ou de classes de software que são produzidos em Java, por exemplo. Vamos ver a perspectiva de aplicabilidade da UML.

A primeira perspectiva que vamos destacar é a conceitual que envolve a descrição do mundo real ou de um domínio de interesse. Nessa perspectiva, o foco costuma ser o cliente e tudo o que está no estudo do desenvolvimento.

A segunda perspectiva é a de especificação, nessa etapa vamos trazer um nível maior da abstração, pois irá descrever interfaces da arquitetura, nos principais métodos. Mas nesse caso, não há a implementação, ou seja, certas coisas não são mostradas, pois para quem essa perspectiva é mostrada não irá interessar, como exemplo o gerente do projeto não precisa saber de detalhes do desenvolvimento, por isso não iremos colocar nessa etapa.(LARMAN, Craig)

A terceira perspectiva é de implementação

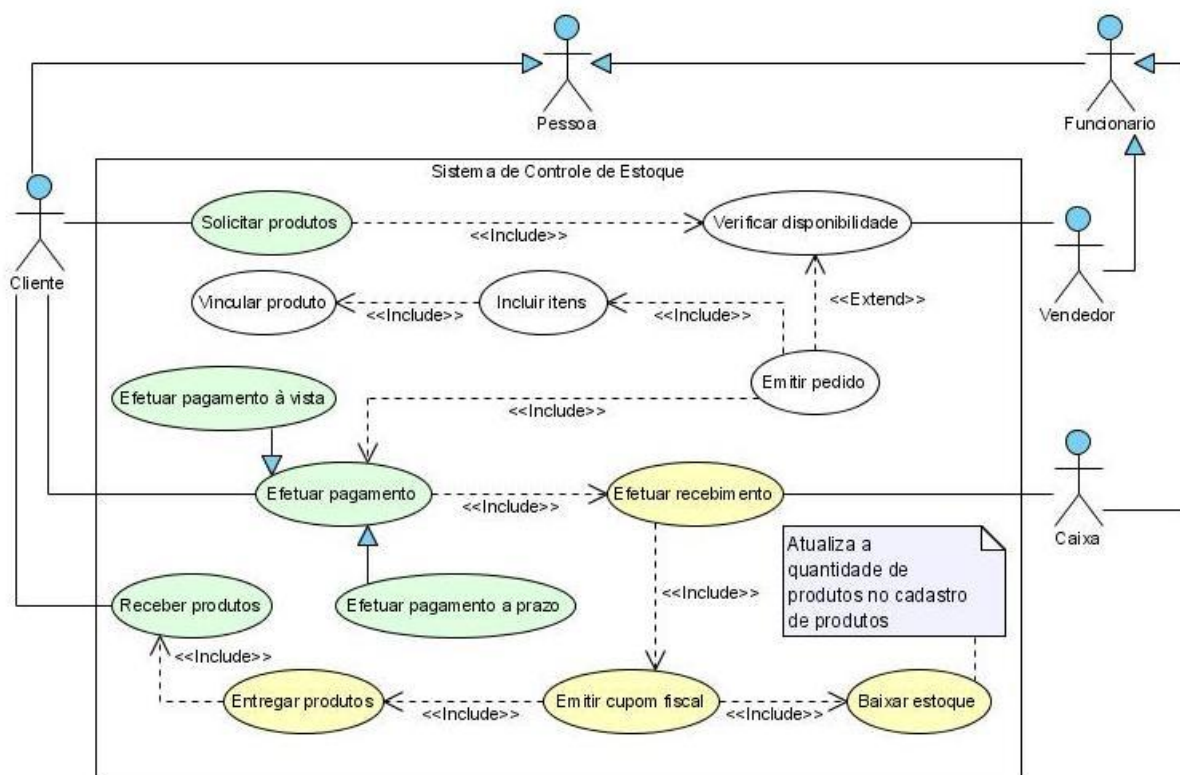
Os diagramas nesse momento auxiliam na modelagem da arquitetura física de um sistema. É possível ter uma visão dos relacionamentos entre os componentes de software e hardware no sistema e na distribuição física do processamento. Durante a fase do desenvolvimento da implementação estrutura a organização física dos elementos computacionais em um sistema distribuído, os artefatos que estão armazenados em cada elemento.(IBM).

Como vimos até aqui, a ideia principal da aplicação da UML é facilitar o entendimento de um projeto, como também até proporciona uma facilidade na execução de um determinado processo. Contudo, antes de concluirmos, vamos ver alguns diagramas principais da UML e uma breve descrição.

Diagramas da UML

Diagrama de Caso de Uso: a principal função desse diagrama é documentar tudo o que o sistema realiza de uma maneira que seja possível o usuário identificar cada ponto. Ou seja, trata das funcionalidades e das descrições de um sistema criando uma relação com o usuário desse mesmo. (DevMedia)

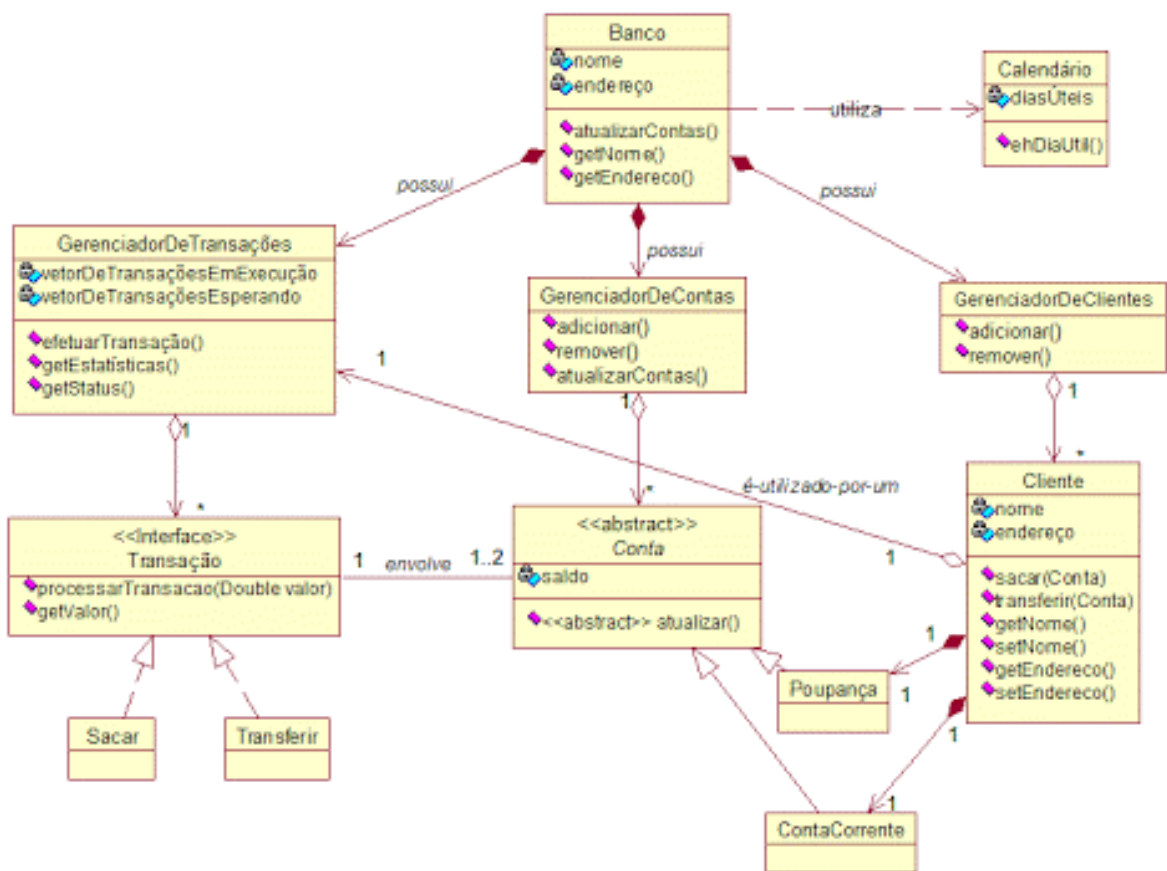
Abaixo vamos ver um exemplo desse diagrama



Fonte: <https://dtic.tjpr.jus.br/widget/wiki> , pesquisa às 00h34 em 30/05/2021.

Como podemos notar, esse diagrama descreve claramente todas as funcionalidades de um processo e inclui a relação com os usuários desse sistema.

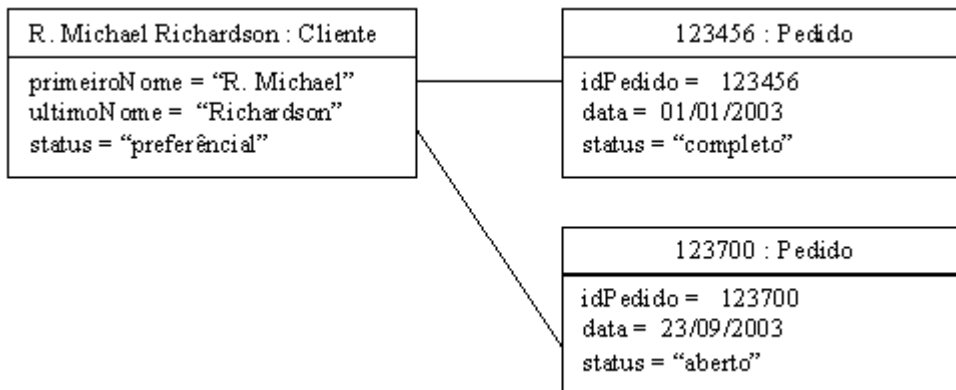
Diagrama de Classes: esse diagrama especifica elementos de um sistema e como tais componentes se interligam tendo por base uma visão estrutural. Ele se assemelha a outros diagramas, possuindo caixas que se encontram ligadas e conectadas por meio de setas. Para entender um modelo de classes é importante entender a função das caixas e os compartimentos das caixas. (ateomomento) Abaixo segue um modelo:



Fonte da imagem:

http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/classes/images/especificacao_exemplo_banco.GIF , consulta realizada em: 30/05/2021 às 7h41.

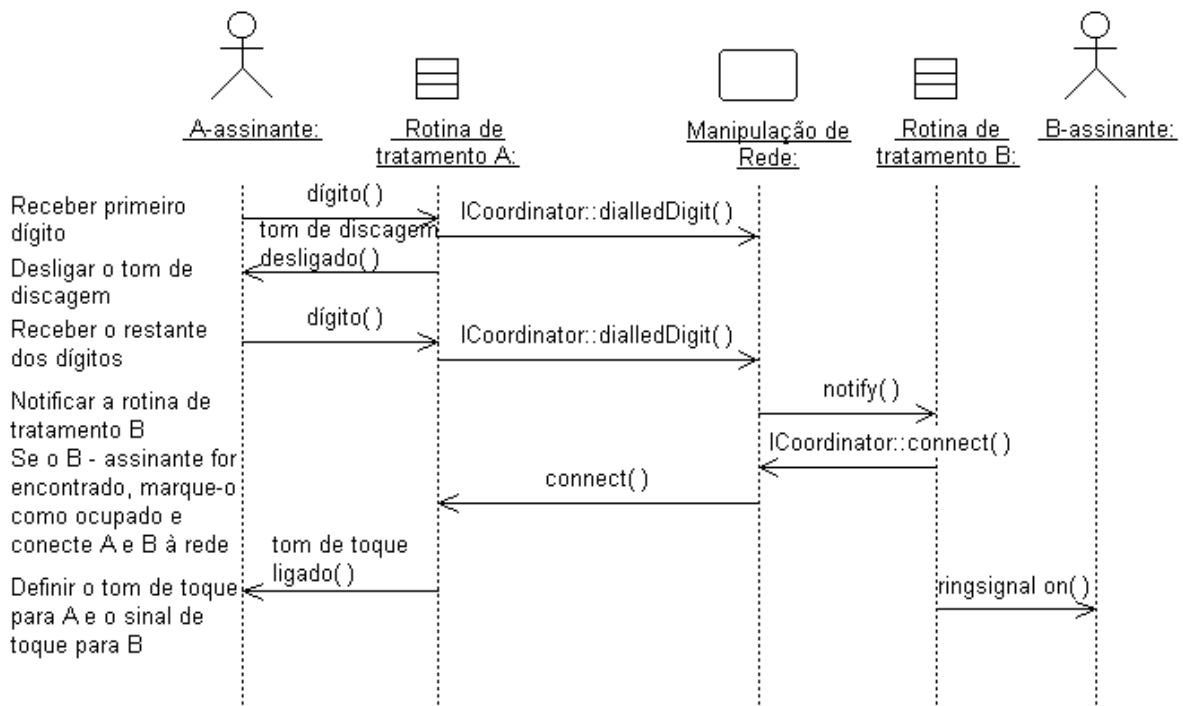
Diagrama de objetos: nesse modelo, as instâncias de classes são modeladas, ou seja, o diagrama de objetos evidencia um conjunto e seus no tempo. São de suma importância para construir os aspectos estáticos do sistema. Naturalmente, são compostos por: objetos e vínculos. Observemos o exemplo abaixo:



Fonte: https://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/objetos/diagrama_de_objetos.htm , consulta em 30/05/2021 às 7h50.

Como podemos observar de maneira simples, o diagrama mostra o objeto que o é o cliente R. Michael Richardson, e podemos notar que ele está associado a ambos os objetos 123456 e 123700 da classe Pedido. O diagrama de objetos é um retrato do sistema em um determinado ponto ou momento, mostrando uma visão estática. (A. Mendes)

Diagrama de Sequência: Esse diagrama representa a sequência do que ocorre no desenvolvimento, sendo orientado por tempo e colaboração entre os objetos. Vejamos o exemplo:



Fonte:

https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/guidances/guidelines/sequence_diagram_AFA76EBB.html consulta 30/05/2021 às 8h00.

É comum usar esse diagrama para ilustrar a realização de caso de uso, ou seja, deixar visível a interação entre os objetos que possibilitam a execução do comportamento de todo ou parte de um caso de uso.

A importância dos diagramas é de um grau alto para os designers, pois esses diagramas esclarecem as funções dos objetos em um fluxo, fornecendo assim a entrada básica para determinar a responsabilidade de classe e interfaces.(UFPE)

Conclusões:

Podemos observar que UML é uma ferramenta essencial para o desenvolvimento de um software de qualidade, pois permite a visualização por meio de diagramas dos processos envolvidos.

Essa visualização permite uma melhor comunicação entre as equipes envolvidas e até mesmo o cliente. Vale lembrar, que para cada etapa do projeto ou do desenvolvimento, existe um diagrama a ser utilizado, isso ocorre para não haver informações desnecessárias em uma determinada etapa, ou seja, para que as informações fornecidas pelos diagramas sejam usadas cada uma em seu devido setor ou etapa.

Vimos que a UML não é uma linguagem de programação, mas que pode ser utilizada para execução, mas a ideia dela é facilitar e padronizar o desenvolvimento de software.

Portanto, para que desenvolvamos um software de qualidade e que siga um padrão em sua criação do sistema, é importante que a equipe domine essa linguagem que desenha os fluxos por meio de diagramas.

Referências:

Livro Utilizando UML e Padrões, 3ª Edição, LARMAN, Craig.

<https://eufacoprogramas.com/diagramas-uml-resumo/#caso-de-uso> 29/05/2021 às 00h30.

http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/classes/images/especificacao_exemplo_banco.GIF , consulta realizada em: 30/05/2021 às 7h41.

https://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/objetos/diagrama_de_objetos.htm , consulta em 30/05/2021 às 7h50.

https://www.cin.ufpe.br/~gta/rup-vc/core.base_rup/guidances/guidelines/sequence_diagram_AFA76EBB.html consulta 30/05/2021 às 8h00.