



Iniciativa da CNI - Confederação
Nacional da Indústria

SÉRIE TECNOLOGIA DA INFORMAÇÃO - SOFTWARE

MODELAGEM DE SISTEMAS





*Iniciativa da CNI - Confederação
Nacional da Indústria*

SÉRIE TECNOLOGIA DA INFORMAÇÃO - SOFTWARE

MODELAGEM DE SISTEMAS



CONFEDERAÇÃO NACIONAL DA INDÚSTRIA – CNI

Robson Braga de Andrade
Presidente

DIRETORIA DE EDUCAÇÃO E TECNOLOGIA - DIRET

Rafael Esmeraldo Lucchesi Ramacciotti
Diretor de Educação e Tecnologia

SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL - SENAI

CONSELHO NACIONAL

Robson Braga de Andrade
Presidente do Conselho Nacional

SENAI – DEPARTAMENTO NACIONAL

Rafael Esmeraldo Lucchesi Ramacciotti
Diretor-Geral

Gustavo Leal Sales Filho
Diretor de Operações



Iniciativa da CNI - Confederação
Nacional da Indústria

TECNOLOGIA DA INFORMAÇÃO - SOFTWARE

MODELAGEM DE SISTEMAS



© 2019. SENAI – Departamento Nacional

© 2019. SENAI – Departamento Regional de Santa Catarina

A reprodução total ou parcial desta publicação por quaisquer meios, seja eletrônico, mecânico, fotocópia, de gravação ou outros, somente será permitida com prévia autorização, por escrito, do SENAI.

Esta publicação foi elaborada pela equipe de Educação a Distância do SENAI de Santa Catarina, com a coordenação do SENAI Departamento Nacional, para ser utilizada por todos os Departamentos Regionais do SENAI nos cursos presenciais e a distância.

SENAI Departamento Nacional

Unidade de Educação Profissional e Tecnológica - UNIEP

SENAI Departamento Regional de Santa Catarina

Gerência de Educação e Tecnologia

FICHA CATALOGRÁFICA

S491m

Serviço Nacional de Aprendizagem Industrial. Departamento Nacional.
Modelagem de sistemas / Serviço Nacional de Aprendizagem Industrial.
Departamento Nacional, Serviço Nacional de Aprendizagem Industrial.
Departamento Regional de Santa Catarina. Brasília : SENAI/DN, 2019.
101 p. : il. (Série Tecnologia da informação - Software).

ISBN 978 - 85 - 505 - 2421 - 4

1. Engenharia de software. 2. UML (Computação). 3. Ciclo PDCA. I. Serviço Nacional de Aprendizagem Industrial. Departamento Regional de Santa Catarina.
II. Título. III. Série.

CDU: 004.41

SENAI

Serviço Nacional de
Aprendizagem Industrial
Departamento Nacional

Sede

Setor Bancário Norte • Quadra 1 • Bloco C • Edifício Roberto Simonsen • 70040-903 • Brasília – DF • Tel.: (0xx61) 3317-9001 Fax: (0xx61) 3317-9190 • <http://www.senai.br>

Lista de Ilustrações

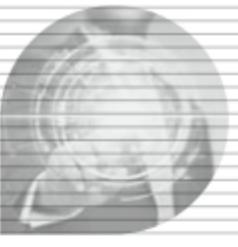
Figura 1 - Análise do escopo da modelagem	13
Figura 2 - Evolução dos requisitos	15
Figura 3 - Exemplo de pessoas-chave das regras de negócio	19
Figura 4 - Enterprise Architect	39
Figura 5 - Astah UML	40
Figura 6 - Umbrello	41
Figura 7 - Diagrama de classe de uma conta bancária	45
Figura 8 - Exemplo de diagrama de objetos para o caso conta bancária	48
Figura 9 - Exemplo de diagrama de componente	49
Figura 10 - Exemplo de diagrama de implantação	50
Figura 11 - Exemplo de diagrama de pacotes	52
Figura 12 - Exemplo de diagrama de estrutura composta	53
Figura 13 - Exemplo de restrição	54
Figura 14 - Exemplo de diagrama de caso de uso	55
Figura 15 - Exemplo de diagrama de transição de estado – estações do ano	59
Figura 16 - Exemplo de diagrama de atividade – venda de produtos	60
Figura 17 - Exemplo de diagrama de sequência – caso de uso efetuar login	62
Figura 18 - Exemplo de diagrama de visão geral de interação	63
Figura 19 - Exemplo de diagrama de comunicação	64
Figura 20 - Exemplo de diagrama de tempo	65
 Tabela 1 - Exemplo de Planilha para Entrevista de Requisitos	20
Tabela 2 - Exemplo de preenchimento da planilha para entrevista de requisitos	21
Tabela 3 - Modelo de documento dos requisitos do sistema	23
Tabela 4 - Modelo de Análise de Ferramenta UML	42
Tabela 5 - Tabela de análise do diagrama exemplo	46
Tabela 6 - Tabela de significado da anotação de multiplicidade	47
Tabela 7 - Tabela de explicação dos elementos do exemplo de diagrama de caso de uso	56
Tabela 8 - Modelo proposto para estudo dos casos de uso	57
Tabela 9 - Exemplo de uso para modelo proposto de estudo dos casos de uso	58
Tabela 10 - exemplo de tabela de monitoramento de resultado	71
Tabela 11 - exemplo de formulário de organização de etapas e atividades	80
Tabela 12 - exemplo de utilização do formulário de organização das etapas e atividades	82

Sumário

1 Introdução	9
2 Requisitos de Sistemas	11
2.1 Regra de negócio	12
2.2 Levantamento de requisitos do sistema.....	14
2.2.1 Requisito funcional.....	16
2.2.2 Requisito não funcional.....	17
2.2.3 Técnica de análise de requisitos	18
3 Modelagem de Sistemas.....	27
3.1 Regras e requisitos para a modelagem de sistemas.....	28
3.2 Definição de sistema.....	30
3.3 Tipos de sistemas.....	31
3.4 Características.....	33
4 Técnicas de Modelagem.....	37
4.1 Ferramentas.....	38
4.2 Linguagem UML	42
4.2.1 Diagramas da UML	44
5 Iniciativa	67
5.1 Formas de demonstrar iniciativa	68
5.2 Resultado.....	69
5.3 Autonomia	71
5.4 Consequências favoráveis e desfavoráveis	73
6 Organização de Dados.....	75
6.1 A importância da organização	76
6.2 Roteiro de trabalho	79
6.3 Organização de dados para análise	82
7 Métodos e Técnicas de Trabalho	87
7.1 Análise de informações e dados	88
7.2 Ciclo PDCA	90
7.2.1 Planejar	92
7.2.2 Fazer	92
7.2.3 Checar	93
7.2.4 Agir	94
Referências.....	97
Minicurrículo do Autor	99
Índice	101

Introdução

1



Olá! Seja bem-vindo à unidade curricular de Modelagem de Sistemas! A partir de agora você vai identificar a importância determinante da modelagem de sistemas. Saiba que, para um sistema computacional ter qualidade, robustez e segurança, ele necessita ser desenvolvimento levando em conta diversas técnicas de desenvolvimento e gestão que assegurem a integridade dos requisitos de negócio até a implantação e manutenção do sistema criado.

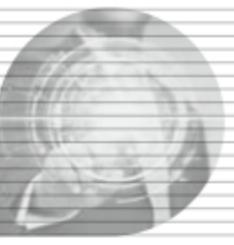
Nessa unidade curricular, você vai dominar as técnicas amplamente utilizadas no mercado atual para efetuar os mapeamentos dos requisitos de negócio de um sistema, tal como a utilização de ferramentas e técnicas que permitem definir e descrever diagramas universais de modelagem que podem ser interpretados pelos desenvolvedores para a conversão dessas necessidades de negócios em sistemas computacionais exequíveis.

Você descobrirá, assim como eu descobri há muitos anos, que fazer sistemas de computação é muito mais lidar com pessoas (e seus desejos) do que programar milhares de linhas de códigos de programação.

Ficou curioso para iniciar? Então vamos aos primeiros passos para a modelagem de sistemas!

Requisitos de Sistemas

2



Neste capítulo você vai conhecer quais são os requisitos e como eles se dividem entre funcionais e não funcionais. Além disso, também vai aprender como registrar e analisar documentos. Você vai perceber que a participação dos usuários no levantamento de requisitos é fundamental para que as necessidades possam realmente ser identificadas e posteriormente atendidas. Partindo desses estudos, ao final deste capítulo você terá subsídios para:

- a) Identificar requisitos funcionais;
- b) Identificar requisitos não funcionais;
- c) Identificar a regra de negócio para o desenvolvimento de sistemas;
- d) Interpretar requisitos levantados para o desenvolvimento de sistemas.

Inicie seus estudos conhecendo o que são regras de negócio.

2.1 REGRA DE NEGÓCIO

Ser capaz de colher, identificar, documentar, interpretar e analisar os requisitos é fundamental para modelar um sistema que tenha aderência às reais necessidades de um negócio. E você sabe por quê? Vamos a um exemplo!

Quando você vai fazer um jantar, deve ter muito claras diversas questões, tais como:

- a) Quantas pessoas estarão nesse jantar;
- b) A que horas ele deverá ser servido;
- c) Qual será a entrada;
- d) Qual será o prato principal;
- e) Quais serão os acompanhamentos do prato principal;
- f) Qual será a sobremesa.

Entender cada uma dessas questões é fundamental para que um jantar não seja um verdadeiro fracasso, você não concorda? Na modelagem de sistemas não é muito diferente. É preciso entender os requisitos que envolvem aquele sistema que vai ser criado.

Voltando ao exemplo, imagine que, se você errasse apenas dois requisitos apontados, poderia ter um resultado nada agradável. Você poderia, por exemplo, errar na quantidade de pessoas que estariam no jantar e na hora em que ele deveria ser servido. Como consequência, poderia ocorrer falta de comida para todos (ou a sobra, com o excesso) e a comida estar fria (se servida muito antes do horário), ou os convidados estariam com muita fome, entupidos de entradas (no caso de servir muito depois do horário).

Não compreender, ocultar, ignorar ou errar um requisito pode levar a um sistema com um "gosto" muito ruim. Agora que você já tem uma breve noção sobre a importância dos requisitos, confira a importância da regra de negócio.

Quando uma empresa o contrata como um profissional que atuará na Modelagem de um Sistema, serão as regras de negócio que determinarão o escopo do projeto que será desenvolvido.

Um bom profissional modelará um sistema que atenda integralmente às necessidades do cliente, sem exagerar em funcionalidades desnecessárias e sem deixar de fora questões essenciais para o sucesso do cliente.

E como isso é possível? É preciso que o profissional, por meio de diversas técnicas, consiga extrair, documentar e analisar diversas informações (requisitos) do negócio do cliente. Essas técnicas envolvem entrevistas, reuniões, formulários, canvas, planilhas, entre outros.

É importante que você entenda que para ser possível modelar um bom sistema, é essencial que você extraia da melhor forma possível as regras de negócio que seu sistema impactará. Da mesma forma que é importante você entender o escopo da modelagem, deixando de fora, e assim não perdendo tempo, o que não será impactado pelo sistema.



Vamos a mais um exemplo! Imagine que você está em uma grande empresa e deve modelar um sistema para sanar diversos problemas de controle financeiro que a empresa possui. O que seria pertinente para ser foco do seu trabalho de modelagem? Observando esse pequeno cenário, poderíamos afirmar que:



Figura 1 - Análise do escopo da modelagem
Fonte: Do autor (2019)

Note que isolar as regras de negócio é muito importante para que o projeto não fique inflado demais nem inferior às necessidades do cliente. Inicialmente, essas regras de negócio, que farão parte do escopo, são facilmente identificadas a partir de algumas reuniões de levantamento dos desejos do seu cliente. Nessas reuniões, procure entender dois pontos importantes:

O que o sistema deverá fazer de forma macro:

- Em quais áreas o sistema deve atuar?
- Existe algum sistema atualmente?

Sobre o negócio do cliente:

- O que é feito na empresa?
- De que forma é feito?
- Quantos funcionários?

- d) Quantos clientes?
- e) Existem filiais?
- f) Quais os produtos e serviços?
- g) Existe algum planejamento estratégico que vise atender à área que será impactada pelo sistema?

Com essas questões respondidas, você já teria uma excelente visão das principais regras de negócio que devem fazer parte do Levantamento de Requisitos e dos possíveis pontos de atenção para a construção de um bom documento que conterá as particularidades dessa modelagem. Como esse foi apenas um pequeno exemplo, é importante ressaltar que no mundo real caberiam muitas outras perguntas para identificar diferentes possíveis necessidades da empresa. Vamos à técnica do levantamento de requisitos do sistema!

2.2 LEVANTAMENTO DE REQUISITOS DO SISTEMA

O levantamento de requisitos é a técnica utilizada para, de forma sistemática e documental, efetuar o mapeamento dos requisitos funcionais e não funcionais dentro do escopo das regras de negócio que são impactadas pelo sistema que será modelado.

É importante que você entenda bem o que são requisitos, pois, uma vez que eles tenham sido entendidos, ficará muito mais fácil classificá-los entre funcionais e não funcionais.



FIQUE ALERTA

Os requisitos são objetivos ou restrições estabelecidas pelo seu cliente e pelos usuários do futuro sistema, e esses objetivos e restrições definem as diversas propriedades do sistema que será modelado. Sendo assim, os requisitos de sistema são as propriedades essenciais que delimitam o escopo do software que será modelado.

Os principais passos para executar o levantamento de requisitos são:

- a) Elicitação de Requisitos:** identificação dos requisitos do sistema, a partir das técnicas de coleta de informações.
- b) Análise de Requisitos:** os requisitos coletados são analisados detalhadamente e os representantes dos usuários (*stakeholders*) devem negociá-los de forma a chegarem em consenso.
- c) Documentação:** os requisitos são formalizados em documento de comum entendimento.
- d) Validação:** identificar possíveis problemas nos requisitos antes de partir para a modelagem e posterior desenvolvimento.

E veja bem! É determinante que você, como profissional que modelará sistemas, entenda que os requisitos evoluem de formas distintas, conforme a visão e contribuição de cada pessoa que participa no criativo e subjetivo processo de criação de um sistema computacional. Para ilustrar melhor, observe a

figura seguinte, que procura fazer uma analogia da mutação que ocorre com os requisitos, conforme você avança em um documento para efetivar o Levantamento de Requisitos.

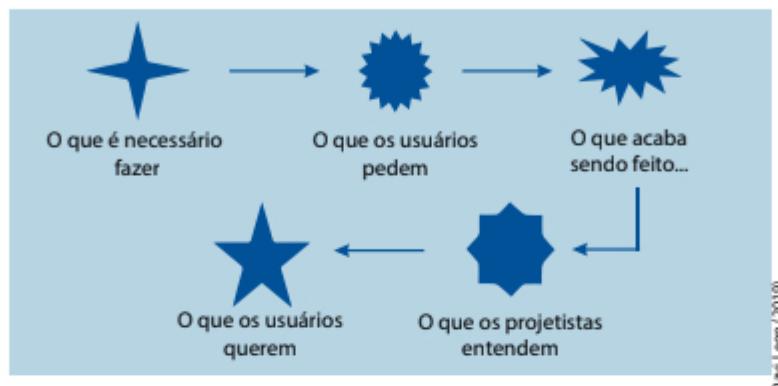


Figura 2 - Evolução dos requisitos
Fonte: Do autor (2019)

É muito provável que você tenha percebido o enorme gap que existe entre o que realmente era necessário ser realizado e o que o usuário falou que desejava, entre o que se projeta e o que se desenvolve. Isso ocorre principalmente pelo fato de que é muito complexo explicar o que se deseja. A maioria das pessoas sabe expressar de forma sucinta o que quer, porém não consegue explicar de forma fragmentada e com diversos detalhes.

Além da barreira do "contar o que se deseja", há um problema está do outro lado, em quem está "escutando os desejos". Saber extrair de forma fiel e sucinta aquilo que está sendo contado como desejado é um desafio gigante, pois deve-se traduzi-lo para o formato de requisitos, isolando o que deve ser alvo de modelagem do que é meramente um processo externo ao sistema ou um modo operante que sequer existirá após a implantação do sistema.

Então você deve estar se perguntando: "ok, eu já colhi diversas informações-macro sobre o negócio do cliente, mas como vou, agora, transformá-las em Requisitos?". Bem, antes de fazer isso, você deve saber a diferença entre requisitos funcionais e não funcionais. E é isso que você vai ver no próximo tópico. Acompanhe!



Steve Jobs, fundador da Apple, foi um dos maiores gênios da nossa era, inovando constantemente seus produtos e ditando durante décadas a direção dos produtos mais desejados do universo tecnológico. Ele sabia com maestria lidar com essa complexa relação de desejos dos seus clientes com o movimento do mercado. Invista algum tempinho, olhando algumas frases muito famosas dele, você identificará questões muito bacanas, que, com certeza, lhe serão úteis. Em destaque, veja esta brilhante frase do Steve Jobs, que define muito bem o que você está vendo aqui: "As pessoas não sabem o que querem, até mostrarmos a elas."

<https://www.pensador.com/autor/steve_jobs/>.

2.2.1 REQUISITO FUNCIONAL

É possível definir como requisito funcional as diversas funções que os clientes desejam/precisam no futuro sistema. Esses requisitos definem então as funcionalidades desejadas para o software. É importante entender que o termo “função” é aplicado no sentido genérico de uma operação qualquer que poderá ser realizada pelo sistema, seja por meio da interação com os usuários, seja pela execução de eventos externos/internos do sistema.

Os requisitos funcionais que você mapeará são exatamente aqueles que apontarão o comportamento do sistema. São eles que explicam a essência funcional do sistema que será modelado e desenvolvido, o verdadeiro “para que esse sistema existe e o que ele faz”.

Para que você compreenda melhor ainda, veja alguns exemplos de possíveis requisitos funcionais de um sistema:

- a) O sistema deve possibilitar o cálculo dos gastos diárias, semanais, mensais e anuais com pessoal;
- b) O sistema deve emitir relatórios de compras a cada dez dias;
- c) O usuário deve conseguir obter o número de aprovações, reprovações e trancamentos em todas as disciplinas por um determinado.



Jirapong Manustrong

Sempre procure ajudar o seu cliente na construção dos requisitos funcionais, pois muitas vezes eles estão ocultos nos processos de trabalho que diariamente ele executa ou gostaria de executar. O processo para extrair esses requisitos funcionais não é unilateral, do usuário para você, pois muitas vezes você será o proponente de um requisito que, obviamente, deverá passar pelo crivo do usuário. Afinal, ninguém melhor do que o usuário para definir se algo será ou não pertinente para seu dia a dia de trabalho.

Aliás, aproveitando esse tema, não existe forma melhor de se criar um bom sistema computacional que pelo constante contato com os futuros usuários. É importante que você comprehenda que, se criar um sistema do inicio ao fim apenas com o que você pensa, terá por garantia um usuário. Agora, quando você envolve várias pessoas nesse processo, procurando escutar ao máximo as sugestões, críticas, ideias, etc., poderá ter um sistema que possui maior aderência às múltiplas necessidades dos usuários. E isso, muito provavelmente, será mais do que um usuário apenas.

Nos primórdios da computação, os profissionais de TI exerciam suas atividades muito distantes dos usuários finais do sistema. No entanto, atualmente essa realidade é simplesmente descabida e sem sentido.

Hoje é possível saber que, quanto mais próximos e mais interação houver com os usuários finais do sistema, maior será a chance de entrar em contato com as reais necessidades daquele negócio. E é exatamente essa oportunidade que todo bom profissional que modela um sistema deve perseguir.



CURIOSIDADES

A especificação de um requisito funcional deve determinar o que se espera que o software faça, sem a preocupação de como ele faz.

Ficou claro o que são os requisitos funcionais? Então vamos aos requisitos não funcionais.

2.2.2 REQUISITO NÃO FUNCIONAL

Já os requisitos não funcionais são as qualidades globais de um sistema, tais como a manutenibilidade, usabilidade, desempenho, confiabilidade, performance, robustez, etc. Em grande parte, os requisitos não funcionais se relacionam com os padrões de qualidade.

Para entender melhor, vamos a alguns exemplos de requisitos não funcionais.

- a) O banco de dados deve ser protegido para acesso apenas de usuários autorizados;
- b) O tempo de resposta do sistema não deverá ultrapassar 30 segundos;
- c) O sistema deve ser operacionalizado no Linux;
- d) O tempo gasto com o desenvolvimento não deverá ultrapassar oito meses.

Uma boa maneira para você identificar um requisito não funcional é perceber que, muitas vezes, ele não faz exatamente algo no sistema, no sentido de executar uma função, mas restringe, qualifica.

Então, quando se fala que o sistema deve fazer algo, muito possivelmente trata-se de um requisito funcional, se este algo deve ser feito com a restrição X ou a exigência Y, provavelmente estariamos falando de um requisito não funcional. Inicialmente, talvez, esses dois conceitos lhe pareçam difíceis de identificar, mas naturalmente você notará que é bastante fácil e requer apenas um pouco de prática.

2.2.3 TÉCNICA DE ANÁLISE DE REQUISITOS

Para coletar, registrar, organizar, categorizar e, posteriormente, analisar os requisitos, sejam eles funcionais ou não funcionais, é determinante que você tenha boas ferramentas e um método claro para a execução dessa tarefa. Aqui você vai conhecer apenas uma das inúmeras formas possíveis que um bom profissional poderia utilizar para vencer o desafio de uma boa modelagem. Vamos lá!

Imagine que uma empresa que atua na sua região solicita que você modele um sistema para controlar sua área financeira. Primeiramente, você faz um levantamento das regras de negócio que compõem a empresa e chega à seguinte visão-macro:

A empresa possui 15 anos de existência;
A empresa possui 32 colaboradores;
A empresa vende calçados;
As vendas são todas feitas presencialmente;
Os vendedores ganham comissão por venda efetivada;
A empresa possui três unidades espalhadas pela cidade (uma matriz e duas filiais);
A empresa não possui nenhum sistema computacional dedicado para a área financeira, atualmente utilizando planilhas eletrônicas, calculadoras e livros financeiros;

a) A área financeira é dividida da seguinte forma: Na matriz, existem três profissionais que cuidam de todo o financeiro:

O gerente financeiro é o responsável por todo o fechamento do que entra (recebimentos) e do que sai (pagamentos), apresentando mensalmente os resultados (balanço financeiro) para o dono da empresa;
O responsável pelas contas a pagar é a pessoa que cuida de todas as contas que devem ser pagas (tanto da matriz quanto das filiais);
O responsável pelas contas a receber é a pessoa que cuida de todas as contas que devem ser recebidas (tanto na matriz quanto das filiais). E é também o responsável por apresentar o desempenho de vendas de cada vendedor.

b) As filiais enviam semanalmente malotes com as contas a pagar e a receber para a matriz;
c) A matriz, por sua vez, lança semanalmente todos esses registros e faz o acompanhamento.

Tendo essas excelentes informações em mão, você teria, então, que partir para uma rodada de entrevistas com algumas pessoas-chave, procurando extrair o máximo de requisitos para a modelagem futura do sistema.

Ao analisar as regras de negócio, é possível extrair algumas pessoas-chave para participar da nossa rodada de requisitos:

Dono da empresa	Gerente financeiro
Responsável contas a pagar	Responsável contas a receber
Vendedor	Cliente

Figura 3 - Exemplo de pessoas-chave das regras de negócio
Fonte: Do autor (2019)

Propositalmente estão pintadas de verde as pessoas que se acredita que participarão da rodada de levantamento de requisitos. As que estão em amarelo talvez sejam acionadas. Já as em vermelho muito provavelmente ficarão de fora do mapeamento de requisitos.

Talvez você possa estar se perguntando, mas porque essa diferenciação? Vamos juntos analisar isso, olhando para a solicitação do cliente. "Uma empresa que atua na sua região solicita que você modele um sistema para controlar sua área financeira". Ao fazer o mapeamento das regras de negócio da empresa, é possível chegar a esses principais possíveis usuários do sistema: dono, gerente, contas a pagar, contas a receber, vendedor e, obviamente, ainda que de maneira oculta, o cliente (que é quem compra os calçados). Quando você imaginou onde estariam os requisitos que devem ser mapeados para um sistema financeiro, certamente percebeu que eles existiriam nas atividades realizadas justamente pelas pessoas pintadas de verde. Foi esta a lógica utilizada! Em amarelo estão pessoas que talvez tenham interferência em alguns requisitos e, por isso, estão em um ponto de atenção, para serem ou não entrevistadas apenas depois da entrevista dos que estão em verde.

Mas, para que você possa realizar qualquer tipo de entrevista para mapear os requisitos, devemos estar preparados com ferramentas e técnicas.

Como ferramenta, você poderia criar uma planilha para inserir cada requisito que perceber, enquanto a técnica utilizada para extraír os requisitos dessas pessoas-chave poderia ser a entrevista.

Você poderia, então, elaborar uma planilha para facilitar o registro dos requisitos encontrados. Veja um exemplo!

Responsável		Nº	
Entrevistado		Data	
Cargo		Validado	
RNº	TIPO		
Responsável Empresa	Responsável Modelagem		

Tabela 1 - Exemplo de Planilha para Entrevista de Requisitos
Fonte: Do autor (2019)

Compreenda melhor essa proposta com o detalhamento de cada um dos campos.

- a) Responsável:** é o profissional técnico (você, por exemplo) que está realizando a entrevista com a pessoa chave.
- b) Nº:** um número identificado para a entrevista, esse valor tem que ser único, para facilitar buscas e referências futuras.
- c) Entrevistado:** o nome completo da pessoa entrevistada.
- d) Data:** dia, mês e ano em que aconteceu aquela entrevista.
- e) Cargo:** qual o cargo da pessoa entrevistada na empresa.
- f) Validado:** se a versão desse documento se encontra validada para fechamento.
- g) RNº:** número do requisito mapeado na entrevista, pode ser um número sequencial e único.
- h) Requisito:** a descrição explícita do requisito mapeado. É essencial você sempre possuir bons requisitos, de forma a possuir um conteúdo rico e claro. Um bom requisito não é muito extenso (a ponto de ser de difícil leitura) nem muito superficial (dificultando a compreensão de seu propósito).

i) **Tipo:** se o requisito é funcional (F) ou não funcional (NF).

j) **Responsável Empresa:** a assinatura do responsável da empresa, ou responsável pelo projeto de implantação do sistema na empresa.

k) **Responsável Modelagem:** a assinatura do profissional que fez a entrevista (você, por exemplo).

Então, nesse cenário hipotético, a execução dessa planilha poderia ficar da seguinte maneira:

Responsável	Rafael Meyer	Nº	001
Entrevistado	João Pedro	Data	17/02/2019
Cargo	Gerente Financeiro	Validado	Não
RNº	TIPO		
01	O sistema deve possuir três módulos: Contas a pagar, Contas a Receber e Balanço Financeiro.		
02	Cada módulo deve ser acessado apenas pelas pessoas com permissão específica.		
03	O sistema deve ser capaz de fazer o fechamento (balancete) mensal, bastando informar as respectivas datas de início e fim.		
04	A matriz deve ser capaz de cadastrar e identificar contas a receber a partir das vendas que ocorrem em cada loja.		
05	Os usuários não podem ser capazes de editar contas que sofreram fechamento (balancete).		
06	O sistema deve permitir definir metas de vendas e de gasto, possibilitando comparar metas e resultados.		
07	O sistema deve permitir manter o acesso nos três módulos por meio de autenticação de login e senha.		
08	As senhas devem ser criptografadas.		
09	O sistema deve executar backup diariamente.		
10	O sistema deve funcionar em rede privada (intranet) sem precisar estar conectado na internet.		
Responsável Empresa	Responsável Modelagem		

Tabela 2 - Exemplo de preenchimento da planilha para entrevista de requisitos

Fonte: Do autor (2019)

Note que nesse exemplo foram extraídos diversos requisitos do gerente financeiro, o senhor João Pedro. No total foram 10 requisitos, sendo cinco funcionais e cinco não funcionais. Lembre-se de que não existe uma regra de proporção de requisitos funcionais ou não-funcionais. Tudo vai depender do que o cliente quer e do que ele necessita.

Para se destacar em sua atividade profissional, lembre-se de que não basta apenas escutar o que seu cliente deseja. Também é seu papel orientá-lo, tanto por meio de sugestões como de questionamentos.



seth_j_ra ([20-?])

No exemplo de mapeamento anterior, talvez uma das sugestões poderia ser o avanço de um sistema que, ao invés de funcionar apenas em uma rede local e privada, poderia ser substituído por um sistema totalmente na internet. Obviamente essa sugestão teria que ser debatida profundamente com seu cliente e nesse debate seria pertinente que você sempre apresentasse o lado positivo e o negativo dessas escolhas.

Após executar cada uma das entrevistas previstas e preencher corretamente cada um dos documentos, deve-se verificar se eles foram devidamente assinados. Agora, é importante que você crie uma planilha consolidada com todos os requisitos de todos os entrevistados. Isso será útil como documento norteador para a modelagem futura. Observe que foi utilizada uma estratégica de documentos primeiramente separados para cada usuário entrevistado e depois aprovados para, então, serem consolidados numa única versão.

Esse documento compilado de requisitos poderia seguir a seguinte proposta de modelo:

DOCUMENTO DOS REQUISITOS DE SISTEMA			
Empresa		Responsável	
Data de início		Data final	
Quantidade de usuários entrevistados			
Quantidade de entrevistas			
Quantidade de requisitos levantados			
Quantidade de requisitos funcionais			
Quantidade de requisitos não funcionais			

ENTREVISTA	RNº	REQUISITO	TIPO
Observações:			
Responsável pela modelagem			

Tabela 3 - Modelo de documento dos requisitos do sistema
Fonte: Do autor (2019)

Para que a utilização desse modelo fique mais clara e você veja o que representa cada dos campos descritos, leia as definições a seguir:

- a) **Empresa:** o nome da empresa contratante.
- b) **Responsável:** o nome do profissional responsável pelo projeto de modelagem (você, por exemplo).
- c) **Data de início:** data da primeira entrevista que compõe esse compilado de requisitos.
- d) **Data final:** data da última entrevista que compõe esse compilado de requisitos.
- e) **Quantidade de usuários entrevistados:** a quantidade de usuários que participaram na elaboração das entrevistas.
- f) **Quantidade de entrevistas:** a quantidade de entrevistas que foram necessárias para extrair todos os requisitos. Às vezes é preciso de mais de um encontro para cada usuário.
- g) **Quantidade de requisitos levantados:** a quantidade total de requisitos levantados em todas as entrevistas realizadas entre a data de início e data final.
- h) **Quantidade de requisitos funcionais:** a quantidade total de requisitos funcionais encontrados em todas as entrevistas.
- i) **Quantidade de requisitos não funcionais:** a quantidade total de requisitos não funcionais encontrados em todas as entrevistas.
- j) **Entrevista:** cada formulário de entrevista, possui um número identificador único, este é o número utilizado aqui.

- k) RNº:** o número de identificador do requisito no formulário de entrevista.
- l) Requisito:** o requisito em questão, que foi mapeado no formulário de entrevista.
- m) Tipo:** se o requisito é funcional (F) ou não funcional (NF).
- n) Observações:** espaço para que o profissional de modelagem possa colocar anotações referentes ao projeto. Aqui podem entrar questões como decisões ou recomendações técnicas e alguma especificação futura para facilitar a modelagem e desenvolvimento.
- o) Responsável pela modelagem:** a assinatura do responsável pelo documento de modelagem (você, por exemplo).

É importante que você compreenda que este modelo é apenas uma proposta de como poderia ser elaborado um documento que registrasse formalmente os requisitos. Sinta-se livre para, a partir de suas capacidades e sua criatividade, elaborar novos documentos, contendo outros campos que considere importantes para essa etapa de modelagem de sistemas. Outra verdade é que você notará, conforme for participando de casos reais de modelagem, que esses modelos de documento costumam naturalmente evoluir à medida que você conhece novos aspectos do universo da Tecnologia e Informação.

Afinal, até poucos anos atrás não existiam sistemas híbridos para aplicativos *mobile*, por exemplo. Então, alguns documentos para modelagem poderiam não conter os campos que ajudassem a mapear essa particularidade. Veja que, assim como a tecnologia evolui, os documentos criados para modelar sistemas costumam também sofrer influência de novas realidades.



CASOS E RELATOS

Adaptando o levantamento de requisitos

Em certo projeto de uma empresa de tecnologia, o trabalho passou a ser intermunicipal, ou seja, os usuários-chave estavam espalhados em algumas cidades, com uma distância considerável entre elas. Para não inflar demais os gastos com deslocamento, e evitar expor a equipe à insalubridade de viagens e mais viagens em uma BR perigosa, boa parte dos processos de entrevista e documentos utilizados para essa etapa de levantamento de requisitos foram adaptados.

Ao invés do deslocamento para uma reunião presencial, foram utilizadas salas de videoconferência. No lugar de registros impressos, foram criados documentos on-line compartilhados pela internet com os participantes.

Assim, os encontros presenciais foram reservados apenas para os momentos mais importantes, como a validação dos documentos ou debate sobre algum ponto crítico não resolvido nas reuniões virtuais. Com essas ações, foram economizados dinheiro, tempo e, no final, obteve-se um resultado tão eficaz quanto o de um modelo presencial.



RECAPITULANDO

Neste capítulo você viu a importantíssima arte da captação de requisitos dos usuários, sua respectiva categorização, devida documentação e posterior análise.

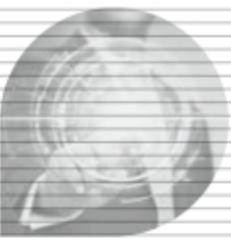
Viu também o quanto é fundamental para o sucesso de um bom projeto computacional que o sistema gerado tenha uma aderência funcional plena com as regras de negócio do cliente.

Você teve exemplos e modelos de documentos passíveis de serem utilizados para essas etapas que buscam a extração de requisitos. Da mesma forma, foi possível identificar que esses modelos devem acompanhar as necessidades de cada projeto e da própria evolução tecnológica.

Continue firme nos seus estudos, pois ainda tem muita informação importante esperando por você. Esses conhecimentos, aliados a sua determinação, gerarão um resultado incrível em sua vida profissional.

Modelagem de Sistemas

3



No capítulo anterior, você descobriu a importância de executar um bom mapeamento das regras de negócio e como, a partir dessas regras, gerar um documento com o levantamento de requisitos. Lembre-se de que, uma vez compreendendo bem o que deve ser feito (o domínio de negócio), é possível finalmente modelar o sistema. Mas, afinal de contas, o que é modelar o sistema? Neste capítulo você vai ver as definições de sistema, quais são os tipos de sistemas e suas características. Sendo assim, ao final deste capítulo, você terá subsídios para:

- a) identificar os diferentes requisitos de modelagem de sistemas;
- b) reconhecer diferentes definições de sistemas;
- c) compreender os diferentes tipos e características dos sistemas;

O primeiro assunto a ser tratado neste capítulo é regras e requisitos para a modelagem de sistemas. Bons estudos!

3.1 REGRAS E REQUISITOS PARA A MODELAGEM DE SISTEMAS

Para iniciar o estudo, é preciso definir os requisitos e as particularidades para a modelagem de sistemas. Para isso, vamos a um exemplo. Imagine que você deseja construir a casa dos seus sonhos. Você, muito provavelmente, buscará algum profissional da área de arquitetura, para que ele lhe auxilie na realização desse sonho. O arquiteto irá, inicialmente, conversar com você e entender melhor o que você quer (imagine que nesse momento ele fará o mapeamento das regras de negócio). Posteriormente, ele buscará maiores detalhes da casa, procurando definir melhor algumas particularidades que você deseja (imagine que agora ele está fazendo o levantamento dos requisitos). Depois disso, ele irá, a partir dessas duas coletas de informação (regras de negócio e requisitos), desenhar efetivamente a planta da casa, com todas as suas especificações técnicas (isso seria, finalmente, a modelagem de sistema).

Primeiro – Observe que nesse exemplo é possível fazer a seguinte analogia direta. Quando o arquiteto procura entender sua vontade e você explica que seu sonho é ter uma bela casa – pois acabou de se casar – e que, futuramente, quer ter dois filhos, ele executará o mapeamento das “regras do negócio”, definindo assim:

- a) Ele não quer apartamento;
- b) Ele quer uma casa;
- c) A motivação é o fato de ter se casado;
- d) Pretende ter dois filhos.

Segundo – Uma vez que o arquiteto tenha entendido essas condições essenciais das regras de negócio, com certeza ele avançaria no mapeamento de novos detalhes (requisitos) que facilitariam o desenho da casa. Após algumas conversas, seria possível exemplificar que ele levantaria os seguintes requisitos para a casa:

- a) Tem que ter uma suíte e mais dois quartos;
- b) Tem que ter uma garagem para dois carros;
- c) Um espaço nos fundos, para fazer futuramente uma piscina ou uma oficina;
- d) Os quartos devem ficar virados para o sol da manhã;
- e) Ter uma cozinha ampla;
- f) Ter uma área com churrasqueira voltada para os fundos.



Rawpixel ([20-7])

Com esse exemplo, ficou mais claro entender os papéis fundamentais que a análise das regras de negócio (o que você quer) e o levantamento de requisitos (as particularidades envolvidas no desejo) possuem no sucesso da modelagem de sistema, não é mesmo?

**FIQUE ALERTA**

Só é possível ter uma boa modelagem de sistema se o profissional responsável tiver de forma muito clara os aspectos fundamentais que existem no projeto em questão. E uma das melhores formas de se fazer isso é executar um pertinente mapeamento das regras de negócio com um detalhado e rico levantamento de requisitos.

Portanto, modelar um sistema nada mais é do que, a partir das regras de negócio e respeitando os requisitos, desenhar um sistema utilizando ferramentas e técnicas para que, futuramente, os programadores possam codificá-lo corretamente e o resultado tenha grande aderência com o desejo do cliente.

Com o que estudou até aqui, você acha viável construir uma moradia sem entender os desejos do cliente? É possível acertar na quantidade de quartos, estilo arquitetônico, posição da casa, cômodos em destaque, etc., sem efetuar um bom levantamento com o cliente? Com certeza você já deve ter chegado à grande conclusão de que isso é praticamente impossível. E é essa maturidade profissional que você precisa alcançar quando se fala em modelar e desenvolver sistemas computacionais.

É determinante que você tenha cravado em seu “DNA Tecnológico” que, para desenvolver um excelente sistema, deve compreender profundamente o que o seu cliente deseja. Só assim você trilhará o caminho do sucesso e de referência profissional.

3.2 DEFINIÇÃO DE SISTEMA

Até aqui muito se tratou de sistema, mas, afinal de contas, você sabe o que é um sistema? Ao buscar o significado dessa palavra em um dicionário, é possível encontrar “reunião dos elementos que, concretos ou abstratos, se interligam de modo a formar um todo organizado”. É possível encontrar sistemas em diversos contextos, como, por exemplo:

- a) Sistema solar;
- b) Sistema digestivo;
- c) Sistema operacional;
- d) Sistema de governo.

Quando você analisa o significado de sistema no contexto da Ciência da Computação, é possível defini-lo com um conjunto de diversos dispositivos eletrônicos, comumente chamados de hardware, que são capazes de processar as informações conforme as demandas de um programa, que normalmente é chamado de software.

Quando você for modelar um sistema, muito possivelmente vai focar na composição e detalhes futuros do software e de como ele interagirá com o hardware e, obviamente, com os usuários. Mas para que isso se concretize é preciso entender os tipos e as características dos sistemas, pois assim nossa capacidade de modelagem será consequentemente refinada.



**SAIBA
MAIS**

Para complementar os seus estudos, é importante que você busque se aprimorar na área de gestão de processos organizacionais. Atualmente, muitas empresas procuram diagnosticar e especializar cada vez mais seus processos, controlando-os com padrões bem definidos, ferramentas, sistemas, etc. E é exatamente pela pertinência de sistemas no contexto de processos, que você deve investir na área. Sabe-se que, invariavelmente, a boa gestão da empresa passa também pelos sistemas computacionais e dominar a colaboração que estes podem trazer para a gestão baseada em processos é um grande diferencial nas modernas técnicas gerenciais.

Lembre-se sempre: os limites dos seus conhecimentos são inteiramente de sua responsabilidade, é você que restringe ou amplia os seus saberes.



istockphoto #20-7)

Em uma empresa real, um sistema de gestão é formado por diversos componentes, procedimentos, processos, atividades, ferramentas, equipamentos, softwares, hardwares, pessoas, documentos, etc. Vamos aos tipos de sistemas? Acompanhe!

3.3 TIPOS DE SISTEMAS

Agora você já entendeu um pouco mais sobre as definições de sistemas, notou que eles estão espalhados em diversos contextos, tendo propósitos distintos, mas com um significado muito próximo. Porém, quando se fala especificamente de sistemas computacionais, é preciso compreender principalmente como eles se dividem e se classificam. Quais as características pertinentes a cada um desses tipos e como é possível classificá-los.

Os sistemas computacionais (softwares) podem ser classificados em dois grandes grupos:

- a) Software de sistema;
- b) Software de aplicação.

Softwares de sistema formam uma coleção de programas escritos para fornecer serviços a outros programas. Seu objetivo é dar suporte para que softwares de aplicação possam desempenhar suas funcionalidades-fim sem prejuízo de interpretação e resultados.

Se de um lado estão os softwares de sistema, que fornecem uma "plataforma" confiável de execução, por outro, há os softwares de aplicação, programas *stand-alone* (que podem rodar sem internet ou outros recursos em rede) que procuram resolver um problema de negócio específico.

Dessas duas significativas divisões, você terá diversas outras classificações, que estão, de certa forma, vinculadas a esses grupos primários.

Software de engenharia ou científicos: são programas desenvolvidos para rodar grande quantidade de processamentos numéricos, com ênfase de aplicação em áreas como astronomia, biologia molecular e outras que possuem essa característica.

Software embutido: são programas contidos dentro de um produto ou sistema, normalmente usados para controlar algum tipo de equipamento, que fornecem diversas funcionalidades ao usuário final.

Software para linha de produtos: esses programas são projetados para solucionar questões específicas para serem usadas por diferentes consumidores. São programas de venda de prateleira como, por exemplo, o Microsoft Office, que provê ferramentas de uma linha de produtos amplamente usadas em diferentes áreas do mercado.

Aplicações web: são programas utilizados em rede, abrangendo muitas aplicações acessadas a partir de um navegador, por computadores ou dispositivos móveis.

Software de inteligência artificial: são programas que usam algoritmos com o intuito de resolver problemas complexos, procurando traçar, por exemplo, algum comportamento, reconhecimento, tendência, etc.



www.shutterstock.com



CURIOSIDADES

Em 2015, quando a Microsoft anunciou o lançamento do Windows 10, a meta era chegar a 1 bilhão de dispositivos em três anos, ou seja, os planos eram obter esse resultado até 2018. Os números reais, no entanto, acabaram sendo um pouco mais modestos: a empresa recentemente confirmou que chegou a 700 milhões de aparelhos ativos ao redor do planeta.

O importante é que você compreenda que quando se fala em um sistema computacional, é possível tratar de diversos e muito diferentes conceitos de aplicação e de propósito final. A própria forma para se desenvolver essas soluções muda de drasticamente. E, se estamos afirmando que o modo de desenvolver se altera, com certeza a forma de se modelar esses sistemas também se modifica significativamente.

Quando você iniciar o seu estudo nos capítulos de modelagem, utilizando a Unified Modeling Language (UML), notará que, apesar de se tratar de modelagem para desenvolver um sistema, ela se alterará para atender às características deste.

3.4 CARACTERÍSTICAS

Ao classificar de forma mais apropriada os sistemas computacionais, você pode observar que cada uma dessas classificações fornece características particulares para esses sistemas. Tais características são diferentes exatamente porque respeitam as particularidades de cada negócio que esse sistema procura resolver.

Enquanto um sistema que será usado por engenheiros na construção de pontes, estradas, prédios e usinas hidroelétricas está extremamente focado em fornecer solução de cálculos, precisão nas medidas, previsibilidade de recursos necessários para executar aquela obra, outro sistema de prateleira, por exemplo o LibreOffice, está focado em entregar para seus usuários (de qualquer formação, idade, cultura, etc.) uma solução para diversas questões relacionadas à produção de documentos, apresentações, planilhas para cálculos, esquemas de desenhos, etc.

Você viu que temos dois sistemas com características fortemente estabelecidas, não é mesmo? No primeiro caso você tem uma solução focada nas atividades de determinado profissional (engenheiro), já o segundo exemplo não possui nenhum tipo de perfil profissional traçado. Na verdade, o programa poderia ser utilizado por uma criança que prepara um trabalho para a escola ou até mesmo por este professor e autor que está escrevendo esta unidade curricular.

Quando um desenvolvedor resolve, por exemplo, criar um sistema na web, são diversas as questões que ele deverá perceber. Qual problema esse sistema resolverá? Quais usuários irão usá-lo? Será necessária uma divisão de perfil de acesso? Quais os principais navegadores que deverão ser validados na sua utilização? Em qual servidor de aplicativo ele rodará? Qual plano de hospedagem ele utilizará?

Veja que, como nos exemplos citados, existem características de criação, manutenção, comercialização e análise de público totalmente diferentes.



FIQUE ALERTA

É muito importante que você, como futuro profissional com a competência de modelar sistemas, tenha a perspicácia de entender que cada sistema poderá ter características distintas. Imaginar que todos os sistemas serão modelados da mesma forma é um grande equívoco.

Então, além das referentes à modelagem, os sistemas com características particulares também nos levarão a uma reflexão sobre outras questões, como a infraestrutura, modelo de negócio, estratégias de manutenibilidade, desenvolvimento contínuo, suporte, implantação, treinamentos e muitas outras questões pertinentes ao mundo da tecnologia e informação.



CASOS E RELATOS

Documentos on-line

Carlos foi contratado para o desenvolvimento de uma solução *mobile* para um cliente. Porém, quando procurou Carlos, e enquanto ele fazia o levantamento das regras de negócio, o cliente usava sempre a expressão "um aplicativo em Android", e, na cabeça dele, o sistema seria desenvolvido apenas em Android.

Posteriormente, conforme o projeto foi avançando, assim como o levantamento de requisitos, Carlos foi entendendo melhor os desejos do cliente e a necessidade de negócio, descobrindo que uma parte considerável dos usuários futuros do sistema, possuíam iPhone em vez de aparelho Android.

A solução encontrada, após entender as regras de negócio e colher os requisitos, a partir das características que definem os sistemas, foi a escolha pelo desenvolvimento de um sistema híbrido, e não especificamente em Android. Assim, Carlos desenvolveu um aplicativo que rodava tanto em Android como no IOS. Um mesmo código fonte, com múltiplas saídas. Com isso, foi possível evitar a necessidade de ter dois projetos, um para cada sistema operacional *mobile*.



RECAPITULANDO

Neste capítulo você viu o que exatamente queremos falar quando usamos a expressão “modelar um sistema”. Você teve também a oportunidade de entender um pouco mais sobre os inúmeros conceitos existentes na palavra Sistema.

A partir disso, foi possível, ainda, classificar e analisar mais profundamente os tipos e as características dos sistemas. Você constatou que para cada tipo de sistema, e conforme suas particularidades, é preciso aplicar diferentes abordagens para, enfim, modelar um sistema condizente com o modelo de negócio ao qual se aplica.

Técnicas de Modelagem

4



Aqui você inicia um dos capítulos mais importante desta unidade curricular. É a partir deste capítulo que você vai começar a aprender a aprimorar a técnica de modelagem, utilizando ferramentas e linguagens pertinentes e precisas. Todos os capítulos anteriores foram os alicerces necessários para uma boa modelagem. Mas a modelagem propriamente dita é realizada utilizando-se de técnicas e algumas ferramentas. Como resultado, você vai obter diversos artefatos de modelagem, que podem ser interpretados por programadores, analistas de sistema, desenvolvedores, gerentes de produto e qualquer outro profissional que também domine essas técnicas. Compreender essas técnicas e dominar o uso dessas ferramentas é um grande diferencial na consolidação profissional, pois, além de ter um pensamento lógico atrelado ao conhecimento de algumas linguagens de programação, as empresas demandam cada vez mais profissionais que dominem padrões internacionais de modelagem de sistema.

Sendo assim, ao final deste capítulo você terá subsídios para:

- a) aplicar as ferramentas de linguagem UML;
- b) empregar os diagramas da UML;
- c) reconhecer a classificação e aplicabilidade da linguagem UML.

Já que o propósito é oferecer a você conhecimentos práticos sobre a modelagem de sistemas, inicie os estudos conhecendo as ferramentas.

4.1 FERRAMENTAS

Você lembra que aprendeu a mapear as regras do negócio e, posteriormente, a efetuar um bom levantamento de requisitos? Pois bem, agora vamos aprender a traduzir todos esses conhecimentos em artefatos da Unified Modeling Language (UML – Linguagem de Modelagem Unificada).

Mas, antes de aprender especificamente sobre a UML e todos seus ricos diagramas e técnicas, é preciso desbravar algumas possíveis ferramentas que você pode utilizar para tornar viável a execução das técnicas propostas pela UML.

Da mesma forma que um engenheiro civil ou arquiteto utilizam ferramentas para elaborar seus projetos (Autocad, por exemplo), profissionais que modelarão sistemas computacionais também possuem recursos que ajudam no ofício. Essas ferramentas podem ser facilmente encontradas pesquisando em qualquer site de busca, por exemplo, digitando a pesquisa "UML programa". Que tal efetuar essa busca? Vislumbre o amplo universo de programas e sites disponíveis para quem deseja utilizar ferramentas bacanas para modelagem de sistemas que utilizem o padrão proposto pela UML.



Serão apresentadas aqui diversas opções de ferramentas para modelagem que contenham uma abordagem da UML. Mas lembre-se de que as tecnologias evoluem muito rápido. Novas alternativas surgirão, assim como outras simplesmente podem desaparecer. No entanto, para cumprir os objetivos aqui definidos, muito provavelmente as ferramentas apresentadas continuarão durante um bom tempo, tendo em vista a análise do histórico de seu *road-map*.

Para estabelecer um bom mapeamento de ferramentas, primeiro você vai conhecer algumas dessas soluções e, posteriormente, fazer uma análise comparativa em uma tabela. Assim, ficará mais claro para você quais são os pontos fortes e fracos de cada uma dessas soluções. No entanto, é importante responder algumas questões comuns, como: Qual a melhor ferramenta para UML? A resposta é: depende do projeto, recursos disponíveis, quantidade de pessoas envolvidas, plataforma de desenvolvimento, maturidade da equipe e cliente.

São inúmeras as variáveis que podem levá-lo a escolher a solução de modelagem A ou B, mas o importante mesmo é que você tenha muito claras essas variáveis e saiba analisá-las de forma profissional e coerente.



CASOS E RELATOS

Software pirata, não!

Na universidade, logo nas primeiras fases, Antônio tomou uma importante decisão na sua vida formativa profissional: não usaria nenhum tipo de software pirata. Isso o impulsionou em direção a um universo novo e desconhecido até então, pois, como ele era estudante e não tinha dinheiro, teve que correr atrás de opções gratuitas e que o atendessem.

Foi então que Antônio teve o primeiro contato com Linux e um catálogo gigante de softwares muito interessantes, que não exigiam licenças comerciais inviáveis.

Você notará que muitas das soluções apresentadas aqui são oriundas dessa perspectiva. Portanto, saia da zona de conforto e fuja um pouco do tradicional, pois pensar diferente às vezes é fundamental.

Vamos começar por uma das ferramentas mais tradicionais para modelagem de sistemas, amplamente utilizada por engenheiros de software, o Enterprise Architect (EA). Mas o que é o EA? É uma plataforma de desenvolvimento colaborativa para modelagem, design e gerenciamento baseada em UML e padrões similares. É uma solução bastante robusta e completa, que possibilita criar projetos e dividir as aplicações da UML, separando e interligando os artefatos desenvolvidos. Além disso, é possível gerar código diretamente a partir do que se modela. Outro recurso cativante é a produção de documentação detalhada e de qualidade do projeto nos formatos RTF, PDF e HTML.

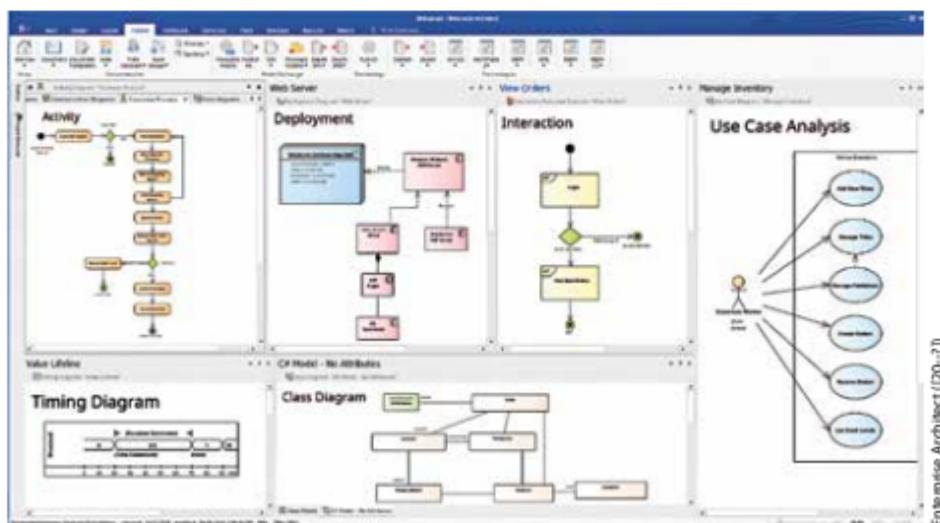


Figura 4 - Enterprise Architect

Obviamente é uma ferramenta muito ampla e robusta e não vamos perder tempo tentando explicar cada um dos recursos existentes, até porque, como você viu, recursos novos surgem assim como os antigos somem ou são modificados.

**SAIBA
MAIS**

Que tal saber mais sobre o EA? Visite a página oficial do EA <<https://sparxsystems.com/products/ea/index.html>> e leia um pouco sobre os recursos disponíveis nessa ferramenta.

Um projeto que pode ser usado para substituir o EA, e que se popularizou rapidamente, ganhando inclusive um novo nome, é o Astah.

O Astah UML é uma ferramenta muito completa, escrita em Java, que está disponível em qualquer sistema operacional. Além disto, ela possui alternativas gratuitas interessantes, uma versão comunitária e uma versão para estudantes.

Assim como o EA, o Astah permite uma excelente organização do projeto em pastas, permitindo manter de forma mais adequada os artefatos de UML. Ele também é capaz de gerar códigos a partir da modelagem, além de gerar uma documentação bem elaborada.

O Astah se divide em diversos projetos e cada um deles tem sua respectiva licença e abordagem.

**SAIBA
MAIS**

Vale a pena conferir com mais cuidado cada uma dessas propostas e suas respectivas licenças. Para isso, acessa o site do Astah <<http://astah.net/editions/uml-new>> e conheça melhor todas as opções das funcionalidades previstas nessa poderosa ferramenta. O Astah pode ser considerado uma grande e viável solução para a maioria dos profissionais e entusiastas que querem iniciar na caminhada de modelagem de sistemas.

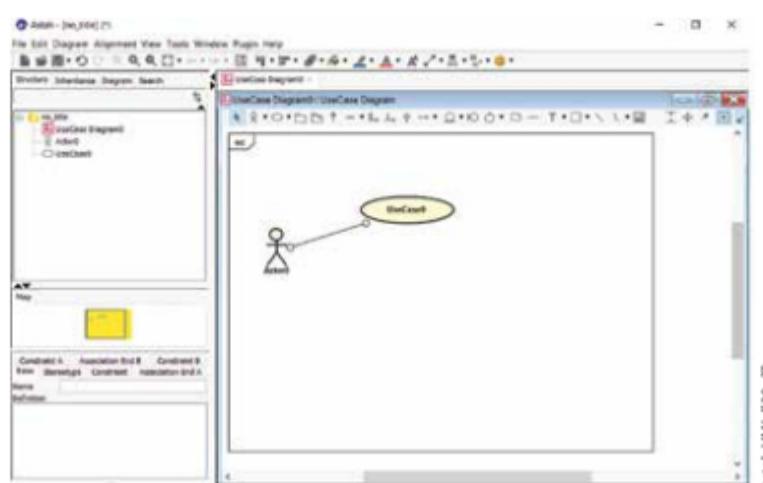


Figura 5 - Astah UML
Fonte: Do autor (2019)

Como terceira opção, você vai conhecer uma ferramenta chamada Umbrello, um cativante projeto desenvolvido em KDE para usuários Linux.

Você vai ver que essa ferramenta fornece facilidade e produtividade, além de ser executada no Linux.

O Umbrello hoje em dia também está disponível para Windows e para Mac, trazendo diversos recursos, como organização, documentação e geração de código.

Obviamente, como se trata de um projeto menor do que o poderoso EA ou até mesmo o Astah, ele terá menos recursos disponíveis. Porém é interessante que, na falta de outras boas opções, também conheça o Umbrello, pois, no mínimo, você ampliará seu leque de opções como profissional que atua na Modelagem de Sistemas.



**SAIBA
MAIS**

Para saber mais sobre o Umbrello, visite o site: <<https://umbrello.kde.org/>> e boa pesquisa!

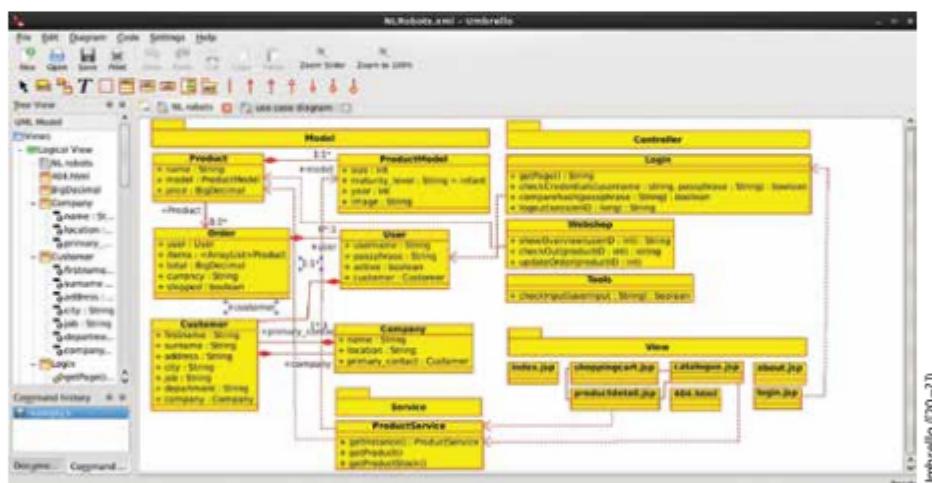


Figura 6 - Umbrello

Além dessas três ferramentas, você poderá encontrar diversas outras possíveis soluções para Modelagem de Sistemas que utilizem o padrão UML. Talvez encontre o *Diagram Editor* (DIA), um programa disponível para Windows, Mac e Linux com recursos bacanas para modelar em UML. Pesquisando mais um pouco, você também poderá encontrar o LucidChart, uma solução 100% on-line para produção e compartilhamento de modelagem de sistemas, inclusive com suporte para UML.

Além desses, existe uma gama de opções muito interessantes para escolher no mundo das ferramentas para modelar sistemas. Mas como saber qual deve ser a escolhida? Como escolher? Isso realmente pode ser uma grande dor de cabeça para a maioria das pessoas, mas não para nós! Pois, como profissionais de modelagem, é preciso saber que para decidir algo deve-se coletar informações, documentar e, posteriormente, analisar com calma os aspectos envolvidos, tomando, assim, a melhor decisão possível.

Portanto, veja alguns exemplos de como você poderia analisar algumas soluções em busca da mais apropriada para a sua realidade. Observe a proposta da tabela analítica a seguir, que será aplicada nas três soluções apresentadas neste tópico.

CRITÉRIO DE ANÁLISE	EA	Astah	Umbrello
VALOR DA LICENÇA	US\$ 229	US\$ 125	x
POSSUI VERSÃO GRATUITA	Não, apenas <i>trial</i>	Sim, a versão estudante	Sim
RODA EM WINDOWS	Sim	Sim	Sim
RODA EM MAC	Sim	Sim	Sim
RODA EM LINUX	Sim	Sim	Sim
SUPORTA UML	Sim	Sim	Sim
ORGANIZA O PROJETO	Sim	Sim	Sim
GERA CÓDIGO	Sim	Sim	Sim
POSSUI TRADUÇÃO EM PORTUGUÊS?	Sim	Sim	Não

Tabela 4 - Modelo de Análise de Ferramenta UML
Fonte: Do Autor (2019)

Obviamente, essa é apenas uma proposta que facilmente você poderia refinar adicionando, alterando ou removendo critérios de análise. Isso facilita muito para você fazer uma escolha mais precisa. Além do mais, fique livre para escolher a ferramenta de que gostar mais para acompanhar os estudos. Aqui será utilizado o Astah UML para gerar os artefatos que serão apresentados nos próximos tópicos, com o objetivo de elucidar, de forma detalhada, os diagramas da UML e suas respectivas aplicações.



CURIOSIDADES

Não utilize softwares piratas ou caqueados, pois, além de prejudicar seu computador, você estimulará uma indústria ilegal que joga contra a sua futura profissão, na área de TI. Existem muitas opções gratuitas e de grande qualidade disponíveis na internet, basta você sair da zona de conforto, procurar um pouco e aceitar o desafio de operar softwares menos conhecidos.

4.2 LINGUAGEM UML

Nos tópicos anteriores você viu tanto sobre essa tal Unified Modeling Language (UML – Linguagem de Modelagem Unificada), mas, afinal de contas, o que é isso? Bem, antes de mais nada, fique atento para a tradução desta sigla, que tão bem a define. Trata-se de uma linguagem para modelagem que segue uma proposta unificada, ou seja, padronizada e utilizada da mesma forma por vários profissionais.



SARVYAPINGAM (20-71)

**FIQUE ALERTA**

No meio profissional, é possível perceber muitos colegas erroneamente se referindo à UML como uma metodologia de modelagem. Mas isso na verdade é um equívoco, pois o que existe são metodologias de desenvolvimento ou de projeto que utilizam a UML como padrão para expressar os artefatos de modelagem.

A UML nasceu em 1997 a partir da fusão de três outras linguagens de modelagem (OMT, Booch e OOSE). Com o passar do tempo, foi sendo amplamente utilizada nas instituições de ensino e pelas empresas em todo o mundo, consolidando-se como um importante padrão.

Esta linguagem possui características muito favoráveis a sua adoção:

- a) Sua utilização é independente do domínio de aplicação, podendo ser usada em diferentes tipos de projetos (web, desktop, etc.);
- b) Independente do tipo de metodologia que será utilizada, pois o que ela fornece é um padrão de linguagem unificado;
- c) Não depende de ferramentas de modelagem específicas;
- d) Apresenta excelente escalabilidade;
- e) Possui um conjunto interessante de diferentes artefatos e técnicas para as mais variáveis linguagens de programação.

Esses são apenas alguns exemplos de fatores que foram extremamente favoráveis para consolidar a UML como uma linguagem de modelagem amplamente utilizada em todo mundo ao longo das últimas décadas. Mas vale destacar também que a UML conta com contribuições e direitos de autorias de grandes empresas e organizações. A UML é promovida pela Object Management Group (OMG), e possui em seu DNA as contribuições e direitos de autoria das empresas: Hewlett-Packard, IBM, Icon Computing, i-Logix, IntelliCorp, Electronic Data Services, Microsoft, ObjecTime, Oracle, Platinum, Ptech, Rational, Reich, Softeam, Sterling, Taskon A/S e Unisys.

É possível dizer então que a UML não é apenas uma moda passageira, como às vezes é possível ver, na acelerada e irrequieta área de TI, mas uma linguagem de padrão com mais de duas décadas de existência, com grandes empresas patronas nos bastidores. Além disso, contribui com artefatos, técnicas e padrões de diagramas para diversas metodologias e possui inúmeros entusiastas e profissionais espalhados pelo mundo inteiro.

É possível considerar que os conhecimentos sobre UML são um excelente diferencial para os profissionais de TI hoje em dia. E as empresas, em sua grande maioria, também pensam assim, pois identificam que um profissional que entende sobre UML possui uma visão crítica muito mais apurada na análise e desenvolvimento de boas aplicações.



**SAIBA
MAIS**

A Object Management Group (OMG) disponibiliza a possibilidade de certificação em UML a partir de uma avaliação. São 80 questões que procuram traçar o seu domínio de conhecimento em UML. Todas são de múltipla escolha. Porém, existem algumas em que você deve assinalar mais de uma como correta. A prova é bem conceitual, possibilitando validar bem o conhecimento da UML. Existem três níveis de certificação UML: *Fundamental, Intermediate e Advanced* (*Fundamental, Intermediário e Avançado*).

4.2.1 DIAGRAMAS DA UML

Agora que você conhece um pouco sobre UML (o que é, de onde veio, para que serve, quem a criou), está na hora de começar a descobrir um pouco mais sobre como utilizá-la, quais os diagramas, como eles se dividem e, principalmente, qual o propósito de cada um deles. Preparado? Então mãos à obra!



Sergii Mudnuk // 20-70

Até bem pouco tempo atrás, a versão da UML era a 2.5, comumente referida como pertencente às versões 2.x. Com o passar do tempo, talvez existam novas atualizações, o que não vai interferir no seu aprendizado, até porque você, como aluno aplicado e grande entusiasta do universo de TI, já deve saber que é impossível se manter atualizado na área tecnológica sem "remar" sempre a favor das novas tendências e novidades.

A OMG organiza os diagramas da UML em dois grandes agrupamentos: diagramas de comportamento e diagramas estruturais. Ao todo são apresentados 14 tipos, sendo sete comportamentais e sete estruturais.

Mas o que são eles? Os diagramas comportamentais existem com o propósito de apresentar os aspectos dinâmicos de determinado sistema. É possível considerar nessa categoria as relações de negócio realizadas dentro do sistema, de forma abstrata. Ao fazer uma analogia rápida, imagine que as pessoas que vivem dentro de uma casa e por ali interagem são os aspectos dinâmicos e dão o comportamento de uma casa.

Em contrapartida, os diagramas estruturais seriam todas as especificações técnicas que definem a materialidade de recursos computacionais do sistema. No mesmo exemplo, quando se fala de uma casa, estaríamos agora nos referindo a todas as questões estruturais dessa casa (material da construção, medidas, pintura, encanamentos, fiação elétrica, etc.).

Chegou a hora de conhecer um a um esses diagramas, sua classificação e, principalmente, sua aplicabilidade.

DIAGRAMA DE CLASSES

Vamos iniciar por esse importante diagrama, amplamente utilizado por grande parte dos profissionais de TI, principalmente porque traz diversas informações pertinentes à implementação de um sistema.

Mas o que é o diagrama de classes? Ele é classificado como um diagrama de estrutura, pois define as características que as classes do domínio do sistema possuem. O diagrama de classes se dá pela formação de conjunto de informações sobre determinadas classes que, unidas entre si, definem um sentido geral do projeto.

Veja, a seguir, um exemplo de um diagrama de classe.

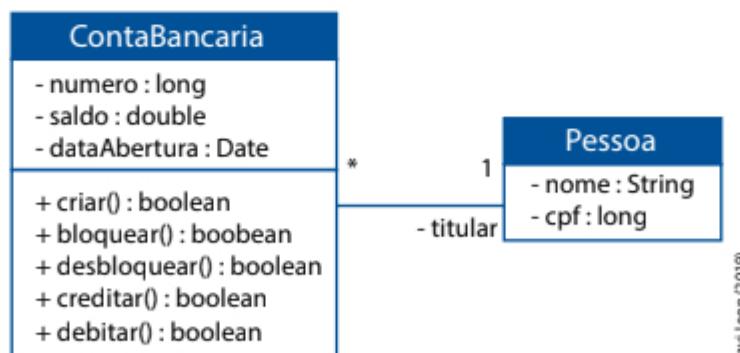


Figura 7 - Diagrama de classe de uma conta bancária
Fonte: Do autor (2019)

Davi Leon (2019)

Para que você entenda melhor esse diagrama, vamos analisar ponto a ponto cada item que compõe esse pequeno exemplo. Na sequência, você vai ver uma tabela que descreve os detalhes dos componentes que formam a figura do exemplo. Acompanhe!

ITEM	DESCRIÇÃO SOBRE O ITEM
ContaBancaria	Aqui está o NOME que define esta classe. No caso, ela se chama ContaBancaria. Note que, quando você está criando um diagrama de classe e ela procura definir questões estruturais, você deve evitar (a partir do que se fala nas boas práticas de programação) utilizar caracteres especiais, como acentuação, espaços em branco, etc. Por isso essa classe se chama "ContaBancaria" em vez de "Conta Bancária".
Pessoa	Aqui está o NOME que define a classe pessoa. Então, quando você olhar para esse diagrama de exemplo, poderá afirmar que ele possui apenas duas classes (ContaBancaria e Pessoa).
- numero : long - saldo : double - dataAbertura : Date	Esses são os Atributos que existem na classe ContaBancaria. Nesse exemplo, estou considerando que uma conta vai ter somente esses três atributos, que representarão o número, o saldo e a data de abertura. Novamente, note que na hora de definir os nomes dos atributos, não utilizei caracteres especiais. Posteriormente ao nome do atributo, é possível definir o tipo de dado que ele armazenará (Ex.: <i>long</i> , <i>double</i> e <i>date</i>).
- nome : String - cpf : long	Esses seriam os atributos da classe pessoa, apenas para concretizar o exemplo. Eles definem o nome e o CPF da pessoa que será titular de uma conta bancária.
+ criar() : boolean + bloquear() : boolean + desbloquear() : boolean + creditar() : boolean + debitar() : boolean	Agora, observe os métodos que a classe ContaBancaria possui. Trata-se de atributos muito importantes, pois definem o comportamento das classes. Nesse exemplo, existem alguns métodos que fazem ações na conta bancária e, como retorno, apresentam um valor <i>boolean</i> , ou seja, verdadeiro ou falso.
* 1 - titular	Por fim, você tem aqui a associação (representada pela linha que liga as duas classes) e a multiplicidade, representada pelo número 1 de um lado e o símbolo asterisco do outro. Abordarei essas questões sobre associação e multiplicidade logo a seguir.

Tabela 5 - Tabela de análise do diagrama exemplo
Fonte: Do Autor (2019)

Quando falamos em associação estamos nos referindo mais especificamente a um tipo de relacionamento que uma classe está tendo em relação a outra. No diagrama de classe existem diversos tipos de relacionamentos:

- a) Generalização ou herança:** Indica relacionamento entre um elemento mais geral e um mais específico (superclasse e subclasse, respectivamente). A subclasse pode conter somente informação adicional acerca da superclasse. Por exemplo, um médico é um funcionário.
- b) Agregação:** Usada para denotar relacionamentos todo/parte. Por exemplo, um item de compra é parte de um pedido.
- c) Associação:** Usada para representar relacionamentos entre as classes (por exemplo, um cliente pode alugar várias fitas de vídeo). As associações podem ser unárias (quando uma classe se associa a si mesma) ou binária (quando uma classe se associa a outra).
- d) Dependência:** Um relacionamento entre um elemento independente e outro dependente, no qual uma mudança no elemento independente afetará o elemento dependente.

Ao falar em multiplicidade, pense na cardinalidade das associações existentes em um diagrama de classes. Essa relação de proporcionalidade é denominada multiplicidade e apresenta quantas instâncias de uma classe podem participar da associação (semelhante à abordagem ER).

MULTIPLICIDADE	SIGNIFICADO
0..1	Zero ou uma instância
1	Somente uma instância
0..*	Maior ou igual a zero instâncias
*	Maior ou igual a zero instâncias
1..*	Maior ou igual a uma instância
1..N	Maior ou igual a "N" instâncias (ex.: 1..5)

Tabela 6 - Tabela de significado da anotação de multiplicidade
Fonte: Do autor (2019)

DIAGRAMA DE OBJETOS

Agora você vai estudar um diagrama que, em sua essência, é uma derivação do diagrama de classes: o diagrama de objetos. Você notará que existe uma semelhança em sua notação, mas eles diferem pelo fato de que este último mostra os objetos que foram instanciados nas classes. É possível pensar que o diagrama de objetos é como se fosse uma “foto” do sistema em certo momento de sua execução.

O diagrama de objetos assim como o diagrama de classes é classificado como um diagrama de estrutura.

O diagrama de objetos é muito usado quando é preciso apresentar a estrutura de um diagrama de classes para algum cliente ou parceiro que não possui a formação técnica apropriada para compreender esse artefato da UML. Nesses casos, ele seria um excelente recurso para vencer essa barreira, pois não traz consigo detalhes sobre boas práticas de nome de classe, atributos e funções e não apresenta uma definição técnica de tipagem de dados ou relações e multiplicidade (pelo menos não na forma técnica tradicional). Assim, o diagrama de objeto procura ser um pouco mais “humanizado”, apresentando o que pode estar ocorrendo em determinado momento do sistema.

Não é surpresa também utilizar o Diagrama de Objetos com profissionais iniciantes em programação, que podem não estar muito acostumados com a UML. Muitas vezes, ele foi utilizado como um recurso introdutório para, posteriormente, apresentar-se o diagrama de classes.

Lembra-se do exemplo de conta bancária no diagrama de classes apresentado no tópico anterior? Você vai utilizá-lo para fazer uma releitura dentro de um diagrama de objetos. É importante que você volte até o tópico anterior e olhe novamente de forma bem atenta para o diagrama de classes, para ter maior habilidade de comparação entre os dois tipos de diagrama.

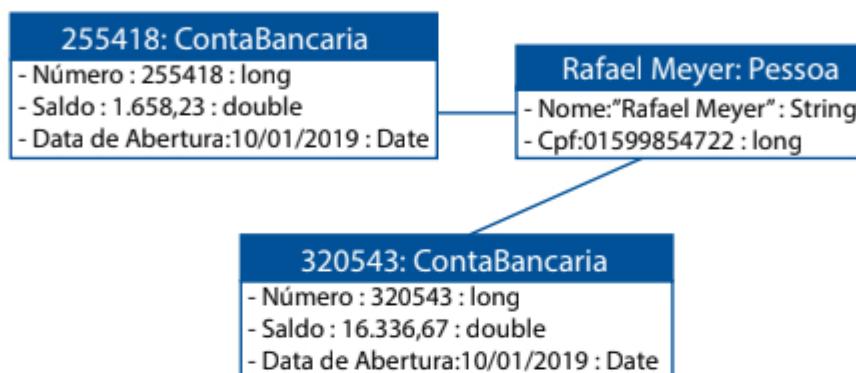


Figura 8 - Exemplo de diagrama de objetos para o caso conta bancária
Fonte: Do autor (2019)

Davi Leon (2019)

Veja como ficou claro na figura anterior um possível comportamento da estrutura do sistema quando recebe essas instâncias em seus elementos de classe. No exemplo, você tem a pessoa "Rafael Meyer", que possui duas contas bancárias. A primeira poderia representar uma conta corrente (255418), enquanto a outra (mais volumosa) talvez represente uma conta poupança (320543).

Note que, ao invés de ter uma linguagem visual mais técnica (como você viu no diagrama de classe), nesse exemplo você tem uma visão muito mais humanizada, mostrando uma "foto" do que poderia acontecer no sistema e como ele interpretaria e interagiria com os dados.

Muitas vezes você, como futuro profissional com competência para modelar sistemas, terá que utilizar esse recurso para sanar dúvidas mais profundas de seus clientes. Isso é algo louvável e pertinente, pois nenhum cliente gosta de olhar para o sistema que ele encomendou como se fosse apenas uma caixa preta que faz coisas mágicas. Ele realmente gostaria de compreender um pouco melhor o que se passa dentro do sistema, e o diagrama de objetos é, com certeza, um dos diagramas estruturais que mais se adéquam a esse público.

Pronto para seguir os estudos? Depois do diagrama de objetos, vamos ao diagrama de componentes.

DIAGRAMA DE COMPONENTES

Esse é mais um exemplo que pertence à categoria diagrama estrutural. Ele mostra os artefatos de que os componentes são feitos, como, por exemplo, os arquivos de código fonte, as bibliotecas de programação ou tabelas de bancos de dados que compõem o sistema.

A ideia principal desse diagrama é apresentar, do ponto de vista estrutural, como as partes (ou módulos) de um sistema se organizam. Além disso, ele pode mostrar de forma detalhada a interação entre diversos outros organismos tecnológicos que poderão estar em contato com o sistema.



Figura 9 - Exemplo de diagrama de componente
Fonte: Do autor (2019)

DaviLeon (2019)

Nesse exemplo você pode ver a conta bancária e sua respectiva relação, utilizando os componentes caixa eletrônico, seu controlador e o gerenciador de contas e as respectivas interfaces criadas para interação desses componentes.

Aqui fica fácil imaginar que, a partir de um caixa eletrônico que se comunica por uma interface com um controlador, os dados e ações solicitadas no caixa seriam validados e controlados pelo controlador de caixa que, por sua vez, poderia disparar ações pertinentes nas contas a partir de uma interface que se conecta com o gerenciador de contas.



Marcos Pimentel (2018)

Você deve ter percebido como o diagrama de componentes alerta, estruturalmente falando, os futuros desenvolvedores sobre como o sistema deverá ser estabelecido, não é mesmo? Essa ação facilita, assim, a manutenibilidade e a capacidade de escalabilidade. Vamos ao diagrama de implantação!

DIAGRAMA DE IMPLANTAÇÃO

O diagrama de implantação obviamente pertence à categoria dos diagramas estruturais, pois possui função muito típica de estrutura, que é explicar como o sistema se comporta com os diversos aspectos de uma implantação.

Sendo assim, ele apresenta como deverá ser realizada a distribuição do sistema por meio dos nós de hardware, seus componentes e dependências de softwares e as respectivas relações de comunicação.



FIQUE ALERTA

Uma maneira fácil de compreender o diagrama de implantação é saber que ele modela o inter-relacionamento entre diversos recursos, sendo eles de rede, infraestrutura ou artefatos de sistema. E cada um desses recursos é chamado, nesse diagrama, de nós (*node* no inglês).

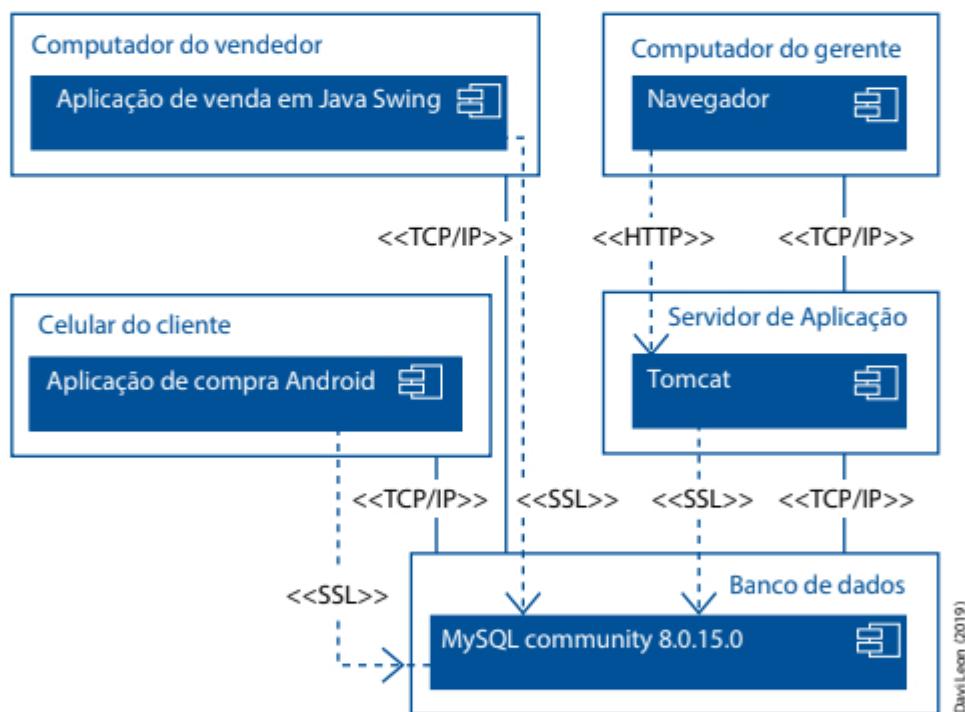


Figura 10 - Exemplo de diagrama de implantação
Fonte: Do Autor (2019)

Davi Leon (2019)

No exemplo anterior fica evidente no caso de uma implantação, quais seriam os componentes, recursos de rede e estratégias de comunicação que deverão ser adotados para que esse sistema de vendas possa se efetivar.

DIAGRAMA DE PACOTES

O divertido diagrama de pacotes é o quinto diagrama de estrutura que você vai conhecer. Saiba que ele é muito utilizado para apresentar como será organizada a arquitetura de um sistema, mostrando o agrupamento de suas classes na lógica da organização estrutural adotada.

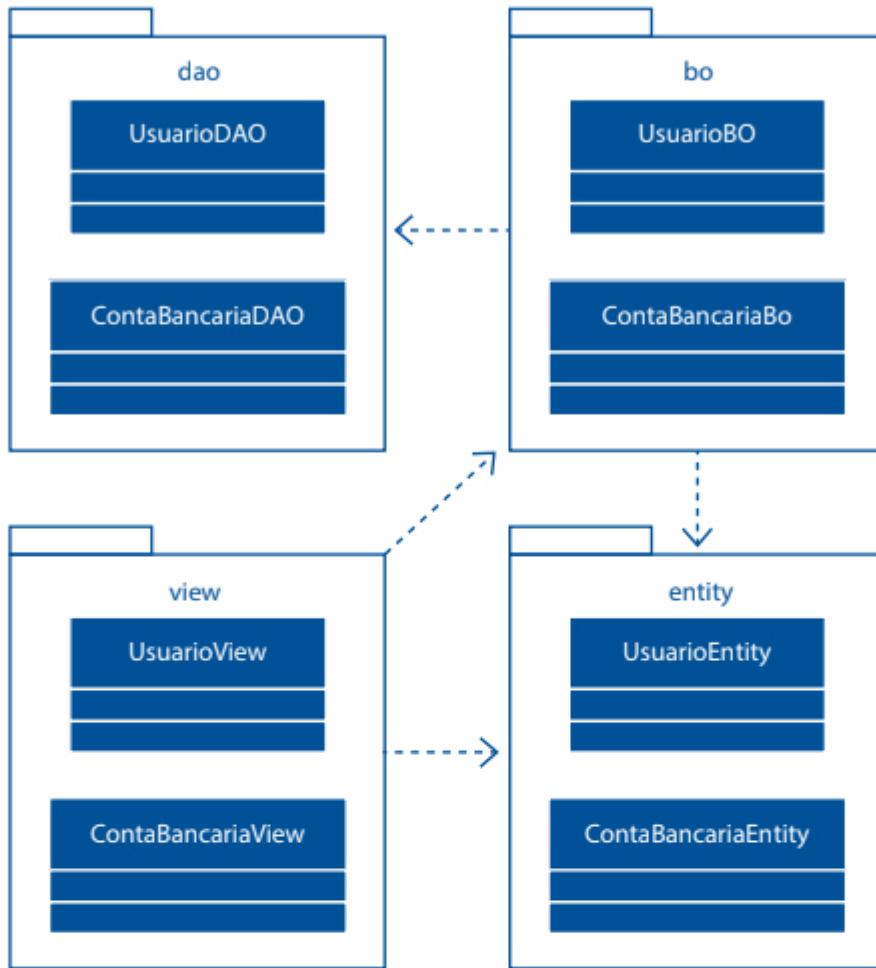
Lembre-se de que, quando for trabalhar com projetos reais de desenvolvimento de sistemas, você deverá entender que os arquivos-fontes não ficam aleatoriamente espalhados dentro do sistema, mas sim organizados em diretórios específicos (pacotes), que melhor definem a função para aquele arquivo (como, por exemplo, uma classe).

E acredite, não é difícil você encontrar sistemas que possuem milhares (sim, milhares) de classes criadas para desempenharem diversas funções específicas. Agora, imagine se todos esses arquivos ficassem dentro de uma única pasta, sem nenhuma lógica de organização por negócio ou propósito. Ficaria muito confuso e difícil trabalhar em um projeto organizado dessa maneira, você não concorda?

Por isso, é utilizado o diagrama de pacotes, que consegue traduzir como será a organização do sistema, estabelecendo, a partir de uma lógica razoável, uma ordem e disposição dos arquivos.

No exemplo a seguir, você vai ver como poderia, em um projeto, desenvolver um sistema que teria quatro pacotes principais de organização, sendo eles:

- a) **Pacote DAO:** responsável pelas classes que farão a interação e comunicação com o banco de dados;
- b) **Pacote BO:** camada responsável pelas validações de negócio e para fazer a conexão entre a camada DAO e a camada VIEW;
- c) **Pacote ENTITY:** camada para representar todas as entidades que serão desenvolvidas no sistema;
- d) **Pacote VIEW:** camada responsável por apresentar ao usuário a forma de interagir com o sistema. É a partir dela que o usuário efetua ações, que são levadas para a camada BO e, posteriormente, para a camada DAO.



Davi Leon (2019)

Figura 11 - Exemplo de diagrama de pacotes
Fonte: Do Autor (2019)

DIAGRAMA DE ESTRUTURA COMPOSTA

O penúltimo diagrama da estrutura que você vai conhecer é o diagrama de estrutura composta, responsável por mostrar a estrutura interna dos classificadores estruturados, que, para isso, utiliza em sua notação gráfica: peças, portas e conectores. Vamos a ele!

Um diagrama de estrutura composta é, de certa forma, parecido com um diagrama de classe, porém representa peças individuais em vez de classes propriamente ditas. A ideia é que você possa modelar as peças que representam as instâncias que o classificador contido possui e incluir conectores para vincular duas ou mais peças em um relacionamento de associação ou dependência.

Para entender melhor, veja um exemplo de diagrama de estrutura composta que procura identificar um classificador contido: automóvel. Da mesma forma, ele mostra quatro peças internas do classificador contido e procura representar as quatro rodas do carro (frontal esquerda e direita, traseira esquerda e direita). Você também perceberá uma conexão de comunicação que conecta as rodas dianteiras e as traseiras, a partir de conectores chamados de eixo frontal e eixo traseiro.

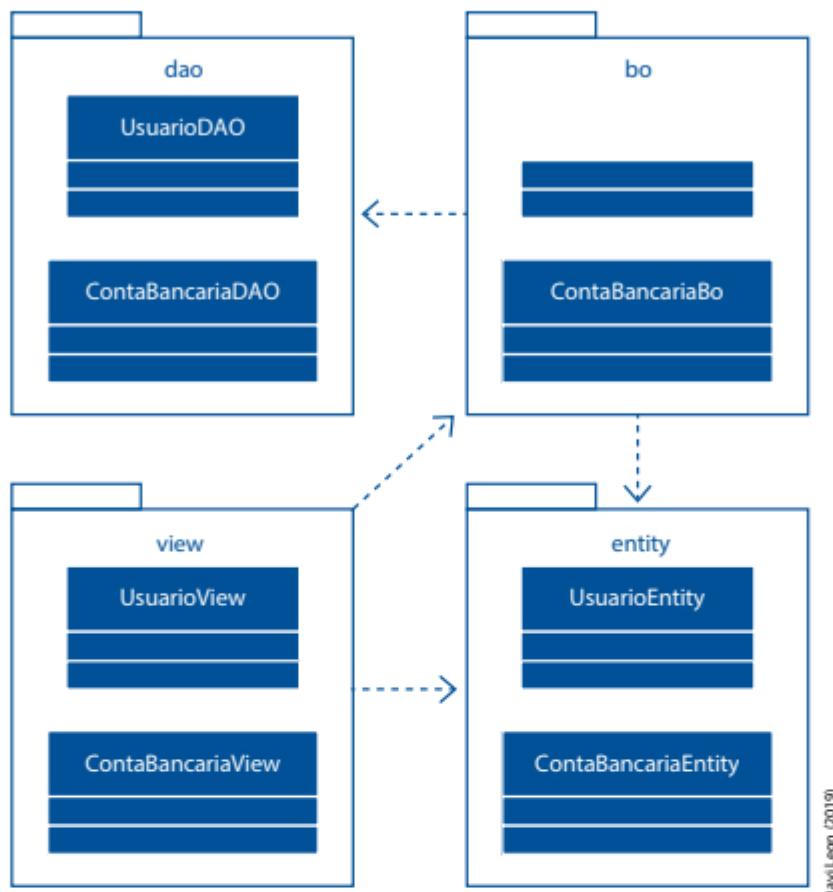


Figura 12 - Exemplo de diagrama de estrutura composta
Fonte: Do Autor (2019)

David Leon (2019)

DIAGRAMA DE PERFIL

O último diagrama estrutural que você vai estudar é o diagrama de perfil, basicamente um mecanismo de extensibilidade que permite estender e personalizar a UML, adicionando novos blocos de construção. Da mesma forma, ele procura criar propriedades e especificar nova semântica, a fim de tornar a linguagem adequada ao seu domínio de problema específico.

O diagrama de perfil possui três tipos de mecanismos de extensibilidade, sendo eles:

- Estereótipos (*stereotypes*);
- Valores marcados (*tagged values*);
- Restrições (*constraints*).

Mas o que representa cada um deles? Os estereótipos (*stereotypes*) permitem aumentar o vocabulário da UML. Você pode adicionar, criar elementos de modelo, derivados dos existentes, mas que possuem propriedades específicas adequadas ao seu domínio de problema. São usados para introduzir novos blocos de construção ausentes na língua do seu domínio ou que parecem primitivos. Além disso, possibilitam que você introduza novos símbolos gráficos, por exemplo: ao modelar uma rede, você pode precisar de símbolos para <<roteador>>, <<switches>>, <<hub>>, etc.



Scyther5 (2017)

Os valores marcados (*tagged values*) são usados para estender as propriedades da UML para que você possa incluir informações adicionais na especificação de um elemento de modelo. Ele permite que você especifique pares de valores de palavras-chave de um modelo em que estas são os atributos. Os valores marcados são renderizados graficamente como um literal entre colchetes. Por exemplo: considere uma equipe de liberação responsável pela montagem, teste e implantação de um sistema. Nesse caso, é necessário manter o controle sobre a versão e os resultados do teste do subsistema principal. Valores marcados são usados para adicionar essas informações.

Já as restrições (*constraints*) são as propriedades para especificar algumas condições que devem ser mantidas verdadeiras a todo momento. Elas permitem estender a semântica do bloco de construção UML, adicionando novos protocolos. Graficamente, uma restrição é renderizada como um literal entre chaves, colocado próximo ao elemento associado. Por exemplo: no desenvolvimento de um sistema em tempo real, é necessário adornar o modelo com algumas informações necessárias, como o tempo máximo de resposta de uma função.



David Leon (2019)

Figura 13 - Exemplo de restrição
Fonte: Do Autor (2019)

Ficou mais fácil entender o diagrama de perfil com o exemplo? Conheça, a seguir, o diagrama de caso de uso.

DIAGRAMA DE CASO DE USO

Finalmente você chegou aos diagramas de comportamento, e já vamos iniciar com um dos diagramas da UML. O diagrama de caso de uso possui uma aderência muito forte com o documento de requisitos de sistema, pois trabalha justamente com o que o sistema deve fazer (caso de uso) e com quem pode executar essa ação (ator). Se você notar algumas propostas de levantamento de requisitos, observe que é exatamente essa a relação que se busca:

- a) O que o sistema deve fazer;
- b) Quem faz essas ações no sistema.

Para que você entenda melhor esse importante diagrama, vamos trabalhar em um pequeno exemplo prático, supondo que você está criando um sistema para uma videolocadora. Clientes (mensalistas e usuais) locam e solicitam reservas de filmes, e são atendidos por um funcionário que, além de cuidar do catálogo de filmes da locadora, efetua ações no cadastro dos clientes, das reservas e locações dos filmes. Pensando nesse cenário, você poderia ter o seguinte diagrama de caso de uso:

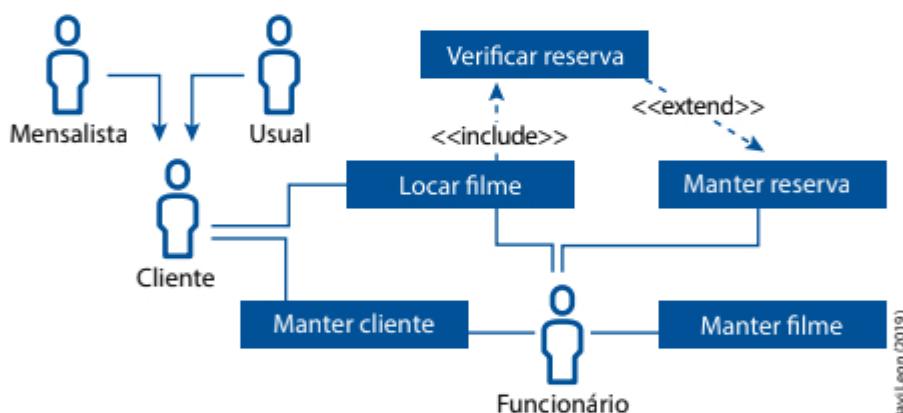


Figura 14 - Exemplo de diagrama de caso de uso
Fonte: Do autor (2019).

Davi Leon (2019)

Com esse exemplo é possível fazer conexões diretas do significado do diagrama. É possível ver os clientes mensalistas e usuais, o funcionário e suas atividades, como reservar ou locar um filme, manter as informações sobre algum filme, manter os dados da conta do cliente, etc.

Chegou a hora de entender cada um dos elementos existentes no diagrama, para que você possa ter um profundo conhecimento desse importante instrumento de modelagem.

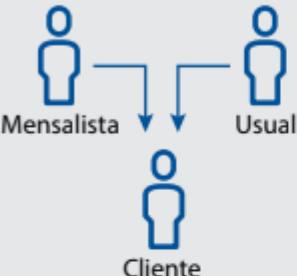
ELEMENTO DO DIAGRAMA	EXPLICAÇÃO
 Funcionário	Aqui você tem um ator do sistema, denominado "Funcionário". Este ator é capaz de executar cinco casos de uso, quatro de forma direta e um de forma indireta. É possível ver que ele pode executar os casos "Locar Filme", "Manter Cliente", "Manter Filme", "Manter Reserva" e, obrigatoriamente, quando executar uma locação, o sistema sempre executará o "Verificar Reserva".
 Mensalista Usual Cliente	Agora você tem aqui um caso de generalização (ou herança). Temos o ator "Cliente", que pode ser um "Mensalista" ou um "Usual". Ou seja, todo mundo é um cliente, mas ele pode ser um cliente que paga uma mensalidade e, em contrapartida, pode pegar x filmes todo mês. Ou ele é aquele cliente usual, que paga por filme locado individualmente. Você deve entender que tanto "Mensalista" quanto "Usual" vão possuir as mesmas informações existentes em "Cliente", porém individualmente eles poderão ter algumas informações diferenciadas.
Manter cliente	Eis o primeiro caso de uso que vamos falar, o "Manter Cliente". É o caso de uso responsável por manter os dados do cliente. Podem acessar esse caso de uso dois atores: o cliente e o funcionário. Obviamente, na descrição deste caso, eu destacaria que o cliente só poderá manter os seus próprios dados, enquanto o funcionário poderá manter os dados de todos os clientes.
 Locar filme Verificar reserva Manter reserva	Agora você vai ver esses três casos de uso, "Locar Filme", "Verificar Reserva" e "Manter Reserva", pois eles possuem um peculiar relacionamento. O caso de uso "Locar Filme" pode ser feito pelo cliente ou funcionário. Porém, em ambos os casos esse caso de uso incluirá sempre (<i>include</i>) a necessidade de executar o caso de uso "Verificar Reserva", para verificar se aquele filme específico possui reserva agendada, o que impediria a sua locação. Já na relação entre "Verificar Reserva" e "Manter Reserva" é algo opcional (<i>extend</i>), ou seja, essa função pode estender-se a outra função opcionalmente. Então, quando você está verificando uma reserva, é possível mantê-la (editar, deletar, etc.). Já o "Manter Reserva" é feito exclusivamente pelo funcionário, que consegue manter todas as reservas dos filmes por solicitação dos clientes.
Manter filme	Por fim, mas não menos importante, você vai ter o caso de uso "Manter Filme" realizado também exclusivamente por funcionários. Aqui serão mantidas todas as informações dos filmes.

Tabela 7 - Tabela de explicação dos elementos do exemplo de diagrama de caso de uso
Fonte: Do autor (2019).

Agora que você entendeu um pouco melhor os elementos que formam um bom diagrama de caso de uso, chegou a hora de compreender realmente como funciona a elaboração do documento que conterá os

casos de uso, pois você vai ver com detalhes o diagrama, de tal forma que todas as questões relacionadas aos atores e aos respectivos casos de uso podem ser explicadas da melhor forma possível.

Entenda que o documento de caso de uso é muito mais do que apenas a parte gráfica (o desenho do diagrama), mas comprehende as relações e explicações relativas a cada uma dessas partes formadoras do diagrama. Observe, na tabela anterior, que foram explicadas várias relações entre atores diferentes para um mesmo caso de uso. Então, quando você for trabalhar com casos de uso, na verdade, terá que entender que se faz necessário elaborar documentos detalhados.

Veja, a seguir, algumas propostas de como poderíamos elaborar um documento de caso de uso.

CASO DE USO	
DESCRÍÇÃO	
ATOR	
PRÉ-CONDIÇÃO	
FLUXO DO CASO DE USO	
ATOR	SISTEMA
FLUXO ALTERNATIVO DO CASO DE USO	
ATOR	SISTEMA
PÓS-CONDIÇÃO	

Tabela 8 - Modelo proposto para estudo dos casos de uso
Fonte: Do autor (2019).

Para potencializar o seu aprendizado, preencha essa proposta de modelo, levando em consideração o caso de uso da locadora de filmes. Vamos utilizar os casos de uso "Manter Filme" e "Manter Reserva", ambos utilizados pelo ator "Funcionário".

CASO DE USO	"Manter Filme"
--------------------	----------------

Descrição	O objetivo deste caso de uso é manter os filmes disponíveis na locadora. Manter significa salvar, editar, listar e remover, adicionando todas as particularidades e informações exigidas para estabelecer o sucesso do negócio da locadora.
Ator	Funcionário
Pré-Condição	Nenhuma
FLUXO DO CASO DE USO	
Ator	Sistema
1. Funcionário mantém as informações do filme: título, descrição, categoria, preço para locação e quantidade de cópias disponíveis.	
	2. Sistema gera um código único para o filme.
3. Funcionário cola etiqueta com código único do filme na capinha do filme.	
FLUXO ALTERNATIVO DO CASO DE USO	
Ator	Sistema
1. Funcionário não informa campos obrigatórios para manter o filme: título, categoria, preço e quantidade.	
	2. Sistema não efetua a ação de salvar e envia uma mensagem ao usuário informando os campos obrigatórios.
3. Funcionário insere valor zero ou inferior a zero no preço do filme.	
	4. Sistema informa que os valores de locação para os filmes devem ser maior do que zero.
5. Funcionário insere valor zero ou inferior a zero na quantidade de cópias disponíveis.	
	6. Sistema informa que os valores de cópias disponíveis de filmes devem ser maior do que zero.
Pós-Condição	Um filme é mantido no sistema, possuindo as informações obrigatórias cadastradas: título, categoria, preço e quantidade.

Tabela 9 - Exemplo de uso para modelo proposto de estudo dos casos de uso
Fonte: Do autor (2019).

DIAGRAMA DE TRANSIÇÃO DE ESTADOS

O diagrama de transição de estados (ou diagrama de estado da máquina) é o segundo diagrama de comportamento que você vai conhecer aqui. Ele procura representar uma determinada situação em que um objeto pode estar em um determinado momento da execução dos processos do sistema. Você poderá, então, analisar o que se passa entre um determinado estado inicial e final, por meio da leitura do que ocorre durante essa transição.

São elementos que compõem esse diagrama de comportamento:

- a) Estado:** condição ou situação durante a vida de um objeto na qual ele satisfaz algumas condições, executa algumas atividades ou espera por eventos;
- b) Transição:** o relacionamento entre dois estados, indicando que o objeto que está no primeiro estado passará para o segundo mediante a ocorrência de um determinado evento e, em certos casos, de uma condição;
- c) Condição:** causa necessária para que haja a transição de estado. Decorre de um evento ou circunstância que propicia a transição de estado;
- d) Estado inicial:** estado por onde se começa a leitura de um diagrama de estado;
- e) Estado final:** estado que representa o fim de uma máquina;
- f) Barra de sincronização:** semelhante a um garfo (fork) do diagrama de atividade;
- g) Estado composto:** estado composto por outras máquinas de estado organizadas em regiões que são executadas em paralelo;
- h) Sincronização:** permite que os relógios de dois ou mais processos paralelos estejam sincronizados em um determinado momento do processo;
- i) Ação:** atividade do sistema que efetua a transição de estado.

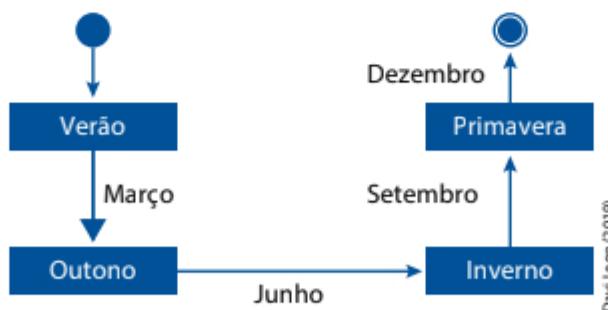


Figura 15 - Exemplo de diagrama de transição de estado – estações do ano
Fonte: Do autor (2019).

DIAGRAMA DE ATIVIDADE

Você chegou ao terceiro diagrama de comportamento, o diagrama de atividade, que, como o próprio nome procura explicar, define as atividades e seus respectivos fluxos que existem em um determinado processo que vai ocorrer no sistema.

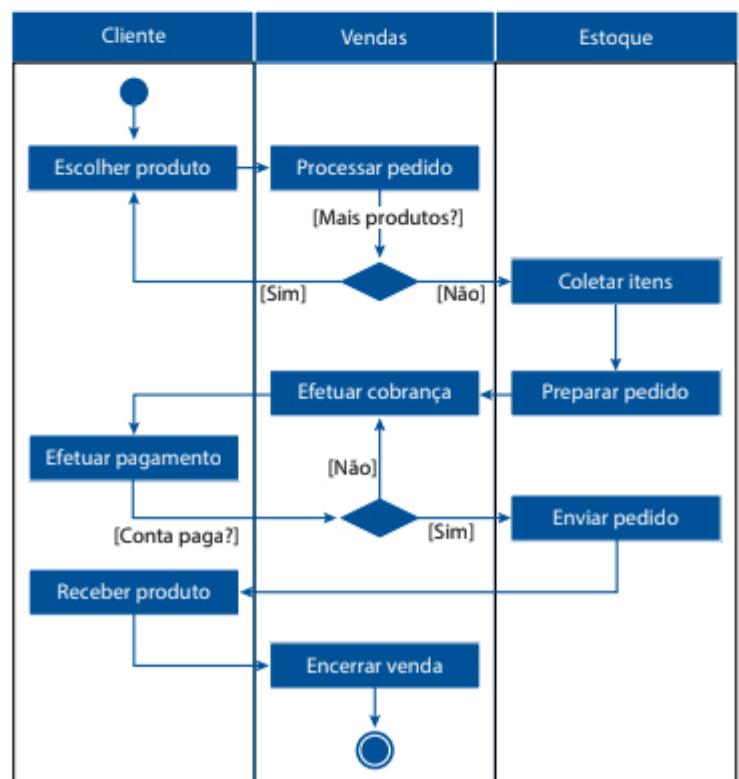
Quando você estudou no capítulo 2.1 sobre regra de negócio e no 2.2 sobre levantamento de requisitos, observou que diversas ações de negócios pertinentes ao escopo da modelagem de um possível sistema deveriam ser mapeadas. Essas ações são, em sua maioria, traduzidas nos diagramas de atividades como fluxos mapeados.



FIQUE ALERTA

É importante você entender que o trabalho de mapeamento das regras de negócio, do levantamento dos requisitos e, posteriormente, da aplicação dos diagramas da UML estão extremamente conectados e são interdependentes. Quando se fala em modelagem de sistemas, estamos falando exatamente da aplicação de diversas técnicas, métodos e artefatos que procuram efetivar o sucesso de um projeto de sistemas.

Para que você entenda melhor os elementos que compõem o diagrama de atividades, vamos criar um pequeno exemplo que envolva uma atividade de venda solicitada por um cliente e processada por uma área de vendas e de estoque. Acompanhe a figura seguinte!



Davi Leon (2019)

Figura 16 - Exemplo de diagrama de atividade – venda de produtos
Fonte: Do autor (2019).

Aqui é apenas um exemplo, que poderia ser muito mais detalhado, com novas regras de decisão, condições, junções e vários outros recursos que um diagrama de atividade comporta. Mas o mais importante nesse momento é que você compreenda que cada atividade mapeada durante a modelagem deverá ser representado em um diagrama de atividade como este.

Fazem parte da composição gráfica do diagrama de atividade os seguintes elementos disponíveis:

- a) **Atividade**: comportamento a ser realizado;
- b) **Subatividade**: execução de uma (ou mais) atividade(s), a partir de uma atividade;
- c) **Transição**: fluxo de uma atividade para outra;
- d) **Ação**: transformação;
- e) **Decisão**: dependendo de uma condição, mostra as diferentes transições;
- f) **Raia**: diferenciação de unidades organizacionais;
- g) **Bifurcação (fork)**: separa uma transição em várias transições executadas ao mesmo tempo;
- h) **Sincronização (join)**: concatenação de transições vindas da bifurcação;
- i) **Objeto**: o objeto da atividade;
- j) **Envio de sinal**: transição para um meio externo;
- k) **Recepção de sinal**: recepção de um meio externo;
- l) **Região**: agrupamento de uma ou mais atividades;
- m) **Exceção**: atividades que ocorrerem em decorrência de uma exceção.

Durante a função de modelagem de sistemas, você vai notar futuramente que os diagramas de atividade são excelentes recursos para serem utilizados nos encontros com os clientes, exatamente pelo fato de eles colaborarem com uma visão comportamental (regras de negócio) do sistema, em vez de uma visão estrutural (um pouco mais técnica).

O bacana deste diagrama é o potencial mapeamento de várias atividades e suas respectivas particularidades de negócio que são explicitadas. Fica fácil para um programador, por exemplo, observar as pertinências de decisão, caminhos alternativos que o negócio pode tomar e como as possíveis funções do sistema interagirão entre si.

DIAGRAMA DE SEQUÊNCIA

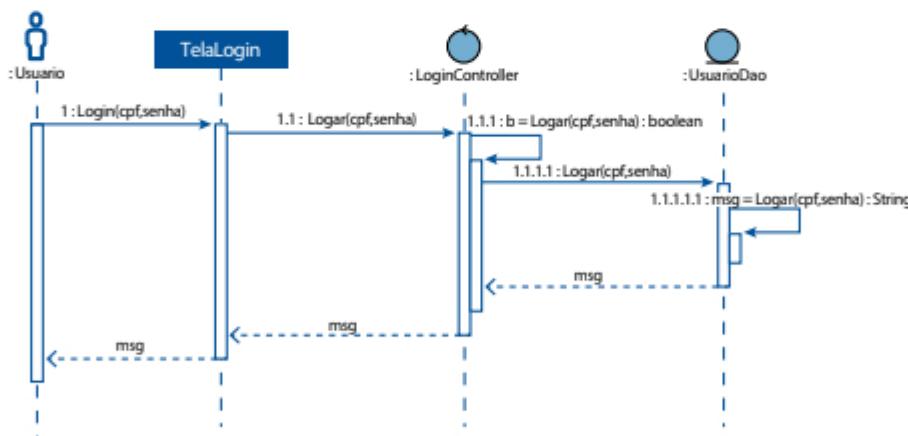
O décimo primeiro diagrama que você vai estudar e que é também o quarto diagrama comportamental é o de sequência, que procura representar a sequência de processos em um sistema. Essas sequências são expressadas como se fossem mensagens entre as classes, em sua maioria executadas pelos métodos (funções ou procedimentos) para efetivarem o objetivo de um determinado caso de uso.

Observe essa condição mandatória do diagrama de sequência, onde cada caso de uso que foi mapeado nos diagramas de caso de uso deverá possuir um diagrama de sequência. Assim sendo, você vai apenas construir esse diagrama quando todo o trabalho de construção e validação dos diagramas de caso de uso forem finalizados.

Pode-se entender, então, que o estudo de caso de uso é pré-requisito obrigatório da execução do diagrama de sequência. Ele se propõe a explicar como será o comportamento do sistema para dentro de suas classes, utilizando seus métodos, concretizando os objetivos propostos nos casos de uso e respeitando as regras de negócio e requisitos mapeados.

São elementos que compõem o diagrama de sequência:

- a) Atores:** representa o ator primário que é o responsável por enviar a mensagem inicial que inicia a interação entre os objetos;
- b) Objetos:** é o uso de uma classe (em sua instância), existente no sistema que será utilizada neste processo;
- c) Gate:** este recurso existe para definir um determinado ponto em que a mensagem pode ser transmitida para dentro ou para fora;
- d) Fragmento:** definem algumas interações: alternativa (*alt*), opcional (*opt*), parar (*break*), repetição (*loop*);
- e) Linha de vida:** uma linha de vida é composta de duas partes: uma linha tracejada e um retângulo que se estenderá por essa linha tracejada. O tamanho do retângulo define a quantidade de participação de um ator, componente, classe, etc. Ou seja, você terá a visão das partes e como elas participaram proporcionalmente daquele caso de uso.



Davi Leon (2019)

Figura 17 - Exemplo de diagrama de sequência – caso de uso efetuar login
Fonte: Do autor (2019).

É possível contemplar nesse exemplo o ator "Usuário", que a partir de uma tela de login fornece seu "CPF" e "Senha" para se autenticar no sistema. Por sua vez, essa tela de login se comunica com um controlador, enviando esses dados, que são validados pelo controlador. O controlador aqui tem o fundamental papel de validar esses dados, antes de enviar para a camada que efetivamente buscara o usuário no banco de dados. Isso evita, por exemplo, que um CPF ou senha inconsistente (por exemplo: em branco, parcial, inválida, etc.) chegue na camada de acesso ao banco de dados.

Uma vez que se efetivam todas as funções, o sistema retornará uma mensagem, dando validade ou invalidade ao acesso do usuário em questão no sistema. Note o comportamento sequencial que este diagrama dá a cada uma das mensagens. Observe, então, a possibilidade de mapeamento de como as regras de negócio irão se traduzir em possíveis métodos no sistema e qual será o provável caminho comportamental de um caso de uso. Esse é o grande legado que um diagrama de sequência fornece para a modelagem de sistema, uma excelente forma de rastrear a sequência comportamental dos casos de uso.

DIAGRAMA DE VISÃO GERAL DE INTERAÇÃO

O diagrama de visão geral da interação (do inglês *Interaction Overview Diagram*) é, em sua essência, um diagrama comportamental que une o diagrama de caso de uso com o de sequência, que você acabou de estudar. Ele apresenta a interação do envio ou do recebimento de dados entre um ator e um caso de uso.

Você pode utilizar, basicamente, dois tipos de quadros: o quadro de interação, que apresenta algum diagrama de sequência inteiro, ou um quadro de referência, quando queremos apenas referenciar um determinado diagrama de sequência. Lembre-se de que quando você trabalha com boas ferramentas de modelagem, é possível utilizar os quadros de referência, fazendo inclusive um *link* dentro do projeto de modelagem, com o diagrama referenciado.

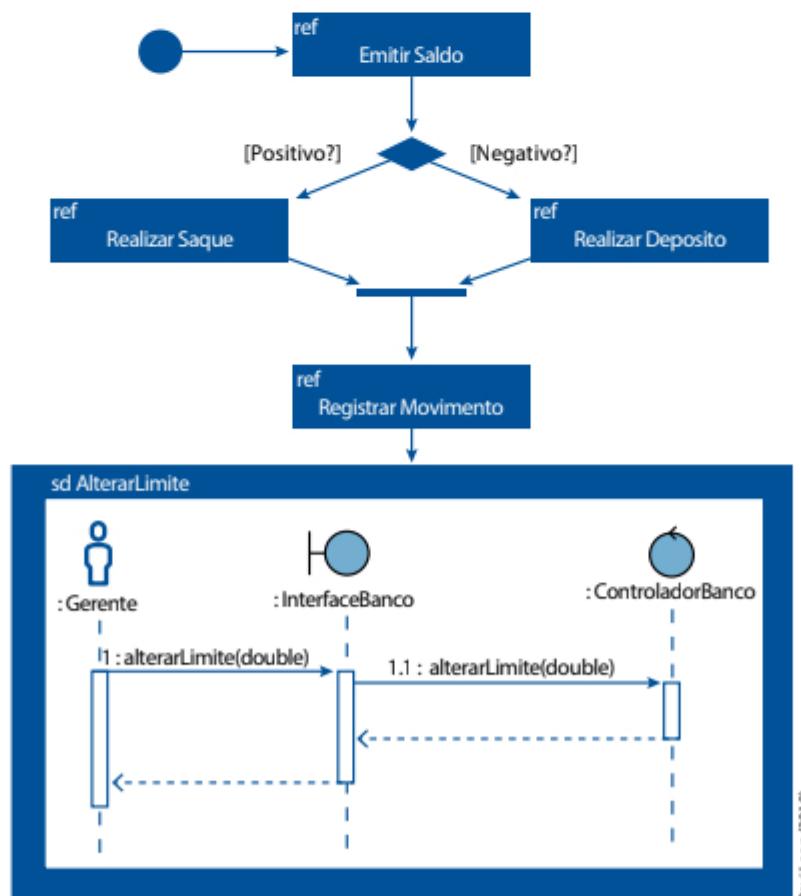


Figura 18 - Exemplo de diagrama de visão geral de interação
Fonte: Do autor (2019).

No exemplo anterior, você tem um diagrama de visão geral de interação, que demonstra um excelente panorama de quais diagramas participam quando um determinado conjunto de negócios acontece. Observe que, na imagem, estão presentes cinco diagramas dentro do diagrama de visão geral de interação. Desses, quatro são diagramas apenas por referência (quadro de referência) e outro colocado de forma explícita (quadro de interação).

DIAGRAMA DE COMUNICAÇÃO

Agora você vai aprender sobre o penúltimo diagrama e sexto diagrama de comportamento. O diagrama de comunicação, que até as versões passadas da UML era chamado de diagrama de colaboração, é um diagrama que procura mostrar, de forma parecida com o diagrama de sequência, a dinâmica colaboração entre os objetos em um determinado contexto do sistema. Você deve entender que os critérios para adotar um diagrama de sequência ou de comunicação se dará na ênfase que queremos dar para o diagrama. Se no caso esta ênfase for o decorrer do tempo, então você deve escolher o diagrama de sequência. Por outro lado, se a ênfase for o contexto do sistema, então nossa escolha deverá ser o diagrama de colaboração.

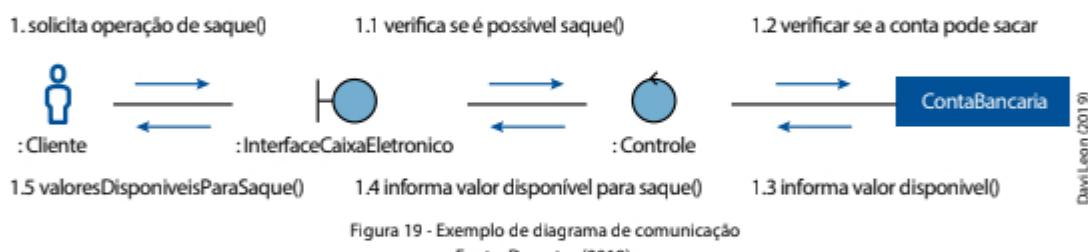


Figura 19 - Exemplo de diagrama de comunicação
Fonte: Do autor (2019).

David Leon (2019)

Fica claro, quando você observa o exemplo anterior, que o foco não é o tempo ou a proporção de interação que levará cada entidade envolvida na operação saque, mas sim como ocorrem a comunicação dentro do contexto deste negócio mapeado. Vamos ao último tipo de diagrama? Siga atento!

DIAGRAMA DE TEMPO

Eis que você chegou ao último diagrama que você vai conhecer no universo da UML. Ele é o sétimo diagrama de comportamento, e saiba que ele é bem recente dentro da UML, chegando apenas na versão 2.x.

O diagrama de tempo procura nos apresentar o comportamento de determinados objetos e suas respectivas interações em uma régua de tempo, demonstrando as condições que podem mudar ao longo desse período.

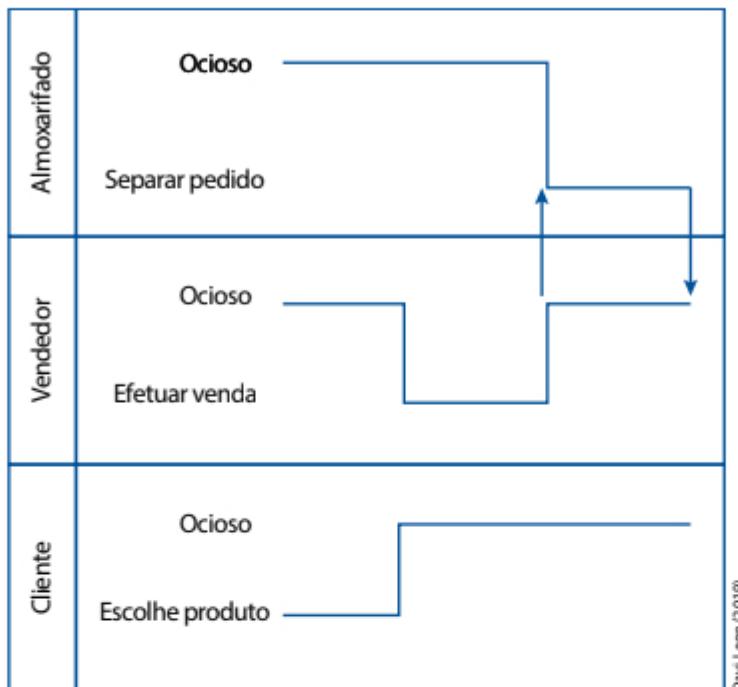


Figura 20 - Exemplo de diagrama de tempo
Fonte: Do autor (2019).

Davi Leon (2019)

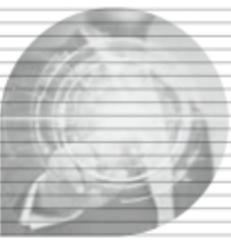
No exemplo é possível observar que os atores “Almoxarifado”, “Vendedor” e “Cliente” alternam na linha do tempo entre algumas situações de ociosidade e de atividade.

No diagrama você pode ver que o “Cliente” começa a ação de escolher produto, depois disso fica ocioso e aguarda o “Vendedor” sair da ociosidade e efetuar a venda do produto escolhido. Posteriormente, uma mensagem é enviada para o “Almoxarifado”, que efetua sua ação de separar o pedido e assim por diante, até que temporalmente todas as ações desse negócio se concluam.



RECAPITULANDO

Neste importante capítulo você aprendeu sobre a mundialmente utilizada UML, um grande padrão de linguagem de modelagem unificado. Viu que os diagramas se agrupam em diagramas de comportamento e diagramas estruturais. Também foi possível perceber que cada um desses diagramas possuem forte conexão ou entre eles mesmos ou com os outros artefatos e técnicas que compõem a modelagem de um sistema, tal como o mapeamento de regras de negócios e/ou levantamento de requisitos.



Agora você já sabe como efetuar o mapeamento das regras de negócio e o levantamento de requisitos, da mesma forma que aprendeu sobre a UML e os diversos diagramas que a compõem. Mas, do que adiantaria tanto conhecimento sem uma característica fundamental para um bom profissional que modelará sistemas: a iniciativa?

Na verdade, a iniciativa é essencial em diversas áreas de atuação, pois conduz o profissional em um caminho de resultados e aprendizado constantes. A iniciativa é como uma mola propulsora que movimenta boa parte das descobertas, invenções e inovações disruptivas do tempo. Pessoas, que não se conformavam com a situação ou forma de trabalho atual, tiveram a iniciativa de pensar diferente.

É importante lembrar que não adianta ter apenas iniciativa se o seu trabalho não se mantiver constante e focado no resultado. E é exatamente sobre esses comportamentos de iniciativa e organização que você vai conhecer neste capítulo. Você verá a importância de ser proativo, demonstrar iniciativa e ter autonomia para o sucesso na modelagem de sistemas.

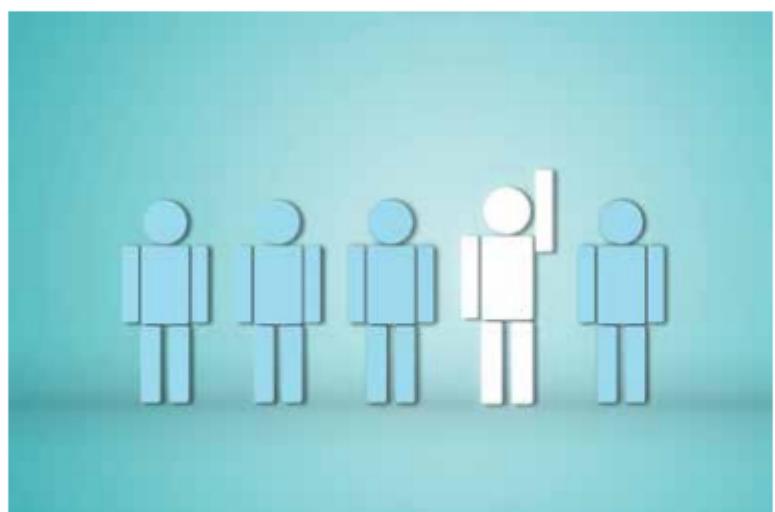
Ao final deste capítulo você terá subsídios para:

- a) Reconhecer a iniciativa como característica essencial para o desenvolvimento profissional;
- b) Conhecer formas de demonstrar iniciativa.

Que tal iniciar os estudos explorando formas de demonstrar iniciativa? Lembre-se que esses conteúdos são essenciais para todo bom profissional em atuação no mercado de trabalho.

5.1 FORMAS DE DEMONSTRAR INICIATIVA

É fundamental que você, como futuro profissional que atuará com modelagem de sistemas, tenha a característica da iniciativa em seus comportamentos, pois, como você estudou nos tópicos anteriores, diversas atividades que desempenhamos para modelar necessitam de um contato constante com os clientes e usuários. E você, enquanto profissional diferenciado, deve ter sempre a iniciativa de buscar o melhor resultado a partir de um trabalho metódico, formal e organizado.



Thirinawat_S ([20-7])

Agora você já entendeu como é importante ter iniciativa na atuação profissional de modelagem, não é mesmo? Mas como podemos demonstrar iniciativa? Como podemos desenvolver atitudes que demonstrem efetivamente que estamos tendo o comportamento de iniciativa? Bem, em algum momento de sua vida profissional você já deve (ou vai) ter entrado em contato com uma descrição de vaga para processo seletivo em que as características de um perfil com iniciativa são exigidas ou desejadas, certo? Normalmente as empresas expressam esse comportamento de iniciativa em diversas atitudes desejáveis, tais como:

- a) Ser proativo;
- b) Engajado;
- c) Interessado em aprender coisas novas;
- d) Gostar de novos desafios;
- e) Não ter medo de mudanças;
- f) Dedicado.

Se você observar bem, notará que uma pessoa com atuação voltada à iniciativa detém um comportamento semelhante às atitudes apresentadas. Exatamente pelo fato de que a iniciativa é combustível para um comportamento proativo, a pessoa com iniciativa tornar-se-á sempre mais engajada,

demonstrará normalmente maior interesse pelos conhecimentos novos, não possuirá aversão aos novos desafios ou mudanças e, por fim, apresentará um comportamento dedicado.

Portanto, é possível chegar à conclusão de que é preciso demonstrar um perfil profissional com iniciativa e apresentar diversas atitudes que possuam natural aderência a essa qualidade. Quando você atua constantemente com essas atitudes invariavelmente em algum momento alguém irá te rotular como "aquele profissional que possui iniciativa".

5.2 RESULTADO

Agora você já sabe claramente que a iniciativa é algo essencial para atuar profissionalmente e que é um parâmetro comportamental muito exigido pelas empresas. Porém, do que vale a iniciativa se não alcançar os objetivos (ou resultados esperados)? Já imaginou alguém que sempre sai bem na largada em uma corrida de carros, porém nunca termina a prova? Essa pessoa nunca será campeã, pois nunca pontuará por não completar o principal requisito de uma corrida, que é cruzar a linha de chegada.

Se por um lado você deve ter sempre iniciativa, demonstrando diversos comportamentos como proatividade, engajamento, etc., por outro lado você tem que ser resiliente para se manter sempre firme na busca do resultado (e objetivo) final. Apesar de sempre termos importantes aprendizados ao longo da jornada, é o resultado que coroa o profissional diferenciado.



FIQUE ALERTA

Tenha sempre em mente que, além de ter a iniciativa de dar o primeiro passo nessa grande corrida profissional, é preciso se manter firme e estratégico, determinado em alcançar o objetivo final, personificado no resultado.



Voltando a essa bela metáfora da corrida, em que a iniciativa é exatamente se engajar na prova e dar o primeiro passo, e o objetivo final é cruzar a linha de chegada (resultado), é fundamental que o atleta entenda o que enfrentará pela frente e quais os desafios presentes na prova, por exemplo:

- a) Qual a distância que correrei?
- b) Quais as condições do terreno?
- c) Qual a previsão climática no dia da prova?
- d) Meus equipamentos estão em ordem e preparados?
- e) Qual será minha estratégia adotada durante a corrida?

Esses são apenas alguns exemplos de possíveis questionamentos que o preparam melhor para participar de uma prova de corrida. Note, então, que não basta apenas receber um desafio e sair correndo no impulso, sem planejamento, sem pensar e sem se preparar. Algumas pessoas cometem esse grande equívoco quando o assunto é iniciativa, pois confundem iniciativa com impulsividade. Iniciativa é tomar uma decisão de começar algo, mas a decisão é tomada de forma coerente e pensando nas questões inerentes ao desafio. Já impulsividade é simplesmente ter a atitude de começar, sem levar em conta as variáveis que estão em jogo. E isso, muitas vezes, pode ser muito perigoso.

Não é difícil encontrar empresas que possuem indicadores voltados para os resultados. Essas empresas procuram medir a performance das áreas, setores, gestores e dos colaboradores a partir do encadeamento dos resultados, promovendo, inclusive, a política de bonificação para os melhores desempenhos.



**SAIBA
MAIS**

Os indicadores de desempenho organizacional são ferramentas básicas utilizadas para o gerenciamento do sistema organizacional. Com esses indicadores, as empresas são capazes de aferir os processos organizacionais e apontar possíveis alterações nos planejamentos já definidos.

E falando especificamente do foco do seu estudo, que é a modelagem de sistemas, você sabe dizer em que ponto se misturam e se identificam a iniciativa e o resultado? Bem, quando você tem a iniciativa de executar diversas atividades de levantamento de requisitos, tendo inclusive a proatividade de buscar por melhorias contínuas no processo de negócio do cliente, você está demonstrando claramente um comportamento de iniciativa. E quando você tem bem claro qual o objetivo daquela ação (Documento de Levantamento de Requisitos), está definindo qual é o resultado esperado. Ou seja, para ter sucesso alcançando o resultado, você terá que ter a iniciativa de gerar diversas interações com o cliente em busca da construção do Documento de Levantamento de Requisitos.

Mas é possível criar métodos que nos auxiliem na busca do resultado? Certamente que sim. Mas para tanto é sempre essencial que você tenha muito claro qual o objetivo que deve ser alcançado em cada etapa do trabalho. Você poderia, por exemplo, criar uma tabela que verificasse o desempenho na evolução do levantamento de requisitos.

ÁREA DE NEGÓCIO	USUÁRIOS-CHAVE	ENTREVISTADOS
Financeira	2	2
Recursos humanos	2	1
Contabilidade	1	1
Logística	4	2
Produção	8	2

Tabela 10 - exemplo de tabela de monitoramento de resultado.

Fonte: Do autor (2019)

Na tabela do exemplo, ficaria fácil efetuar o acompanhamento do resultado esperado versus a performance da atividade realizada. A tabela poderia ser traduzida da seguinte maneira:

- a) Que a performance de levantamento de requisitos da área financeira atingiu o resultado esperado (100%);
- b) Que a performance de levantamento de requisitos da área de recursos humanos está na metade do resultado esperado (50%);
- c) Que a performance de levantamento de requisitos da área da contabilidade atingiu o resultado esperado (100%);
- d) Que a performance de levantamento de requisitos da área da logística está na metade do resultado esperado (50%);
- e) Que a performance de levantamento de requisitos da área de produção ainda está longe do resultado esperado (25%);
- f) Que no total o levantamento de requisitos desse projeto está com 47,05% de evolução para alcançar o resultado esperado.


FIQUE ALERTA

É muito importante você saber o seu objetivo final, bem como os objetivos intermedios e iniciais de uma grande jornada. Só assim você conseguirá mapear cada uma das etapas e analisar sua performance de conclusão.

5.3 AUTONOMIA

Você sabe o que é autonomia? Essa palavra possui grande sentido quando se fala em ter um comportamento cheio de iniciativa. Segundo Immanuel Kant (1724-1804), autonomia define a capacidade da vontade humana de se autodeterminar segundo uma legislação moral por ela mesma estabelecida e livre de qualquer fator estranho ou exógeno com uma influência subjugante, tal como uma paixão ou uma inclinação afetiva incoercível.



tormentu

Então, é possível entender que a autonomia é nossa capacidade de tomar decisões pautadas em parâmetros que julgamos pertinentes e desvinculadas de influências emocionais. Essa capacidade é fundamental para os profissionais que atuam com tecnologia em análise, em desenvolvimento ou em modelagem.

Você deve lembrar que desenvolver um sistema é trabalhar com questões subjetivas, não é mesmo? A partir de diversos desejos dos clientes e usuários, você deve ter a capacidade de aplicar técnicas para extrair tudo aquilo que for pertinente às regras de negócio, para posterior coleta de requisitos e modelagem. Fazer isso de maneira desvinculada de questões emocionais é muito importante, pois, assim, é possível conservar um nível de coerência e igualdade da técnica aplicada.

É importante você entender também que ter iniciativa não é somente ter a liberdade de querer fazer da maneira que bem desejar. Ter iniciativa (em um contexto de modelagem) é trabalhar com autonomia, ou seja, ter a iniciativa de trabalhar com liberdade, mas guiado por uma conduta clara de parâmetros, técnicas e métodos.



CURIOSIDADES

Quando se fala em autonomia no universo automobilístico, é possível dar uma tradução totalmente diferente. Mas, ao mesmo tempo, trazemos um significado muito bacana para o universo de modelagem de sistemas, pois autonomia para um carro é a sua capacidade de percorrer uma distância em determinado tempo e sem que haja necessidade de reabastecimento.

Então, fazendo uma rápida analogia, é possível também pressupor a responsabilidade de criar um sistema novo quando você faz uma viagem, pois você deve estar convicto de que tem combustível (motivação) suficiente para percorrer todo o caminho, certo?

5.4 CONSEQUÊNCIAS FAVORÁVEIS E DESFAVORÁVEIS

Segundo algumas linhas de pensamento, tudo tem dois lados, proporcionais e inversos (positivo e negativo). Quando falamos em adotar um comportamento repleto de iniciativa, autônomo e focado no resultado, você invariavelmente saboreará o doce sabor da vitória, da mesma forma que também terá contato com o amargo sabor da derrota, com diversos momentos favoráveis e desfavoráveis.

Isso vai acontecer, principalmente, pelo fato de que as pessoas são diferentes, possuem ritmos, valores, competências e atitudes muito distintas umas das outras. E quando falamos em pessoas, estamos falando tanto dos clientes, parceiros, usuários, quanto de colegas de trabalho ou projeto. Se por um lado você não deve impor um comportamento próprio de iniciativa, você também não deve simplesmente largar mão do mesmo. Encontrar um bom equilíbrio entre suas motivações e as das outras pessoas é fundamental para sempre manter um ambiente salutar e criativo.



CASOS E RELATOS

Pessoas e comandos

Com mais de 20 anos de experiência profissional, José já presenciou profissionais muito competentes e disciplinados transformarem o convívio no trabalho em um ambiente pesado, insatisfatório e improdutivo. Ele descobriu que isso acontecia porque eles desejavam impor seus métodos como uma ditadura operacional.

Portanto, é importante lembrar que desenvolver bons sistemas computacionais é muito mais lidar com pessoas e seus desejos, do que efetivamente criar milhares de linhas de comando.



RECAPITULANDO

Neste capítulo você aprendeu a importância de manter um comportamento cheio de iniciativa com um ritmo autônomo e a sempre procurar a melhor forma de cadenciar esse comportamento proativo dentro dos múltiplos perfis com os quais você tem contato durante um projeto de modelagem de sistema, podendo ser usuários, clientes, parceiros ou colegas.

Organização de Dados

6



Quanto aprendizado importante sobre a modelagem de sistemas, não é mesmo? Você viu as regras de negócio, requisitos do sistema, UML e um perfil comportamental cheio de iniciativa e proatividade. Além disso, é fundamental desenvolver técnicas e comportamento de organização. Neste capítulo, você verá a importância da organização no contexto de modelagem de sistemas. Verá como é possível guardar, resgatar e, posteriormente, analisar os dados e informações pertinentes ao seu projeto de modelagem. Sendo assim, ao final deste capítulo você terá subsídios para:

- a) Reconhecer os princípios e normas da análise de dados;
- b) Aplicar os procedimentos da análise de dados;
- c) Organizar as atividades laborais diárias.

O primeiro assunto a ser estudado neste capítulo será a organização. Leia e veja a importância dessa capacidade.

6.1 A IMPORTÂNCIA DA ORGANIZAÇÃO

Imagine o seguinte caso: o que adiantaria ter a maior quantidade de livros do mundo em uma biblioteca, se esta não tivesse a capacidade de organizar esses livros? Ela poderia ser a "maior" biblioteca do mundo em quantidade de acervos, porém muito provavelmente seria a mais caótica e desorganizada do planeta.



Diagrama 6.1

A verdade é que pouco ajuda você ter uma grande quantidade de dados, informações e conhecimento, se não consegue organizá-los de forma coerente para serem resgatados e consultados de maneira apropriada.

Os tempos modernos cobram cada vez mais que os profissionais respondam às demandas no menor tempo hábil possível. Ninguém mais quer ficar esperando, porque tempo é um recurso muito valioso, pois é intransferível, esgotável e não se renova. Imagine, então, viver profissionalmente nos tempos atuais em meio a um ambiente sem organização. Isso é simplesmente impraticável nos dias de hoje, você concorda?

Por isso, a solução é a organização! Parece óbvio, não? Mas será que você está praticando o mesmo capricho para organizar seu trabalho? Será que você observa com cuidado cada atividade, trabalho, documento, evidência, histórico, páginas de internet, e-mail, informação, dado, etc.? Esses itens devem estar metodicamente preservados em uma lógica estável de organização. E essa lógica deve proporcionar condições razoáveis para que esses itens organizados possam ser resgatados a qualquer momento, da maneira mais rápida e prática possível.



CURIOSIDADES

Você sabe a diferença entre dados, informação e conhecimento? Notou que essas três palavras importantes foram usadas diversas vezes nesta unidade curricular e são importantes para a modelagem de sistemas? Dado é a fração mais primária de um significado, por exemplo: 1979. Informação já é a interpretação de um determinado dado, transformando-o em algo útil para um determinado contexto, por exemplo: 1979 é um ano. E conhecimento é quando você pega uma determinada informação e consegue construir um conhecimento a partir dessa informação, por exemplo: 1979 é o ano em que o autor desta unidade curricular nasceu.

O que você guarda comumente em um banco de dados (como o próprio nome sugere) são os dados, que normalmente são resgatados por um sistema, que os transforma em informação para o usuário. A partir dessas informações é possível, por exemplo, montar relatórios complexos que ajudam os usuários a criar conhecimento sobre um determinado contexto.

Você acredita que é possível construir um sistema onde os dados não estarão organizados? Ou gerenciar um projeto de modelagem de sistemas, onde todo dado recolhido se encontra espalhado aleatoriamente? Claro que você sabe que isso é inviável, como o caso da biblioteca desorganizada, pois não importará a quantidade de livros, se é impraticável achá-los em um tempo aceitável.



anyaberkut /20-70

E como você poderá organizar o seu projeto de modelagem de sistema? Você tem inúmeras formas (diria até incontáveis) de organizar seu trabalho. Obviamente você terá que encontrar seu caminho, pois as pessoas possuem características distintas e, no que tange à organização, cada um possuirá aptidão e aversão a determinadas técnicas. Mas aqui você verá algumas possibilidades.

Da mesma forma que o mundo moderno nos trouxe alguns desafios (o imediatismo, por exemplo), também nos trouxe diversas facilidades (internet, aplicativos, *smartphones*, etc.). É importante que você busque o máximo de apoio nesses recursos tecnológicos para dar vazão às necessidades de organização dos seus projetos. Existe uma infinidade de aplicativos, softwares e soluções on-line que auxiliam seu trabalho a ficar cada vez mais organizado e disponível.



CASOS E RELATOS

Soluções

Fernando conta com um ritmo de vida muito acelerado e é uma pessoa que abusa das soluções disponíveis para se organizar. É professor, consultor, assessor, empresário, empreendedor, analista, pai de família e jogador de esporte eletrônico. E não foram poucas as vezes em que ele atuava em todas as atividades ao mesmo tempo.

Agora com seus 40 anos, começou a focar naquilo que mais lhe interessava, dizendo não a muitas atividades e dando prioridade para aquilo que efetivamente o deixa realizado. Mas, mesmo assim, Fernando acaba tendo uma vida extremamente corrida. Para dar conta dela, ele utiliza diversos recursos para se organizar, por exemplo:

Agenda on-line: usa a agenda do Google para lançar todos os seus compromissos importantes, deixando-os disponíveis no celular e no navegador;

Pasta de arquivos on-line: usa o Google Drive, criando diversas pastas para seus projetos e trabalhos, tudo organizado e estabelecido com cores e nomes que ajudam a procurar e organizar;

HD externo: todos os projetos que já foram concluídos são colocados em um HD externo com backup para uma pasta no Google Drive;

Anotações: utiliza muito o Google Keep, para fazer todo tipo de lista e anotação necessária (de lembretes de trabalho até as compras no supermercado).

Dessa forma, Fernando consegue se organizar para dar conta das diversas atividades e atendê-las da melhor maneira possível.

Os critérios de escolha dessas soluções vão seguir critérios próprios, por exemplo:

- a) Preço;
- b) Disponibilidade on-line;
- c) Disponibilidade móvel.

Ou seja, segundo esses critérios, uma solução boa para a organização é gratuita, acessada on-line pelo navegador e pelo *smartphone*.

6.2 ROTEIRO DE TRABALHO

É muito importante, se não imprescindível, que você entenda a sequência e a dependência de todas as atividades que serão executadas em seu trabalho, pois só assim você terá a possibilidade de organizar essas atividades de forma coerente e ordenada.

Compreender também que essas atividades devem estar agrupadas dentro de determinadas etapas do seu projeto, vai ajudar ainda mais a organização do projeto, definindo os escopos de organização macro (etapas), micro (atividades) e o modo como cada um contribui para a conquista do resultado esperado.

Vamos a mais um exemplo! Imagine que você vai preparar um jantar, vai receber alguns convidados e precisa causar uma excelente impressão. Sendo assim, que tal organizar o jantar, estabelecendo as etapas e as atividades que devem compor o projeto "jantar"?

Você poderia, por exemplo, criar essas etapas do seu projeto:

- a) Cardápio;
- b) Compras;
- c) Pré-preparação;
- d) Preparo;
- e) Jantar;
- f) Pós-jantar.

E, dentro de cada etapa, você poderia criar diversas atividades, que, quando concluídas, efetivam o objetivo daquela etapa específica. Quando todas as etapas se cumprirem, você terá finalmente conquistado o resultado almejado. Sendo assim, o ideal é construir um documento que facilite o entendimento da sequência das etapas e o modo como as atividades estão distribuídas. Para melhor entendimento, vamos, dentro deste mesmo exemplo de jantar, imaginar uma pequena planilha que crie essa sequência organizada de etapas e atividades.

PROJETO		AUTOR	
DATA		VERSÃO	

Etapas previstas

Atividades previstas nas etapas

Análise das etapas

ETAPA	ANÁLISE SOBRE A ETAPA

Tabela 11 - exemplo de formulário de organização de etapas e atividades
Fonte: Do Autor (2019)

Se você analisar a proposta do formulário, vai perceber que, apesar de simples, ele é bem prático para o seu propósito, que é organizar e auxiliar na boa condução de um projeto. O documento é basicamente dividido em quatro sessões, sendo elas:

- a) Cabeçalho de identificação:** aqui você poderá documentar o nome que melhor define este projeto, o autor, a data de criação e em qual versão de revisão ele se encontra;
- b) Etapas previstas:** quais as etapas que vão existir no projeto, sendo que etapas são consideradas mecanismos que reúnem atividades que procuram cumprir o propósito que o próprio nome da etapa define;
- c) Atividades previstas:** são exatamente as ações que procuram cumprir passo a passo, de modo sequencial ou não, um determinado objetivo da etapa. Quando totalmente finalizadas, concluem a etapa a que pertencem;
- d) Análise das etapas:** é muito importante procurar sempre desenvolver nossa capacidade de organização e de definição de etapas e atividades, para tanto, uma reflexão final deve ser executada para contemplar onde você acertou e onde pode melhorar, para que da próxima vez que executar um projeto igual ou semelhante, tenha um resultado ainda mais satisfatório.

Fica fácil compreender que mesmo um exemplo de formulário simples como esse colabora de forma contundente em um determinado projeto, se comparado a um cenário em que você simplesmente sairia fazendo uma série de atividades, sem parar para refletir na pertinência que uma tem sobre a outra. Além disso, a chance de você deixar algo de fora seria muito grande.

Uma das grandes vantagens de se colocar no papel um plano é que você realmente parará para pensar no mesmo, esforçando-se para perceber que não pode deixar detalhes importantes de fora, que existe uma sequência de atividades que devem ser realizadas e, o mais importante, que poderá acompanhar a execução das mesmas, efetuando uma marcação de realizado e não realizado, promovendo, assim, uma visão de produtividade frente aos objetivos propostos.

Veja, a seguir, este formulário preenchido no exemplo do jantar.

PROJETO	Jantar com amigos	AUTOR	Meyer
DATA	14/04/2019	VERSÃO	1.0

Etapas previstas

ETAPA	INÍCIO	FIM	REALIZADAS	AGUARDANDO
Cardápio	15/04/19		2	1
Compras			0	3
Pré-preparação				
Preparo				
Jantar				
Pós-jantar				

Atividades previstas nas etapas

ID	ATIVIDADE	ETAPA	REQUISITO	DATA	FEITO?
1	Observar a dispensa, geladeira e congelador e observar o que se tem disponível	Cardápio	X	15/04/19	S
2	Levantar o gosto dos convidados	Cardápio	X	15/04/19	S
3	Definir o cardápio	Cardápio	1 e 2		
4	Fazer levantamento do que precisa ser comprado	Compras	3		
5	Fazer pesquisa de preço do que precisa ser comprado nos supermercados	Compras	4		
6	Comprar os itens do jantar	Compras	5		

Análise das etapas

ETAPA	ANÁLISE SOBRE A ETAPA
Cardápio	Muito importante, tinha convidado que era alérgico a leite.
Compras	Melhor comprar em diversos supermercados. Pois o preço da carne é melhor em alguns, a bebida em outros, etc.

Tabela 12 - exemplo de utilização do formulário de organização das etapas e atividades
Fonte: Do autor (2019)

6.3 ORGANIZAÇÃO DE DADOS PARA ANÁLISE

Com toda certeza você deve lembrar dos diversos documentos que viu ao longo desta unidade curricular, nas regras de negócio, no levantamento de requisitos ou, então, na UML e em seus diversos diagramas de comportamento e de estrutura. O importante é que você compreenda que deve procurar elaborar um modo operante que contribua para a organização do trabalho, de modo que possa ser possível ordenar, observar, acessar, editar e buscar os dados. E que, desse modo, você consiga ler as informações pertinentes ao seu projeto e construir o conhecimento necessário para tomar sempre as decisões mais viáveis e corretas.



Novamente, não existe uma receita de bolo pronta para elaborar um documento que o auxilie nesta importante empreitada de organizar seus projetos. Contudo, com todo o conhecimento que já adquiriu aqui, fica fácil você mesmo elaborar, mediante sua cultura profissional e ritmo de trabalho, documentos e métodos que contribuam de forma definitiva para que seu trabalho possa sempre render positivamente em busca dos objetivos determinados pelo escopo do projeto.

No entanto, a seguir você terá algumas dicas para criar o documento que vai lhe auxiliar na organização dos seus projetos.

- a) Eu consigo facilmente trabalhar utilizando esse método? Ou ele me custará tanto esforço que ficará inviável utilizar?
- b) Eu consigo manter esse método? Ou depois de utilizá-lo por algum tempo, ele fica inviável por ser caro, cansativo, inapropriado, etc.?
- c) Eu consigo replicar esse método? Ou a cada vez que vou fazer um novo projeto, tenho que o reformular drasticamente?
- d) Eu tenho sempre ele disponível? Ou necessito estar sempre conectado, possuir HD externo, etc.?
- e) Eu consigo resgatar as informações quando preciso? Ou preciso gastar muito tempo para achar o dado ou informação de que necessito?

Esses são apenas alguns exemplos de critérios para a construção de um método que garanta sempre uma visão organizada e analítica dos projetos. Procure a sua forma e, certamente, você criará métodos que terão muito sucesso.

**FIQUE
ALERTA**

Cuidado para não burocratizar muito seu modo de organização de trabalho, a ponto de comprometer a efetividade e a produtividade. Da mesma maneira que a falta de controle pode levar ao caos, muito controle pode levar à estagnação. A medida certa é dada pela sua cultura profissional e também pela aderência que as soluções adotadas terão entre si.

**SAIBA
MAIS**

O Manifesto Ágil foi um importante movimento que contribuiu, de forma disruptiva, na elaboração de novas metodologias de desenvolvimento de sistemas. Pois ele pregou que:

- a) Indivíduos e interações são mais importantes do que processos e ferramentas;
- b) Software em funcionamento é mais importante do que documentação abrangente;
- c) Colaboração com o cliente é mais importante do que negociação de contratos;
- d) Responder a mudanças é mais importante do que seguir um plano.

**RECAPITULANDO**

Neste capítulo, você teve a oportunidade de compreender o quanto é importante não apenas possuir o conhecimento de técnicas e ferramentas para modelar um sistema, mas também que é muito importante saber organizar todos os dados e as informações que permearão os processos dentro de um projeto de sistema de computação. Organização é a chave para um trabalho bem feito, replicável e apto para análise de melhoria constante.

Métodos e Técnicas de Trabalho

7



Você está chegando na reta final do seu estudo e vai agora aprender sobre dois tópicos muito importantes que, como profissional que será capaz de modelar sistemas, você necessita muito para se manter sempre produtivo e eficaz. Neste capítulo, você vai estudar a análise de informações e dados e o ciclo PDCA. Ambos são muito importantes para que você possa, a partir de todo conhecimento estudado até aqui, aplicar, de forma coerente e perspicaz, extrairindo o melhor resultado possível de um produto/serviço muito bem modelado. Sendo assim ao final deste capítulo você terá subsídios para:

- a) Compreender diferentes métodos e técnicas de trabalho;
- b) Analisar informações e dados de sistemas.

Inicie os estudos deste capítulo compreendendo a importância da análise das informações e dados para o seu dia a dia profissional.

7.1 ANÁLISE DE INFORMAÇÕES E DADOS

Quando você se depara com o termo “análise”, e isso é muito comum na área da computação, por exemplo, no tão almejado cargo de “Analista de sistemas”, deve compreender melhor o significado desta palavra, pois ela carrega consigo uma coleção de responsabilidades que serão exigidas do profissional.



Ilustração: Sergey (Molina) (20-7)

Observe alguns dos possíveis sinônimos da palavra análise:

- a) Observação;
- b) Diagnóstico;
- c) Investigação;
- d) Pesquisa;
- e) Verificação;
- f) Avaliação;
- g) Apreciação;
- h) Exploração;
- i) Inspeção;
- j) Decomposição;
- k) Crítica;
- l) Parecer.

Fica claro que o papel de um analista é extremamente complexo e importante, pois para ele não basta apenas a "informação" ou o "dado". Para ele, o que está em jogo é toda a força da palavra análise (e seus sinônimos), quando se trata de um sistema de informação e como as informações e dados contribuirão neste sistema.

No capítulo anterior, você viu a diferença entre dado, informação e conhecimento. Partindo, então, desse pressuposto, você precisa agora agregar mais uma questão, que é a análise de dados e informações.



FIQUE ALERTA

A necessidade de possuir a competência de analisar um sistema e dominar as informações e dados que nele estão inseridos é fundamental. Quando você se depara com algum problema técnico qualquer no sistema, quando quer aplicar alguma melhoria ou quando quer simplesmente alterar o sistema para que ele possa realizar uma determinada função de forma diferente, é exatamente a visão de análise do sistema e de todas as suas respectivas informações e dados que permite que se possa fazer isso de forma nítida e precisa.

Você deve lembrar que nessa área de atuação, além do conhecimento lógico e técnico, também devem ser consideradas as questões subjetivas, não é mesmo? Pois bem, imagine a seguinte situação: quando seu carro está com problema, você leva a um profissional mecânico e ele analisa o veículo. Ele consegue fazer isso porque conhece de forma muito competente diversas informações e dados do carro. Ele abre o capô do motor e conhece cada uma das peças. Ele comprehende o papel fundamental de cada peça e o modo como elas interagem entre si em busca do objetivo final.

Você vai notar também que, mesmo tendo todo esse conhecimento, ele irá primeiro escutar com atenção seu testemunho, ainda que você não entenda nada de motor, pois a partir do que ouve ele poderá extrair alguma informação pertinente ao caso que tentará solucionar. Depois de escutá-lo, ele vai começar a usar diversas técnicas, analisando com mais profundidade cada ponto do carro, procurando, assim, isolar as possibilidades e, finalmente, chegar no objetivo: o problema atual do carro.

Note que o mecânico, neste exemplo, conseguirá chegar a uma solução porque tem a competência de análise sobre a mecânica do carro. Isso acontece porque quando ele olha para as peças (dados), entende como elas estão interligadas (informação) e como elas contribuem para o funcionalmente total do motor (sistema).

E nos sistemas de informação? Como funciona ter essa visão analítica de um sistema? Simples, você deve ter toda a competência para compreender profundamente (em toda a extensão do significado da palavra análise), os componentes que formam aquele determinado sistema de informação. Por isso, um bom analista de sistema entende profundamente de modelagem de sistema, de engenharia de software, domina a UML e diversas outras técnicas capazes de modelar e manter sistemas computacionais. Só assim, quando um sistema possui alguma falha ou precisa de uma mudança brusca, ele consegue, a partir de sua análise, efetuar todas as atividades técnicas necessárias para concluir o desafio.



CASOS E RELATOS

Visão de modelagem

Na carreira de TI, um dos cargos mais desejados pela maioria do pessoal, que está trabalhando com programação e desenvolvimento de sistema, é justamente o cargo de Analista de sistemas. Esse também era o cargo almejado por Jorge, que acabou de ingressar no mercado de trabalho. No entanto, Jorge já percebeu que, para ser um bom analista, ele não deve apenas ser um grande desenvolvedor, com vasta experiência em programação, em diversas linguagens e para diversos tipos de sistemas. Ele precisa, principalmente, de "tato" e "visão" para compreender as demandas dos clientes ou do mercado e transformar isso em funcionalidades modeladas para o sistema. É justamente essa capacidade que faz a diferença entre um grande analista de sistema e um grande programador com cargo de analista. Possuir uma boa visão de modelagem é com toda certeza um grande diferencial para evoluir em uma carreira de TI.

7.2 CICLO PDCA

Ao criar produtos e serviços, é preciso ter maturidade para compreender que eles dificilmente serão estáticos, ou seja, dificilmente irão se manter da mesma forma (características, qualidades, limitações, etc.) para sempre. Pois ao longo do tempo eles devem ser mantidos, mas em relação a um controle e melhoria continuada, visando manter esses produtos e serviços competitivos e com valor agregado. Existem diversos métodos e técnicas que você pode aplicar para garantir essa evolução constante, e aqui você vai estuar uma que é muito difundida em todo meio da gestão: o Ciclo PDCA.

O PDCA é um método baseado na repetição, aplicado sucessivamente em determinados processos e focado na melhoria contínua que garante o alcance de metas estabelecidas. Ele é composto por quatro etapas, em que cada uma se responsabiliza por um momento de uma determinada interação do ciclo, e tem como objetivo entregar para a próxima etapa uma qualificação de dados, informações e atividades.

A sigla PDCA vem das palavras em inglês:

PLAN (Planejar);

DO (Fazer);

CHECK (Verificar);

ACT (Agir).


CURIOSIDADES

É muito comum, principalmente na área da Computação, que termos em inglês sejam utilizados amplamente para definir técnicas, métodos, etapas, tecnologias, estruturas, etc. Caso você ainda não tenha uma boa formação em inglês (que atualmente é algo muito pertinente), procure ao menos desenvolver um inglês técnico. Ou seja, acostume-se a utilizar os termos técnicos em inglês, evitando traduzi-los. Procure falar, por exemplo: *commit* ao invés de *submeter/enviar/colocar*.

As quatro etapas do PDCA estão interligadas em um determinado ciclo:


istockphoto (120-7)

E veja bem! Não apenas as etapas estão interligadas sequencialmente, como mostra a imagem anterior (*Plan, Do, Check e Act*), mas também cada ciclo está ligado com o anterior. Ou seja, cada ciclo concluído alimentará o próximo, sendo pertinente para o planejamento e, consequentemente, para todas as outras etapas previstas. Note, então, que existe um inter-relacionamento dos ciclos e de suas etapas, e que a palavra continuidade é respeitada profundamente neste método.


SAIBA MAIS

E quais as origens do ciclo PDCA? Bem, ele surgiu em meados de 1950 por mérito do professor William Edwards Deming (1900-1993), que é reconhecido por muitos como o pai do controle de qualidade nos processos produtivos. Existe também uma linha que atribui a criação do PDCA ao engenheiro Walter Shewhart (1891-1967), que era contemporâneo de Deming. Mas independentemente da autoria, o importante é compreender que ambos pretendiam, a partir de uma técnica, acelerar e aperfeiçoar os processos das empresas por meio da identificação de problemas, suas causas e propondo soluções. E resultado dessa proposta foi o PDCA.

Vamos agora procurar entender o que se faz em cada uma das etapas previstas em cada ciclo.

7.2.1 PLANEJAR

Planejar (*Plan*) é a etapa inicial de cada ciclo e procura estabelecer quais os objetivos e processos que serão necessários para que se possa entregar os resultados esperados, definindo, de forma qualificada, os objetivos e metas do ciclo.



FIQUE ALERTA

É importante entender que um planejamento mal feito (ou a ausência de um planejamento) muito provavelmente conduzirá a resultados ruins, contrários ao esperado ou até mesmo não gerará resultados. Por isso, é preciso sempre executar essa etapa de forma cautelosa, colaborativa, com pensamento crítico e analítico. O planejamento deve sempre compreender o que ele deve cumprir naquele ciclo e como este colabora com o objetivo contínuo.

Você já imaginou possuir o objetivo de fazer uma grande viagem, mas não efetuar nenhum tipo de planejamento? Não saber onde você irá se hospedar, em quais restaurantes irá se alimentar, quais os pontos turísticos que deseja visitar, qual a quantidade de roupa e dinheiro que deve levar. Enfim, não se planejar pode levar uma situação que era para ser boa a se transformar em uma experiência extremamente frustrante.

7.2.2 FAZER

Fazer (*Do*) é a etapa responsável por começar a colocar a “mão na massa”, ou seja, deve-se executar tudo aquilo que foi previsto na etapa anterior. Porém, é preciso ter claro que antes de iniciar essa fase é muito importante que todos os envolvidos no processo estejam comprometidos, sabedores daquilo que foi planejado e quais os objetivos e metas para que tudo saia conforme o planejamento.



Da mesma forma que é muito importante ter um excelente planejamento bem detalhado que expresse de forma clara quais os objetivos e metas devem ser alcançados, o ciclo deve executar, realizar, cumprir e fazer aquilo que foi planejado. Em um mundo altamente competitivo e imediatista, planejar e não fazer é algo inaceitável. No entanto, é comum ver muitas pessoas, que planejam pouco e sonham muito, não colocarem em prática.

Depois de planejar e fazer, chegou a hora de verificar. Vamos à próxima etapa.

7.2.3 CHECAR

Checar ou verificar (*Check*) é a fase que será colocada em prática depois de tudo ter sido meticulosamente planejado e de todos os envolvidos estarem cientes do porquê vai ser feito e de como vai ser executado. É muito importante que toda as atividades sejam monitoradas, verificando se elas estão condizentes com o esperado e à luz do que foi estabelecido nas metas e objetivos.

Deve-se criticar aquilo que foi executado na etapa anterior, para que este ciclo possa ter a maior aderência possível com o seu compromisso existencial. Para isso, a partir dos dados coletados na etapa anterior, o profissional deve analisar e verificar se o que foi proposto se realizou.

As verificações podem ser quantitativas ou qualitativas. Elas podem ser feitas pela comparação da quantidade de determinadas atividades que deveriam ser entregues no final de um determinado ciclo do PDCA ou pela qualidade de um determinado artefato gerado a partir das atividades realizadas. Lembre-se que a verificação é sempre uma forma da busca contínua da qualidade, pois, muito mais do que apenas realizar, o que se procura sempre é realizar o certo/desejado.



**FIQUE
ALERTA**

Não é possível garantir uma boa verificação, se não ficar muito claro quais são os objetivos que se deve alcançar naquele determinado ciclo de PDCA. Se, por exemplo, o objetivo final é percorrer 10km em um projeto, mas em um determinado ciclo o objetivo é percorrer 1km, o fator de crítica para desempenho de meta atingida deverá ser 1km, e não 10km. E o que poderíamos verificar nesse caso? Poderia ser verificada: qual a distância percorrida? Andamos na direção correta? Os recursos que utilizamos para andar (tênis, roupa adequada, etc.) estão bem ajustados? A hidratação foi realizada durante a caminhada? Observe que determinar o que será verificado é muito importante, pois gerará dados e informações que muito possivelmente alimentarão ações na próxima etapa (Agir) com o intuito corretivo, de melhoria de performance ou para análise posterior.

7.2.4 AGIR

E você sabe quando deve ser a hora de agir? Devemos agir (*Act*) uma vez que o ciclo anterior tenha revelado inconsistências entre o que foi planejado e o que foi entregue. É necessário tomar todas as ações corretivas sobre aquilo que, de forma significativa, demonstrou-se diferente entre os resultados reais e os esperados pelo planejamento. Dessa forma, é importante analisar as diferenças, procurando determinar quais foram as causas do desvio. Além disso, é preciso definir as ações necessárias para que a rota de desenvolvimento volte ao eixo do planejamento, promovendo o alcance dos objetivos e metas.



RECAPITULANDO

Você chegou ao final do seu estudo, verificando a importância de analisar as informações e dados, e obtendo, assim, a capacidade de ter uma visão analítica e qualitativa de um sistema ou processo. Além disso, você teve a oportunidade de ver como uma técnica de PDCA pode colaborar, e muito, em um projeto de sistemas, agregando mais ainda em um processo de modelagem de sistemas. Sucesso em sua caminhada!

REFERÊNCIAS

- WAZLAWICK, Raul Sidnei. **Análise e design orientados a objetos para sistemas de informação:** modelagem com UML, OCL e IFML. 3. ed. Rio de Janeiro: Elsevier, 2015. 462 p. ISBN 9788535279849.
- DENNIS, Alan; WIXOM, Barbara Haley. **Análise e projeto de sistemas.** 5. ed. Rio de Janeiro: LTC, 2014. 536 p. ISBN 9788521625094.
- AZAMBUJA, Telmo Travassos de. **Documentação de sistemas da qualidade:** um guia prático a gestão das organizações. Rio de Janeiro: Campus, 1996. 283 p. ISBN 8535200770.
- SELEME, Robson; STADLER, Humberto. **Controle da qualidade:** as ferramentas essenciais. 2. ed. Curitiba: IBPEX, 2010. 180 p. ISBN 9788578387181.
- DAVIS, Christopher W. H. **Agile metrics in action:** how to measure and improve team performance. New York: Manning, 2015. 246 p. ISBN 9781617292484.
- MAXIMIANO, Antonio Cesar Amaru. **Administração de projetos:** como transformar ideias em resultados. 4. ed. São Paulo: Atlas, c2010. 396 p. ISBN 9788522460960.

MINICURRÍCULO DO AUTOR

RAFAEL SOUZA MEYER

Rafael Souza Meyer é graduado em Sistemas de Informação pela Universidade do Sul de Santa Catarina (Unisul), possui Curso Sequencial na graduação em Desenvolvimento Java (Unisul), é especialista em Gestão de Projetos em Software Livre (Unisul) e é mestrando no Programa de Pós-Graduação em Tecnologias da Informação e Comunicação na Universidade Federal de Santa Catarina (UFSC). Possui experiência com o desenvolvimento de sistemas Desktop, Web e *Mobile*, trabalhou também como analista de sistemas e gerente de projetos. Como professor leciona as disciplinas de: Engenharia de Software, Desenvolvimento Web Avançado, Desenvolvimento Java, Programação Orientada a Objetos, Mercado de TI, Princípios da Qualidade do Software, Testes, Modelagem de Sistemas e Lógica de Programação. Atualmente é professor do Senai-SC.

ÍNDICE

D

Diagrama comportamental, 61, 63

Diagrama estrutural, 48, 53

E

Engenharia de software, 89, 99

M

Modelagem de sistemas, 1, 3, 7, 9, 12, 24, 27, 28, 29, 37, 38, 39, 40, 41, 60, 61, 63, 67, 68, 70, 72, 73, 75, 77, 89, 94, 99

P

PDCA, 7, 87, 90, 91, 93, 94

Produtividade, 41, 81, 84

R

Regras de negócio, 5, 11, 12, 13, 14, 18, 19, 25, 27, 28, 29, 34, 60, 61, 63, 67, 72, 75, 82

Requisitos funcionais, 11, 14, 15, 16, 17, 21, 22, 23

Requisitos não funcionais, 11, 17, 22, 23

U

UML, 5, 7, 9, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24, 25, 27, 28, 29, 30, 31, 32, 33, 34, 35, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 67, 68, 69, 70, 71, 72, 73, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 87, 89, 90, 91, 92, 93, 94, 97

SENAI – DEPARTAMENTO NACIONAL

UNIDADE DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA – UNIEP

Felipe Esteves Morgado

Gerente-Executivo

Luiz Eduardo Leão

Gerente de Tecnologias Educacionais

Catarina Gama Catão

Coordenação Geral do Desenvolvimento dos Livros Didáticos

SENAI – DEPARTAMENTO REGIONAL DE SANTA CATARINA

Fabrizio Machado Pereira

Diretor do SENAI/SC

Claudemir José Bonatto

Diretor de Educação do SENAI/SC

Daniel Thiesen Horongoso

Gerente Executivo da Regional Sudeste SENAI/SC

Ricardo Maximo Anzolin

Gerente de Operações da Regional Sudeste SENAI/SC

Priscila Carneiro Gallasse Cesconetto

Supervisora de Consultoria em Educação

Rafael Souza Meyer

Elaboração

Felipe Demarchi

Revisão técnica

Daiana Silva

Design educacional

Carlos André Marques de Andrade

Davi Leon Dias

Ilustrações e tratamento de imagens

IStock
SENAI/SC
Banco de imagens

Tatiana Daou Segalin

Diagramação

Tatiana Daou Segalin

Revisão e Fechamento de Arquivos

Luciana Effting Takiuchi

CRB – 14/937

Ficha Catalográfica

i-Comunicação

Projeto Gráfico

Editorar Multimídia Ltda.

Revisão Ortográfica e Gramatical

Editorar Multimídia Ltda.

Normalização



*Iniciativa da CNI - Confederação
Nacional da Indústria*



