TEXTO DE APOIO

AULA 1

Desenvolvimento de Sistemas I

Professora Cláudia Rossi

EaD | Universidade Presbiteriana Mackenzie

Sumário

MODELAGEM DE SISTEMAS E O PARADIGMA ORIENTADO A OBJETOS

INTRODUÇÃO

Você já estudou, nos componentes curriculares anteriores, que os sistemas computacionais estão presentes cada vez mais nas atividades tanto das pessoas como das organizações, não somente em centro de dados (datas centers), em servidores de aplicações corporativas ou em estações de trabalho, mas também em veículos, televisores, geladeiras, telefones celulares e outros.

Você pode notar que um sistema é composto de diferentes elementos que lhe dão "vida" e que determinam sua complexidade. Geralmente, esses elementos são: equipamentos (hardware), processos, procedimentos, regras/políticas, software, firmware, pessoas, entre outros.

Na figura, podemos observar um engenheiro de chão de fábrica controlando uma linha de produção de uma indústria.

Assim, nota-se que um sistema de informação é uma combinação de pessoas, dados, processos, interfaces entre sistemas, redes de comunicação e tecnologia que interagem com o objetivo de fornecer suporte e melhorar o processo de negócio de uma organização com relação às informações que nela fluem.

Podemos definir um sistema de informação como um conjunto de componentes interrelacionados trabalhando juntos para coletar, recuperar, processar, armazenar e distribuir informações com a finalidade de facilitar o planejamento, o controle, a coordenação, a análise e o processo decisório em organizações.

Uma característica dos sistemas de informação é a complexidade. Dessa forma, o sistema pode ser menos ou mais complexo e pode envolver menos ou mais pessoas em seu desenvolvimento e levar mais ou menos tempo para sua construção (desenvolvimento).

Além disso, a formação necessária para um profissional atuar na indústria de software vai além da formação tecnológica; deve fazer parte dessa formação disciplinas de negócio, gestão, estratégia e relações humanas e empresariais, uma vez que este profissional deverá atuar em um mercado no qual se busca obter, com a TI, vantagens competitivas por meio de uma gestão mais eficiente de suas operações e de descobrir meios de colaborar com parceiros comerciais e industriais para fornecer produtos e serviços ao mercado de forma lucrativa.

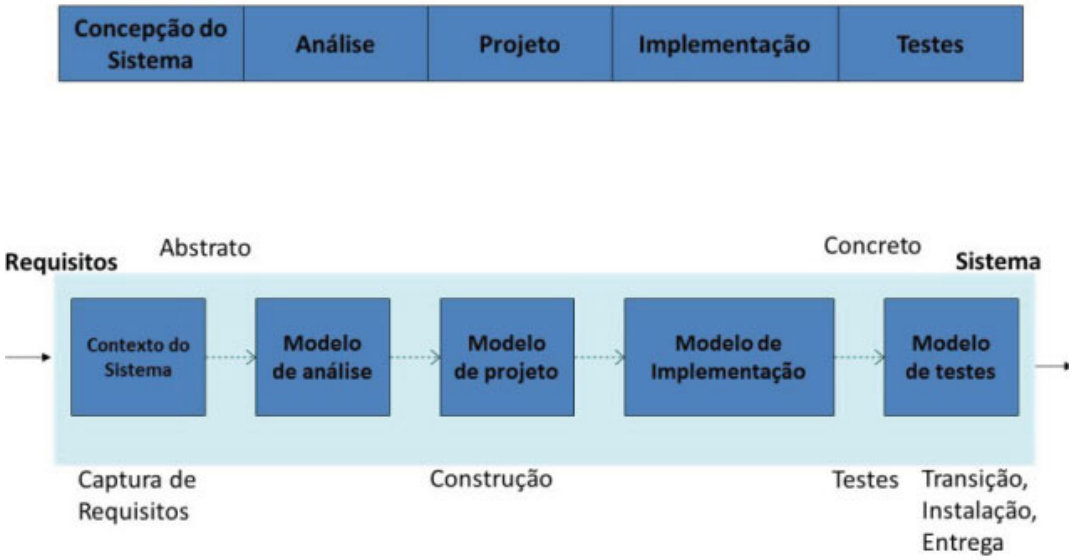
Assim, o que será estudado nesse componente curricular pretende prepará-lo para lidar com esses desafios do contexto corporativo, que são:

- Lidar com a complexidade, traduzir necessidades de negócio e viabilizar estratégias corporativas.

- Desenvolver novas formas de produtos e serviços e, até mesmo, entretenimento às pessoas por meio de sistemas.

Como futuros profissionais da área de desenvolvimento de sistemas, vocês devem utilizar métodos, técnicas e ferramentas para lidar com desafios do mundo real e transformar em soluções de software. Para isso, neste componente curricular, você aprenderá métodos, técnicas e ferramentas para construir abstrações por meio de modelos a fim de permitir a comunicação e o entendimento uniforme entre os diferentes envolvidos durante as fases do processo de desenvolvimento de sistemas, que são: concepção, análise, projeto, implementação e testes, como mostra a figura a seguir.

Figura: 1 Fases do processo de desenvolvimento de sistemas



Fonte: Elaborada pela autora.

A **Modelagem de Sistemas de Software** permitirá, por meio de notação gráficas e textuais, traduzir problemas do mundo real em modelos que representaram as partes essenciais de um sistema, considerando as várias perspectivas diferentes e complementares.

É importante você perceber que nossa necessidade de construir modelos do sistema tem como objetivo:

- Gerenciar a complexidade envolvida em um cenário de negócio no qual temos de desenvolver uma solução de software.
- Possibilitar a comunicação e o entendimento entre as pessoas envolvidas no processo de desenvolvimento do software.
- Conseguir estimar tempo e custo envolvido no desenvolvimento do software.
- Possibilitar a modelagem do comportamento das partes do sistema.

Afinal,

“Uma figura vale por mil palavras”

Você concorda?

ABSTRAÇÃO

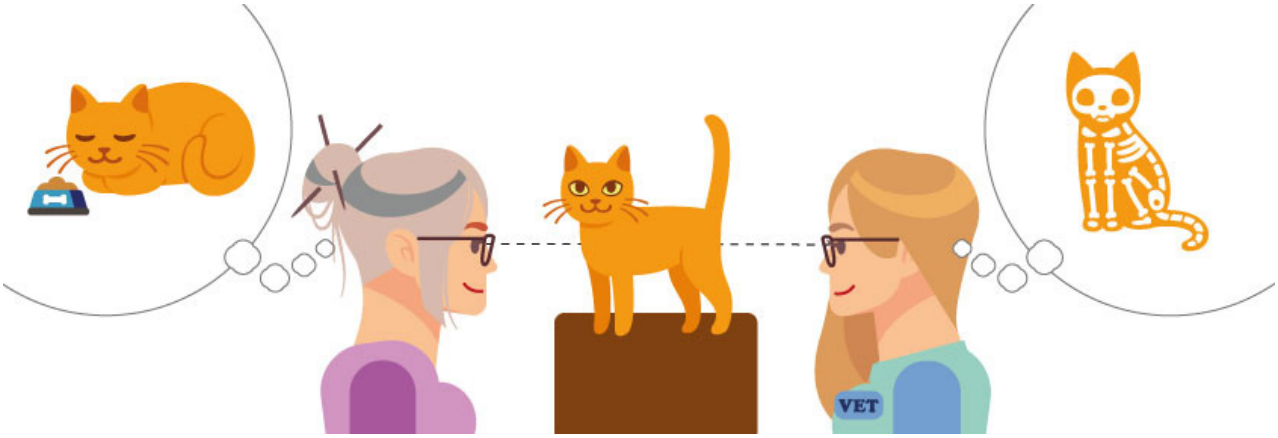
Abstração é o exame seletivo de certos aspectos de um problema. Para Booch (2007), a abstração é o modo fundamental que nós, humanos, lidamos com a complexidade. Portanto, você pode perceber que lidar com a complexidade é um dos objetivos de se construir modelos (abstrações de um sistema).

Observe essas duas definições e os comentários relevantes sobre a abstração:

- Dahl, Dijkstra e Hoare (1972) argumentam que a abstração surge a partir de um reconhecimento de semelhanças entre certos objetos, situações ou processos no mundo real, e da decisão de concentrar-se sobre essas semelhanças e ignorar, ao mesmo tempo, as diferenças;

- Booch (2007) argumenta que uma abstração denota as características essenciais de um objeto que as distinguem de todos os outros tipos de objetos e, assim, proporciona nitidez, definindo os limites conceituais em relação à perspectiva do espectador.

Figura 2- A abstração depende do observador



Abstração foca nas características essenciais de um objeto enquanto ignora detalhes que não são relevantes do ponto de vista do observador.

A abstração é um dos pilares do paradigma orientado a objetos.

O que é Orientação a Objetos?

Organizar o software como uma coleção de objetos distintos, que incorporam estrutura de dados e comportamento.

Um pouco de história, segundo Bezerra (2015, seção 1.2):

> Alay Curtis Kay, um dos pais da Orientação a Objetos, formulou a analogia biológica: Sistema de software deveria funcionar como um ser vivo, ou seja, cada célula interagiria com outras células através do envio de mensagens para realizar um objetivo em comum. Além disso, cada célula se comportaria como uma unidade autônoma.

Princípios da Orientação a Objetos – Alan Kay

1. Qualquer coisa é um objeto.
2. Objetos realizam tarefas por meio da requisição de serviços a outros objetos.
3. Cada objeto pertence a uma determinada classe. Uma classe agrupa objetos similares.
4. A classe é um repositório para comportamento associado ao objeto.
5. Classes são organizadas em hierarquias.

MAS O QUE ISSO SIGNIFICA? VAMOS ENTENDER MELHOR?

Vamos observar as bicicletas da figura a seguir:

Figura 3- Exemplo de Princípios da Orientação a Objetos

Fonte: Elaborada pela autora.

Você pode perceber que todas as bicicletas são diferentes entre si, mas possuem características em comum: todas têm uma cor (vermelha, preta, rosa, verde), são de um modelo (passeio, competição, mais esportivas), têm uma marca (Caloi, Monark, Track Bike), e assim por diante.

Você também pode notar que as bicicletas servem para nos deslocar de um lugar para outro; para isso, devemos averiguar ações que promovem movimentos: elas se deslocam, aceleram, freiam, mudam de marcha, mediante a comandos de pedaladas, comandos de freios e mudanças de marcha.

Essas bicicletas representam objetos do mundo real que pertencem à mesma classe, todas elas são Bicicletas que possuem características (Cor, Modelo, Marca, Número de Marchas, Tamanho de Quadro, algumas têm motor elétrico, outras não) e fazem as mesmas coisas (aceleram, freiam e mudam de marcha).

Ou seja, todas são objetos com a mesma estrutura de dados (características) e comportamento (ações que fazem), ou seja, você pode perceber que elas podem ser agrupadas em uma classe de Objetos – a Classe Bicicleta.

A **classe** é uma abstração que descreve propriedades importantes em um determinado contexto e ignora as demais que não são relevantes naquele contexto.

Cada **objeto** é considerado uma instância de uma classe, ou seja, uma ocorrência daquela classe no mundo real.

A **identidade de um objeto** significa que os dados são quantizados em entidades distintas e distinguíveis chamadas objetos, por exemplo: as bicicletas da Bruna e da Camila, cada uma delas tem uma identidade única e inerente, ou seja, dois objetos distintos, mesmo que tenham todos os seus valores de propriedades idênticos, são objetos distintos.

BICICLETA DA BRUNA

BICICLETA DA CAMILA

É importante você notar que, no mundo real, o objeto simplesmente existe, mas, em linguagem de programação, cada objeto possui uma referência única pela qual ele pode ser acessado. Essas referências dos objetos são uniformes e independentes do conteúdo dos objetos, permitindo a criação de coleções mistas de objetos.

ENCAPSULAMENTO

Outro conceito muito importante da orientação a objetos que devemos conhecer é o conceito de Encapsulamento, o qual trata do ocultamento de informações, ou seja, se houver separação dos aspectos externos de um objeto, que são acessíveis a outros objetos, dos detalhes internos da implementação, que são escondidos em outros objetos.

Para Booch (2007), o **encapsulamento** é o processo de compartimentar os elementos de uma abstração que constitui sua estrutura e seu comportamento; encapsulamento serve para separar a interface contratual de uma abstração e sua implementação.

Dessa forma, podemos concluir que abstração e encapsulamento são conceitos complementares? Reflitam sobre os pontos:

- Abstração foca sobre o comportamento observável de um objeto.

- Encapsulamento foca sobre a implementação que dá origem a este comportamento. O encapsulamento fornece barreiras explícitas entre as diferentes abstrações e, assim, trata de uma clara separação de preocupações (interesses).

Esse conceito pode ser observado na figura a seguir.

Figura 4- Encapsulamento esconde detalhes da implementação de um objeto

"Encapsulamento esconde detalhes da implementação de um objeto." (BOOCH, 2007).

MODULARIZAÇÃO

A Modularização consiste em dividir um sistema em subsistemas, módulos, componentes, serviços, microsserviços, classes, que denominaremos de módulos (para simplificar). Esses módulos podem ser compilados separadamente, mas têm conexões com os outros módulos que compõem o sistema.

A modularidade é uma propriedade de um sistema que foi decomposto em um conjunto coeso e fracamente acoplado de módulos.

Figura 5- A modularidade empacota as abstrações em unidades discretas

“A modularidade empacota as abstrações em unidades discretas” (BOOCH; 2007).

REFERÊNCIAS

BEZERRA, E. *Princípios de análise e projeto de sistemas com UML*. 3. ed. Rio de Janeiro: Elsevier-Campus, 2015. Disponível em:

BOOCH, G. *Object-oriented analysis and design with applications*. 3. ed. Boston: Addison-Wesley, 2007.

DAHL, O.; DIJKSTRA, E.; HOARE, C. A. R. *Structured Programming*. London, England: Academic Press, 1972.