



SISTEMAS DA INFORMAÇÃO

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 6

CQA/UNIP – Comissão de Qualificação e Avaliação da UNIP

SISTEMAS DA INFORMAÇÃO

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 6

Christiane Mazur Doi

Doutora em Engenharia Metalúrgica e de Materiais, Mestra em Ciências - Tecnologia Nuclear, Especialista em Cálculo e Matemática Aplicada, Especialista em Língua Portuguesa e Literatura, Especialista em Recursos Gramaticais para Revisão Textual, Engenheira Química e Licenciada em Matemática, com Aperfeiçoamento em Estatística e MBA em Engenharia Econômica. Professora titular da Universidade Paulista.

José Carlos Morilla

Doutor e Mestre em Engenharia de Materiais, Mestre em Engenharia de Produção, Especialista em Engenharia Metalúrgica, Especialista em Física e Engenheiro Mecânico, com MBA em Gestão de Empresas. Professor adjunto da Universidade Paulista.

Tiago Guglielmeti Correale

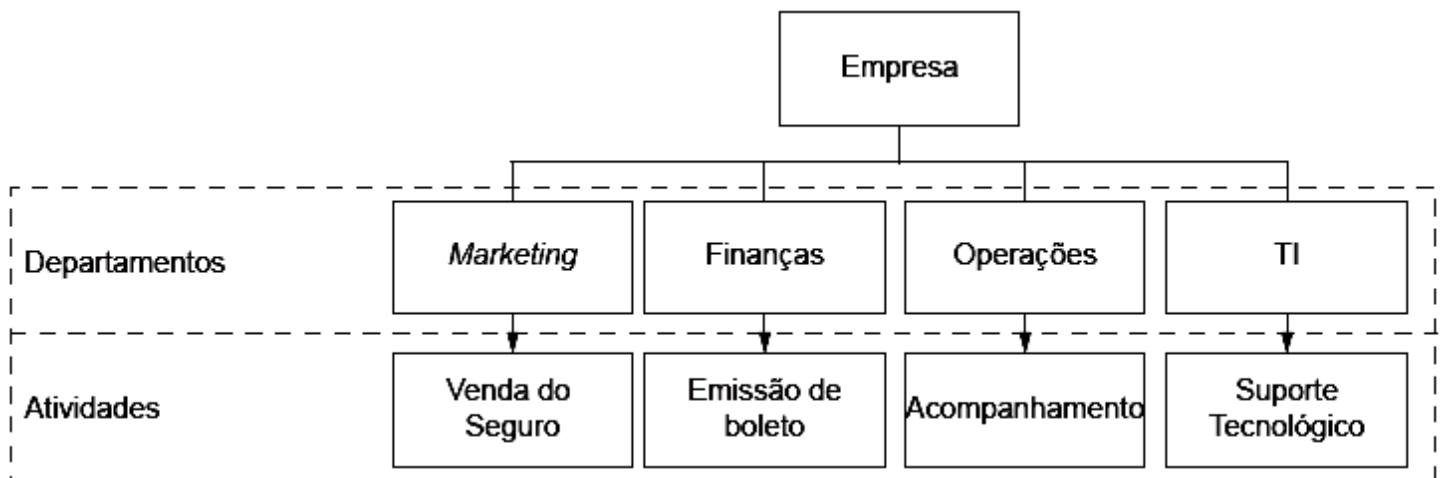
Doutor em Engenharia Elétrica, Mestre em Engenharia Elétrica e Engenheiro Elétrico (ênfase em Telecomunicações). Professor titular da Universidade Paulista.

Material instrucional específico, cujo conteúdo integral ou parcial não pode ser reproduzido ou utilizado sem autorização expressa, por escrito, da CQA/UNIP – Comissão de Qualificação e Avaliação da UNIP – UNIVERSIDADE PAULISTA.

Questão 1

Questão 1.¹

Uma empresa que comercializa seguros conta atualmente com diferentes sistemas de informação, perfazendo cada um deles uma atividade pelos departamentos de *marketing*, finanças, operações e tecnologia da informação (TI), conforme ilustra a figura a seguir. Considerando a importância estratégica de uma arquitetura empresarial de sistemas de informação funcionalmente integrada, a empresa, após estudo de viabilidade, identificou que haveria redução significativa de custos caso as atividades fossem gerenciadas como um processo integrado.



Ao estruturar uma arquitetura de negócios orientada à integração do processo, que forma funcional a empresa deve adotar?

- A. Orientação vertical.
- B. Orientação horizontal.
- C. Orientação a processos de TI.
- D. Orientação a processos departamentais.
- E. Orientação a processos colaborativos.

¹Questão 18 - Enade 2014.

1. Introdução teórica

Arquitetura de negócios

O conceito de negócio é uma expressão condensada dos valores intrínsecos dos clientes da empresa. Esses valores devem ser mapeados e tomados como base para a construção e a evolução do próprio negócio.

Dessa forma, o projeto da arquitetura do negócio deve começar pela clara concepção e pela real compreensão do negócio: seus recursos humanos e materiais, suas atividades, suas capacidades etc.

A arquitetura de um negócio é a base para a descrição e o entendimento de um negócio, pois ela:

- lista e apresenta as partes existentes e as partes requeridas;
- mostra como as partes são estruturadas e como interagem;
- indica como as atividades executadas pela empresa devem evoluir.

Nesse sentido, o conceito do negócio é decomposto em blocos e a arquitetura do negócio atua como o sistema que opera esse processo. A arquitetura do negócio pode ser encarada como a versão operacional do conceito do negócio, isto é, a arquitetura do negócio é o conceito posto em prática.

Assim, uma arquitetura de negócio bem estruturada representa uma visão das pessoas que a operam. Uma arquitetura é útil somente se pode ser entendida por seus usuários.

Uma boa arquitetura de negócio captura a realidade desse negócio da forma mais confiável e correta possível e define o que é realista e viável para implementação, que possibilite atingir os objetivos do negócio. Além disso, ela deve focar em seus processos e estruturas do negócio.

O foco no processo é importante, pois é a integração dos processos que permite à empresa atingir suas metas e seus objetivos. Para que o enfoque seja o processo, é preciso conhecer como as tarefas são executadas e avaliadas, a fim de encontrar meios de facilitação e de melhoria.

No enfoque por processos, o fluxo de trabalho é mais horizontal do que vertical, pois ele cruza as fronteiras funcionais.

2. Análise das alternativas

A - Alternativa incorreta.

JUSTIFICATIVA. Na orientação vertical, não há integração entre as atividades de mesmo nível.

B - Alternativa incorreta.

JUSTIFICATIVA. Na orientação horizontal, não há integração entre as atividades de níveis diferentes.

C - Alternativa incorreta.

JUSTIFICATIVA. Os processos de TI são integradores, ou seja, são meios com os quais os diversos processos da empresa são integrados.

D - Alternativa incorreta.

JUSTIFICATIVA. Na orientação por processos departamentais, podem ocorrer atividades semelhantes em departamentos diferentes e com sistemas diferentes.

E - Alternativa correta.

JUSTIFICATIVA. A orientação a processos é a melhor forma de se ter uma arquitetura de um negócio. Esse tipo de arquitetura permite integração entre as partes com maior realidade e eficácia.

3. Indicações bibliográficas

- BALTZAN, P. *Tecnologia orientada para gestão*. Porto Alegre: AMGH, 2016.
- PÁDUA, C. I. P. da S. *Modelagem da arquitetura de negócio*. Disponível em <<http://homepages.dcc.ufmg.br/~clarindo/arquivos/disciplinas/uml-mpn/material/transparencias/6-arquitetura-negocio.pdf>>. Acesso em 11 jul. 2025.
- TAKEUCHI, H.; NONAKA, I. *Gestão do conhecimento*. Porto Alegre: Bookman, 2008

Questão 2

Questão 2.²

Uma empresa mantém um serviço de *software* que atende a centenas de clientes pelo mundo. O serviço é bastante complexo e requer um esforço contínuo de manutenção. Para melhor gerenciar a manutenção do sistema, a empresa decidiu implantar um programa de medição, em que várias métricas foram agregadas, cada uma delas associada a determinada questão que se desejava responder. A tabela a seguir apresenta as questões e suas métricas associadas.

Questão	Métrica associada
Quantos problemas afetam os clientes?	Número de solicitações de mudança
Quanto tempo é necessário para se corrigir um problema urgente?	Tempo entre levantamento do problema e aceitação da mudança.
Quanto custa uma entrega de manutenção?	Custo por entrega.
Como são alocados os custos?	Custo por atividade.
Que tipos de mudanças são feitas?	Número de mudanças por tipo.
Quanto esforço é despendido por tipo de mudança	Pessoas-dia gastos, agregados por tipo
Quão difícil é a entrega?	Número de recursos computacionais a serem utilizados

O objetivo atual da empresa é reduzir o prazo de suas entregas, sem comprometer o que foi acordado com o cliente. A métrica que permite avaliar se esse objetivo está sendo atingido é o:

- A. custo por entrega.
- B. número de mudanças por tipo.
- C. número de solicitações de mudanças
- D. número de recursos computacionais a serem utilizados.
- E. tempo entre levantamento do problema e aceitação da mudança.

²Questão 24 - Enade 2014.

1. Introdução teórica

Avaliação nas empresas

Qualquer que seja o processo desenvolvido por uma organização, é fundamental que se entenda a relevância de haver indicadores de qualidade apropriados. Saber quais são os indicadores importantes e a maneira correta de como determiná-los é de vital importância para o desempenho da empresa.

A mensuração das atividades faz parte do cotidiano de uma empresa. O resultado dessa atividade permite o monitoramento e o planejamento dos rumos de uma organização.

Há três motivos para a mensuração, conforme segue.

- Determinar a história e, por consequência, a evolução de uma atividade ou problema.
- Tomar os valores encontrados como referenciais para as tomadas de decisão.
- Controlar as atividades.

Para o estabelecimento de um sistema de monitoramento e avaliação, é necessário constituir procedimentos de coleta, análise e divulgação de dados. Ou seja, devem ser respondidas as perguntas a seguir.

- Quais dados devem ser coletados?
- Como deve ser feita a coleta de dados?
- Qual deve ser a análise efetuada?
- Para quem os resultados da análise de dados devem ser divulgados?

2. Análise das alternativas

A - Alternativa incorreta.

JUSTIFICATIVA. O custo por entrega está associado à seguinte questão: quanto custa uma entrega de manutenção? A resposta a essa questão fornece as características de custo, mas não fornece as características relacionadas ao tempo.

B - Alternativa incorreta.

JUSTIFICATIVA. O número de mudanças por tipo está associado à seguinte questão: que tipos de mudanças são feitas? A resposta a essa questão fornece as características de número, mas não fornece as características relacionadas ao tempo.

C - Alternativa incorreta.

JUSTIFICATIVA. O número de solicitações de mudanças está associado à seguinte questão: quantos problemas afetam os clientes? A resposta a essa questão fornece as características de quantidade, mas não fornece as características relacionadas ao tempo.

D - Alternativa incorreta.

JUSTIFICATIVA. O número de recursos computacionais a serem utilizados está associado à seguinte questão: quão difícil é a entrega? A resposta a essa questão fornece as características de qualidade, mas não fornece as características relacionadas ao tempo.

E - Alternativa correta.

JUSTIFICATIVA. O questionamento relativo ao tempo necessário para se corrigir um problema urgente é a única pergunta que trata das características temporais.

3. Indicações bibliográficas

- ALVES FILHO, B. de F. *Processos organizacionais*. São Paulo: Atlas, 2011.
- INTERNATIONAL INSTITUTE OF BUSINESS ANALYSIS – IIBA. *Um guia para o corpo de conhecimento de análise de negócios. (Guia BABOK). Versão 2.0*. São Paulo: IIBA, 2016.

Questão 3

Questão 3.³

No desenvolvimento de um certo produto de *software*, foi determinado que um dos requisitos do produto seria a exibição de informações em planilhas eletrônicas. Contudo, durante uma reunião de revisão, a equipe do projeto verificou que esse formato não facilitava a interpretação das informações e sugeriu que fosse construído um gráfico que permitiria a identificação mais rápida de informações importantes. Diante desse fato, o gerente do projeto deve:

- A. Autorizar a equipe do projeto a implementar o gráfico e registrar a mudança.
- B. Solicitar à equipe do projeto que ignore a sugestão, uma vez que ela está fora do escopo definido.
- C. Analisar o impacto da modificação e submetê-la à aprovação do cliente, independentemente da relevância do impacto.
- D. Analisar o impacto da modificação e, somente se o impacto no projeto for grande, submetê-la à aprovação do cliente.
- E. Autorizar a equipe do projeto a implementar o gráfico e desprezar o registro da mudança, atendendo à recomendação de que sejam rastreadas apenas mudanças cujo impacto seja significativo no custo ou nos prazos do projeto.

1. Introdução teórica

Processos decisórios em projetos

A maioria das decisões em um projeto não exige uma reunião formal, pois elas podem ser tomadas em tempo real, como parte das interações que ocorrem no cotidiano das empresas. Nessas situações, os gerentes de projeto desempenham papel fundamental na condução do projeto, pois eles são os responsáveis por consultar os membros da equipe, solicitar ideias, verificar eventuais mudanças, sejam elas sugeridas ou demandadas, transmitir as mudanças aos interessados e obter a aprovação (ou não) do(s) cliente(s), independentemente da relevância para o projeto.

É importante observar que a participação coletiva reduz resistências e garante apoio às decisões, sejam elas quais forem.

³Questão 25 - Enade 2014.

O gerente de um projeto deve estar consciente de que nem todo detalhe estabelecido no plano inicial será encontrado no resultado final do projeto. As mudanças nos projetos normalmente são inevitáveis e, quando ocorrem, cabe ao gerente administrá-las.

Na maioria das situações, o gerenciamento das mudanças tem as funções elencadas a seguir.

- Identificar a mudança proposta.
- Verificar os efeitos da mudança proposta.
- Negociar e resolver os conflitos provocados pela mudança proposta.
- Obter (ou não) todas as aprovações para a mudança proposta.
- Comunicar a mudança proposta às partes afetadas.
- Fazer os ajustes no cronograma e no orçamento relativos à mudança proposta.
- Acompanhar a implementação da mudança proposta.

2. Análise das alternativas.

A - Alternativa incorreta.

JUSTIFICATIVA. Nessa alternativa, o cliente não é consultado. Uma das funções de um gerente de projeto é garantir que exista a aprovação dos interessados antes da implementação de uma mudança.

B - Alternativa incorreta.

JUSTIFICATIVA. Ignorar uma eventual melhoria no projeto é impedir o seu desenvolvimento e o da equipe que a propõe. Isso irá coibir a participação coletiva.

C - Alternativa correta.

JUSTIFICATIVA. A análise do impacto da modificação proposta e a sua submissão ao cliente mostram que o gerente trabalha em equipe e que não toma nenhuma decisão que altere o projeto sem que o cliente seja consultado. Isso transmite ao cliente uma sensação de segurança quanto à condução do projeto e ao resultado esperado.

D - Alternativa incorreta.

JUSTIFICATIVA. A submissão da mudança proposta ao cliente independe de sua relevância.

E - Alternativa incorreta.

JUSTIFICATIVA. Toda mudança, antes de ser implementada, deve ter a aprovação do cliente.

3. Indicações bibliográficas

- HELDMAN, K. *Gerência de projetos - guia para o exame oficial do PMI*. Rio de Janeiro: Campus, 2015.
- LARSON, E. W.; GRAY, C. F. *Gerenciamento de projetos – o processo gerencial*. Porto Alegre: GMBH, 2016.

Questão 4

Questão 4.⁴

Uma empresa de desenvolvimento de *software* está sendo avaliada no nível E de MPS.BR. O grupo de processos fez um levantamento para avaliar as práticas relacionadas à gerência de recursos humanos e obteve as evidências a seguir.

- **Evidência 1.** A empresa possui um processo de avaliação de desempenho e promoção de indivíduos bem estabelecido, em plano de cargos e carreira.
- **Evidência 2.** A empresa contrata externamente instrutores capacitados para realizar treinamentos internos a seus funcionários sempre que verifica a necessidade de ampliar a capacitação da equipe. Esses treinamentos, registrados como parte do portfólio de cada funcionário, são avaliados pelos funcionários em relação à sua qualidade e aderência às necessidades de capacitação da equipe. Cada funcionário também é avaliado ao final de cada treinamento.

De acordo com o Modelo de Referência do MPS.BR, alguns dos resultados esperados para o processo de Gerência de Recursos Humanos são os seguintes.

- **GRH 6.** *Os treinamentos identificados com sendo responsabilidade da organização são conduzidos e registrados;*
- **GRH 7.** *A efetividade do treinamento é avaliada;*
- **GRH 8.** *Critérios objetivos para avaliação do desempenho de grupos e indivíduos são definidos e monitorados para prover informações sobre o desempenho e melhorá-lo.*

Associação para promoção e Excelência do Software Brasileiro. *SOFTEx. MPS.BR – Guia geral MPS de Software.* 2012. Disponível em <<http://www.softex.br>>. Acesso em 20 jul. 2014 (com adaptações).

Com base no levantamento inicial das práticas da empresa, verifica-se que a evidência 1

- A. atende totalmente ao GRH 8, enquanto a evidência 2 atende totalmente ao GRH 6 e ao GRH 7.
- B. atende parcialmente ao GRH 8, enquanto a evidência 2 atende totalmente ao GRH 6 e ao GRH 7.
- C. atende parcialmente ao GRH 8, enquanto a evidência 2 atende parcialmente ao GRH 6 e ao GRH 7.
- D. atende totalmente ao GRH 8, enquanto a evidência 2 não atende ao GRH 6, mas atende totalmente ao GRH 7.
- E. atende parcialmente ao GRH 8, enquanto a evidência 2 atende totalmente ao GRH 6, mas não atende ao GRH 7.

⁴Questão 26 - Enade 2014.

1. Introdução teórica

Melhoria de software

O MPS.BR é um programa de longo prazo, criado em dezembro de 2003, coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX), que tem como objetivo a melhoria de processos de software e serviços.

Esse programa apresenta duas metas, conforme segue.

- Meta técnica: visa à criação e ao aprimoramento do Modelo MPS.
- Meta de negócio: é a disseminação e a adoção do Modelo MPS em todas as regiões do país, tanto em micro, pequenas e médias empresas quanto em grandes organizações privadas e governamentais.

O modelo MPS baseia-se nos conceitos de maturidade e capacidade de processo para a avaliação e a melhoria da qualidade e da produtividade de software e de serviços correlatos. Também visa à melhoria da qualidade e da produtividade dos serviços prestados.

O modelo MPS estabelece sete níveis de maturidade, que mostram os patamares de evolução de processos e os estágios de melhoria de implementação dos processos na organização. O nível de maturidade em que se encontra uma organização permite prever o seu desempenho futuro ao executar um ou mais processos.

Os sete níveis de maturidade são os que seguem.

- A. Em otimização.
- B. Gerenciado quantitativamente.
- C. Definido.
- D. Largamente definido.
- E. Parcialmente definido.
- F. Gerenciado.
- G. Parcialmente gerenciado.

A escala de maturidade inicia no nível G e progride até o nível A. Para cada um desses níveis de maturidade, é atribuído um perfil de processos que indica o foco em que a organização deve colocar o esforço de melhoria.

O quadro 1 apresenta os níveis de maturidade e os respectivos processos.

Quadro 1. Níveis de maturidades e processos nos quais devem ser aplicados esforços.

Nível de maturidade	Processos
A	
B	Gerência de Projetos
C	Gerência de Riscos Desenvolvimento para Reutilização Gerência de Decisões
D	Verificação Validação Projeto e Construção do Produto Integração do Produto Desenvolvimento de Requisitos
E	Gerência de Projetos Gerência de Reutilização Gerência de Recursos Humanos Definição do Processo Organizacional Avaliação e Melhoria do Processo Organizacional
F	Medição Garantia da Qualidade Gerência de Portfólio de Projetos Gerência de Configuração Aquisição
G	Gerência de Requisitos Gerência de Projetos

Uma organização que se encontra no nível E, por exemplo, deve colocar esforços nos processos elencados entre os níveis G e E.

Considerando, ainda, o nível E, nele é encontrado o processo de Gerência de Recursos Humanos (GRH), cujo propósito é prover a organização e os projetos com os recursos humanos necessários e manter suas competências adequadas às necessidades do negócio. Os resultados esperados para o processo GRH são classificados conforme segue.

- **GRH 1.** As necessidades estratégicas da organização e dos projetos são revistas para identificar recursos, conhecimentos e habilidades requeridos e, de acordo com a necessidade, são feitos planejamentos de como desenvolvê-los ou contratá-los.
- **GRH 2.** Indivíduos com as habilidades e competências requeridas são identificados e recrutados.
- **GRH 3.** As necessidades de treinamento de responsabilidade da organização são identificadas.
- **GRH 4.** Uma estratégia de treinamento é definida, com o objetivo de atender às necessidades de treinamento dos projetos e da organização.
- **GRH 5.** Um plano tático de treinamento é definido, com o objetivo de implementar a estratégia de treinamento.

- **GRH 6.** Os treinamentos identificados como de responsabilidade da organização são conduzidos e registrados.
- **GRH 7.** A efetividade do treinamento é avaliada.
- **GRH 8.** Critérios objetivos para avaliação do desempenho de grupos e indivíduos são definidos e monitorados para prover informações sobre o desempenho e melhorá-lo.
- **GRH 9.** Uma estratégia apropriada de gerência de conhecimento é planejada, estabelecida e mantida para compartilhar informações na organização.
- **GRH 10.** Uma rede de especialistas na organização é estabelecida e um mecanismo de apoio à troca de informações entre os especialistas e os projetos é implementado.
- **GRH 11.** O conhecimento é disponibilizado e compartilhado na organização.

Deve ser observado que a evolução de um nível para outro ocorre quando todos os processos do nível atual (e de todos os outros que o antecedem) são satisfeitos.

2. Análise da questão.

A evidência 1, segundo a qual a “empresa possui um processo de avaliação de desempenho e promoção de indivíduos bem estabelecido, em plano de cargos e carreira”, trata da avaliação e do plano de carreira. Essa evidência satisfaz parcialmente o GRH 8, pois, embora exista um processo de avaliação, não está especificado o monitoramento dos indivíduos para prover informações sobre o desempenho e melhorá-lo.

A evidência 2, segundo a qual a “empresa contrata externamente instrutores capacitados para realizar treinamentos internos a seus funcionários sempre que verifica a necessidade de ampliar a capacitação da equipe”, sendo que “esses treinamentos, registrados como parte do portfólio de cada funcionário, são avaliados pelos funcionários em relação à qualidade e à aderência às necessidades de capacitação da equipe”. A avaliação de cada funcionário ao final de cada treinamento atende totalmente aos processos do GRH 6 e do GRH 7. A empresa identifica as necessidades de treinamento, isto é, ela tem a noção de que a capacitação do pessoal é de sua responsabilidade e que existe a avaliação da efetividade do treinamento.

Alternativa correta: B.

NOTA. A introdução teórica e a análise da questão foram elaboradas tendo como referência o Guia geral SOFTEX de 2012, vigente à época da elaboração do Enade 2014.

3. Indicação bibliográfica

- ASSOCIAÇÃO PARA PROMOÇÃO DA EXCELÊNCIA DO SOFTWARE BRASILEIRO – SOFTEX. *MPS.BR - Melhoria de Processo do Software Brasileiro*. Brasília: SOFTEX, 2012. Disponível em <http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_Geral_Software_2012-c-ISBN-1.pdf>. Acesso em 07 jun. 2017.

Questão 5

Questão 5.⁵

A internet encontra-se organizada em Sistemas Autônomos (SA), que são administrados autonomamente por uma entidade ou organização. Para que uma comunicação ocorra com sucesso entre hospedeiros situados em SAs diferentes, é necessário que pacotes sejam eficientemente transportados do SA origem ao SA destino. Protocolos de roteamento são responsáveis pelo estabelecimento de rotas a serem seguidas pelos pacotes. Para que essas rotas sejam estabelecidas, deve-se usar

- A. Border Gateway Protocol (BGP).
- B. Open Shortest Path First (OSPF).
- C. Routing Information Protocol (RIP).
- D. Interior Gateway Routing Protocol (IGRP).
- E. Intermediate System to Intermediate System (IS-IS).

1. Introdução teórica

1.1. Protocolos de roteamento

A internet é uma rede gigantesca de computadores que cresceu essencialmente interligando redes de diferentes empresas e organizações. Seu caráter distribuído é semelhante ao sistema de transporte de uma cidade ou de um país: existem ruas e avenidas conectando diferentes lugares. Porém alguns locais podem ser tão grandes que podem ter ruas e avenidas internas (como um grande condomínio fechado, por exemplo).

Redes complexas oferecem caminhos diversos para o tráfego da informação. O processo de roteamento deve encontrar um caminho adequado para esse tráfego. Como a internet interliga diferentes redes, um pacote de dados pode ter destinos distintos: interno a um sistema autônomo ou externo a um sistema autônomo, enviando pacotes entre diferentes sistemas autônomos.

Os primeiros protocolos de roteamento interno, chamados de Interior Gateway Routing Protocol (IGRP), baseavam-se no algoritmo de Bellman-Ford, vindo das primeiras versões da ARPANET. Como exemplos, temos o protocolo Routing Information Protocol (RIP), usado até hoje, embora apresente escalabilidade limitada, e os protocolos Open Shortest Path First (OSPF) e Intermediate-System to Intermediate-System (IS-IS).

⁵Questão 20 – Enade 2014.

Para o roteamento entre diferentes sistemas autônomos, emprega o protocolo Border Gateway Protocol (BGP). Seu uso é necessário em virtude da diferença de objetivos entre o protocolo IGRP e o protocolo BGP. Isso porque, para um protocolo intradomínio, o objetivo é garantir pacotes da forma mais eficiente possível entre uma fonte e um destino. A comunicação interdomínio requer que se leve em consideração outros fatores, de ordem econômica ou política, o que implica alterações na sua forma de funcionamento.

2. Análise das alternativas

A - Alternativa correta.

JUSTIFICATIVA. O BGP é um protocolo de comunicação entre diferentes sistemas autômatos.

B - Alternativa incorreta.

JUSTIFICATIVA. O OSPF é um protocolo de comunicação intradomínio que trabalha em um mesmo sistema autômato.

C – Alternativa incorreta.

JUSTIFICATIVA. O RIP é um protocolo de comunicação intradomínio que trabalha em um mesmo sistema autômato (normalmente de pequeno ou de médio porte).

D - Alternativa incorreta.

JUSTIFICATIVA. Os protocolos IGRP são utilizados em um mesmo sistema autômato.

E - Alternativa incorreta.

JUSTIFICATIVA. O IS-IS é um protocolo de comunicação intradomínio que trabalha em um mesmo sistema autômato.

3. Indicação bibliográfica

- TANENBAUM A. S.; WETHERALL D. J. *Computer networks*. Upper Saddle River: Prentice Hall, 2011.

Questão 6

Questão 6.⁶

A área de complexidade de algoritmos abrange a medição da eficiência de um algoritmo frente à quantidade de operações realizadas até que ele encontre seu resultado final.

A respeito desse contexto, suponha que um arquivo texto contenha o nome de N cidades de determinado estado, que cada nome de cidade esteja separado do seguinte por um caractere especial de fim de linha e classificado em ordem alfabética crescente. Considere um programa que realize a leitura linha a linha desse arquivo, à procura de nome de cidade. Com base nessa descrição, verifica-se que a complexidade desse programa é

- A. $O(1)$, em caso de busca sequencial.
- B. $O(N)$, em caso de busca sequencial.
- C. $O(\log_2 N)$, em caso de busca binária.
- D. $O(N)$, em caso de transferência dos nomes para uma árvore binária e, então, realizar a busca.
- E. $O(\log_2 N)$, em caso de transferência dos nomes para uma árvore binária e, então, realizar a busca.

1. Introdução teórica

Complexidade de algoritmos e notação do O-grande

Os problemas computacionais são, normalmente, divididos em classes de complexidade. O objetivo dessa classificação é encontrar uma forma de separar os problemas complexos dos problemas mais simples, já que os primeiros requerem grande capacidade computacional e os últimos requerem recursos computacionais mais modestos.

Aqui, estamos utilizando a palavra complexo de uma forma objetiva: a questão não é se o algoritmo é de fácil ou é de difícil compreensão, mas sim se sua execução pelo computador requer muitos recursos em termos de memória e/ou de tempo de execução.

Por que investir tempo para caracterizar matematicamente a complexidade de um algoritmo se é possível simplesmente executá-lo? Não seria mais simples apenas executar o programa e cronometrar quanto tempo leva para a sua execução ou monitorar o seu perfil de execução e observar o seu consumo de recursos?

⁶Questão 10 – Enade 2014.

De fato, frequentemente, executam-se programas como uma forma de verificar seu próprio desempenho. E, ao fazer uma análise matemática detalhada do seu funcionamento, pode-se compreender muito melhor a natureza da sua execução. Nesse sentido, deve ser considerado que o hardware do computador exerce influência sobre o tempo de execução. Outro item a ser avaliado é a qualidade da implementação de um algoritmo, que também pode afetar o seu tempo de execução. Uma análise puramente matemática permite o estabelecimento de um critério mais consistente de comparação, independentemente do hardware utilizado e da implementação específica.

Alguns algoritmos são tão complexos do ponto de vista computacional que não importa o quão cuidadosamente seja realizada sua construção, seu tempo de execução e seu consumo de memória sempre serão extremamente elevados. Esses casos, em que uma solução exata do problema é impossível do ponto de vista prático (a solução matemática ótima existe, mas o seu cálculo requer uma quantidade de recursos absurdamente elevada), são chamados de problemas intratáveis. Isso não é o mesmo que dizer que o problema é impossível de ser resolvido de forma teórica: a solução teórica existe, mas requer capacidades computacionais extraordinárias.

Essa necessidade leva ao problema da caracterização matemática dos algoritmos. Algoritmos podem ser caracterizados do ponto de vista do seu tempo de execução ou do espaço (memória) necessário para o seu funcionamento.

Se x é um número inteiro que representa o tamanho da entrada de um algoritmo (por exemplo, no caso de um algoritmo de ordenação, x é o número total de elementos a serem ordenados), devemos associar a essa variável uma função $y=f(x)$ que representa o tempo de execução do algoritmo.

Normalmente, não estamos interessados no tempo exato de execução, nem nos valores de $f(x)$ para valores pequenos de x . O objetivo é caracterizar o crescimento da função para valores elevados de x e cotejar funções de diferentes algoritmos, o que fornece um critério de comparação do tempo de execução.

Com base nisso, concluímos que algoritmos com tempos de execução exponencial, indicados por $O(e^x)$, levam tempos muito maiores para serem executados do que algoritmos com tempos de execução polinomial, indicados por $O(x^n)$. Ou seja, para valores elevados de x , algoritmos do tipo $O(x^n)$ são mais rápidos do que algoritmos do tipo $O(e^x)$.

Formalmente, a notação do O-grande pode ser definida como indicado a seguir.

Sejam f e g duas funções tais que $f, g : N \rightarrow \mathbb{R}^+$.

Dizemos que $f(n) = O(g(n))$ se dois inteiros positivos c e n_0 existirem tal que, para cada inteiro $n \geq n_0$, tenhamos:

$$f(n) \leq c \cdot g(n)$$

Quando $f(n) = O(g(n))$, dizemos que $g(n)$ é um limite superior para $f(n)$, ou, mais precisamente, que $g(n)$ é um limite superior assintótico para $f(n)$, enfatizando que estamos suprimindo fatores constantes.

2. Resolução da questão

O enunciado especifica que o algoritmo faz uma busca sequencial, uma vez que é feita a leitura linha a linha do arquivo. Dessa forma, o algoritmo não se aproveita do fato de a lista estar ordenada. Além disso, segundo o enunciado, não se utiliza nenhuma estrutura de dados adicional auxiliar. Fazendo uma busca sequencial, o algoritmo percorre cada linha do arquivo como se percorre um array e verifica se ele é igual ao nome da cidade. Quanto maior o array, mais tempo o algoritmo vai levar para encontrar o elemento desejado, sendo esse tempo diretamente proporcional ao tamanho do array, ou seja, $O(N)$.

Alternativa correta: B.

3. Indicações bibliográficas

- CORMEN, T.; LEISERSON, C.; RIVEST, R.; STEIN, C. *Introduction to algorithms*. 3. ed. Cambridge: MIT Press, 2009.
- SIPSER, M. *Introduction to the theory of computation*. 3. ed. Boston: Cengage, 2012
- TOSCANI, L. V.; VELOSO, P. A. S. *Complexidade de algoritmos*. 2. ed. Porto Alegre: Bookman, 2008.

Questões 7 e 8

Questão 7.⁷

Leia o texto a seguir.

Way back in the late twentieth century, 1997 to be exact, the Object Management Group (MG) released the Unified Modeling Language (UML). One of the purposes of UML was to provide the development community with a stable and common design language that could be used to develop and build computer applications. UML brought forth a unified standard modeling notation that IT professionals had been wanting for years. Using UML, IT professionals could now read and disseminate system structure and design plans, just as construction workers have been doing for years with blueprints of buildings.

It is now the twenty-first century, 2003 to be precise, and UML has gained traction in our profession. On 75 percent of the resumes I see, there is a bullet point claiming knowledge of UML. However, after speaking with a majority of these job candidates, it becomes clear that they do not truly know UML. Typically, they are either using it as a buzzword, or they have had a sliver exposure to UML.

BELL, D. *UML basics: an introduction to the unified modeling language*. IBM Corporation, 2003. Disponível em <<http://www.ibm.com>>. Acesso em 28 jul. 2014 (com adaptações).

Com base no texto nos conceitos da UML, avalie as afirmativas a seguir.

- I. Na maioria dos currículos analisados pelo autor, existe alguma indicação de conhecimento da linguagem UML pelo candidato ao emprego.
- II. A UML é composta por diversos diagramas, como diagrama de caso de uso, diagrama de especificação e diagrama de componentes.
- III. A UML não é uma linguagem de programação, mas sim, uma linguagem de modelagem padronizada para os profissionais de TI cuja notação independe de processos.

É correto o que se afirma em

- A. I, apenas.
- B. II, apenas.
- C. I e III, apenas.
- D. II e III, apenas.
- E. I, II e III.

Questão 8.⁸

Uma empresa deseja lançar um sistema de comércio eletrônico para vender seus produtos. Essa empresa vende produtos de diversas categorias, como roupas, perfumes e eletrônicos, e aceita diversas formas de pagamento, como cartão de crédito e boleto bancário. No sistema de vendas implementado, cada produto deve ser cadastrado com sua descrição, preço de venda, quantidade em estoque e respectiva categoria. Cada cliente que deseja

⁷Questão 35 – Enade 2014.

⁸Questão Discursiva 5 – Enade 2014.

realizar compras tem de se cadastrar no sistema indicando seu nome, endereço e e-mail. Se o cliente for corporativo, deve cadastrar seu CNPJ e, se for individual, seu CPF. O cliente cadastrado pode realizar um pedido de compra dos produtos em estoque na quantidade que desejar. O cliente escolhe uma forma de pagamento disponível e recebe, por e-mail, o número do pedido e as informações do status do pedido. Após a confirmação do pagamento, a loja realiza a entrega dos itens solicitados no endereço do cliente e envia, por e-mail, a nota fiscal eletrônica. Tendo em vista que os preços dos produtos podem ser atualizados a qualquer momento, o sistema tem de ser capaz de reemitir uma nota fiscal de um pedido de compra de qualquer produto e respectivo preço na data da compra realizada pelo cliente.

Considerando esse cenário, proponha um Diagrama de Classes, segundo a UML (*Unified Modeling Language*), indicando nome de cada classe, respectivos atributos e relacionamentos entre as classes com as respectivas cardinalidades. Em sua proposta, identifique, pelo menos, um relacionamento de generalização e um relacionamento de composição, não sendo necessário indicar as operações de cada classe.

1. Introdução teórica

1.1. Unified Modeling Language (UML)

Formalmente, a Unified Modeling Language (UML) é definida como uma linguagem-padrão para a elaboração da estrutura de projetos de software.

Ainda nesse texto, os autores destacam os quatro objetivos básicos da UML, citados a seguir.

- Visualizar os artefatos de um sistema complexo de software.
- Especificar os artefatos de um sistema complexo de software.
- Construir os artefatos de um sistema complexo de software.
- Documentar os artefatos de um sistema complexo de software.

É importante destacar que o principal objetivo da UML está no estabelecimento de um modelo ou de uma representação conceitual e física de um sistema.

1.1.1. UML para visualização

Um projeto de software envolve mais do que simplesmente seu código fonte, visto que um mesmo problema pode ser resolvido de formas distintas, com o uso de códigos muito diferentes. Ainda que programas com códigos diferentes possam atender aos mesmos requisitos, nem todas as soluções são igualmente úteis ou interessantes.

Se dispusermos apenas do código fonte de um programa, sem nenhum tipo de documentação, o entendimento da solução é dificultado. Programas comerciais podem ser muito longos e conter milhões de linhas de código. Dessa forma, é importante ter uma maneira de se visualizar o todo, a estrutura geral de um software, sem necessariamente ser obrigado a ler o código fonte de modo completo. Desenhos e diagramas informais podem ajudar nessa tarefa, mas a UML fornece uma linguagem padronizada que permite que diferentes pessoas e empresas façam e interpretem diagramas e modelos de uma mesma forma.

1.1.2. UML para especificação

Antes de se escrever o código para um programa, é preciso conhecer quais são exatamente as necessidades do cliente. Além disso, grandes projetos envolvem problemas complexos de software e muitas empresas dispõem de profissionais especializados para encontrar a melhor solução, como os arquitetos de software. Esses profissionais precisam de uma linguagem para especificar, projetar e comunicar a solução: a UML oferece uma linguagem especificamente desenvolvida para esse propósito.

1.1.3. UML para construção

Ainda que a UML não seja uma linguagem de programação propriamente dita, é possível estabelecer um mapeamento entre os modelos construídos pela UML e o código em diversas linguagens de programação, como Java ou C#.

Essa facilidade fornecida pela UML no mapeamento vem do alto grau de expressividade aliado à baixa ambiguidade, o que não é caso de outras formas de especificação, como um texto puro. Tal precisão faz com que existam ferramentas capazes de converter alguns diagramas UML em código fonte com pouca ou nenhuma intervenção do usuário.

1.1.4. UML para documentação

Um projeto de software envolve mais do que apenas código. Empresas de software costumam produzir diversos outros elementos, chamados de artefatos, como os citados a seguir.

- Requisitos do software.
- Documentação da arquitetura.
- Projeto e planos de projeto.
- Código fonte.
- Testes e planos de teste.
- Protótipos.

A UML pode ser utilizada para documentar diversos desses artefatos, formalmente e com baixa ambiguidade.

1.2. Diagrama de classes

Formalmente, define-se uma classe como

uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica (BOOCH, RUMBAUGH e JACOBSON, 2006).

É importante diferenciar classes de objetos: uma classe define um conjunto de objetos, mas não é um objeto. Por exemplo, podemos definir uma classe das “xícaras de café” como objetos cerâmicos e ocios que servem para armazenar quantidade pequena de líquido quente. Essa “classe” abrange todas as possíveis xícaras, de diferentes formas e tamanhos. Se considerarmos um caso específico, como, por exemplo, determinada xícara utilizada agora em um restaurante, dizemos que essa xícara é um objeto ou uma instância específica de uma classe. Em resumo, utilizamos uma classe para definir uma categoria de objetos similares.

A UML apresenta um diagrama específico para a representação das classes, chamado de diagrama de classes. Nesse diagrama, as classes são simbolizadas por retângulos contendo o nome, os atributos e as operações, como ilustrado na figura 1. Os atributos são propriedades associadas a cada um dos objetos instanciados pela classe. Por exemplo, a classe “Cliente” pode ter um atributo chamado de “nome”. Outros atributos que essa classe pode ter são: “endereço”, “telefone” e “email”.

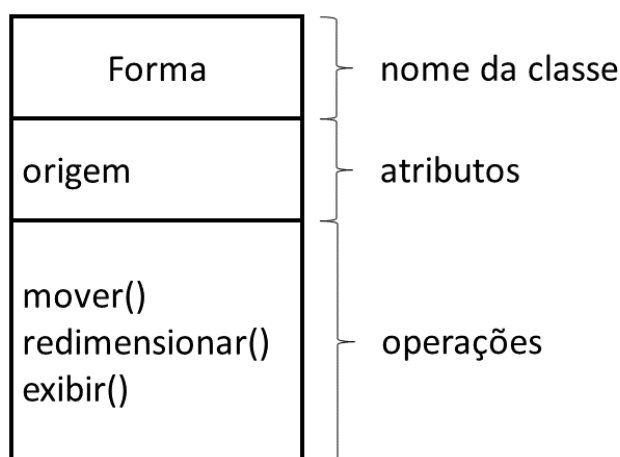


Figura 1. Representação de uma classe na UML.
BOOCH, RUMBAUGH e JACOBSON, 2006 (com adaptações).

Além de atributos, uma classe também apresenta comportamentos associados. Por exemplo, suponha um editor de imagens que ofereça uma funcionalidade produzir desenhos de formas geométricas, como losangos. Essa funcionalidade pode ser implementada por meio de uma classe, chamada de “losango”, que tem uma série de comportamentos associados, como a movimentação do losango e a alteração do seu tamanho, entre outras. Essas operações são representadas dentro do retângulo da classe, logo abaixo dos atributos.

As operações têm um nome e costumam terminar com parênteses “()”, o que indica o fato de essas operações serem similares a funções e poderem receber parâmetros. Os parâmetros são representados dentro dos parênteses.

As diversas classes que compõem um programa podem estar relacionadas entre si. Existem essencialmente três tipos de relacionamentos entre classes:

- dependência;
- generalização;
- associação.

Utilizamos uma relação de dependência quando queremos dizer que uma classe necessita de outra classe para o seu funcionamento. Por exemplo, suponha uma classe chamada “Cachorro” e uma classe chamada “Animal”. Podemos dizer que “Animal” representa uma generalização de “Cachorro”, uma vez que todo cachorro é um animal, mas nem todo animal é um cachorro.

Finalmente, uma associação é um relacionamento semântico entre dois elementos do modelo, ou seja, estabelece o motivo pela qual duas classes se relacionam e as regras que controlam esse relacionamento.

2. Resolução das questões

Questão 7

I – Afirmativa correta.

JUSTIFICATIVA. No texto do enunciado, o autor declara que mais de 75% dos currículos têm alguma menção ao conhecimento da UML.

II – Afirmativa incorreta.

JUSTIFICATIVA. A UML não se limita a um conjunto de diagramas (ainda que alguns diagramas façam parte da sua estrutura). A UML é, na realidade, uma linguagem para modelagem de software.

III – Afirmativa correta.

JUSTIFICATIVA. A padronização de uma linguagem de modelagem facilita a comunicação entre os diversos participantes do projeto e, também, entre as diversas empresas. A independência com relação à linguagem de programação garante que uma mesma estratégia de modelagem possa ser utilizada em diferentes empresas e em diferentes projetos pelo uso de tecnologias variadas.

Alternativa correta: C.

Questão 8

As classes citadas a seguir devem ser incorporadas ao modelo.

- Cliente
- Corporativo
- Indivíduo
- Produto
- Item_pedido
- Forma_pagamento
- Pedido_Compra
- Categoria

O diagrama deve assemelhar-se ao apresentado na figura 2.

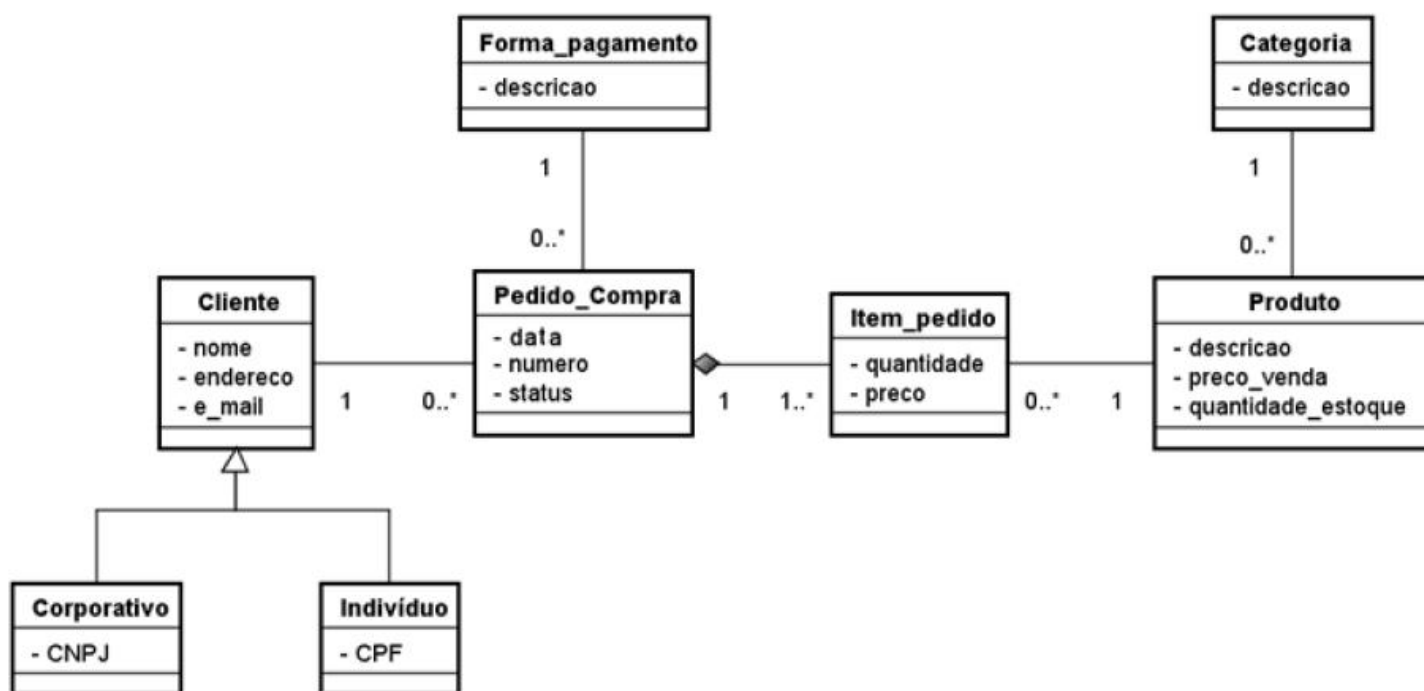


Figura 2. Diagrama de classes do exercício.

Gabarito do Inep para o Enade de Sistemas da Informação (2014).

A classe cliente é uma generalização das classes “Corporativo” e “Indivíduo”, que representam tipos específicos de cliente (observe a notação de generalização no diagrama). Cada cliente pode realizar um pedido de compra, que é representado pela classe “Pedido_Compra”. Essa classe está associada à classe “Cliente”, mas é importante observar que existe uma multiplicidade na associação: apenas um objeto da classe “Cliente” pode fazer zero ou mais pedidos, representado por objetos da classe “Pedido_Compra” (observe os números na extremidade da linha ligando as classes “Cliente” e “Pedido_Compra”). O relacionamento entre a classe “Pedido_Compra” e “Item_pedido” é um tipo especial de associação, chamado de agregação, representada por um losango. No diagrama, um objeto da classe “Pedido_Compra” agrega um ou mais objetos do tipo “Item_pedido” (uma vez que pedido deve conter ao menos um item). Um objeto do tipo “Item_pedido” se relaciona com um objeto do tipo “Produto” por meio de uma associação. Um objeto do tipo “Categoria” se relaciona com zero ou mais objetos do tipo “Produto” também por meio de uma associação. Na mesma figura, é possível observar os atributos de cada uma das classes na figura, que foram diretamente retirados da descrição no enunciado. O diagrama não contém as operações, que provavelmente serão identificadas em uma etapa posterior da modelagem.

3. Indicações bibliográficas

- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: Guia do Usuário*. Rio de Janeiro: Campus, 2006.
- PENDER, T. *UML: a Bíblia*. Rio de Janeiro: Campus, 2004.

Questão 9

Questão 9.⁹

Uma aplicação mantém um grande conjunto de dados ordenados em uma lista linear em memória, implementada na forma de array. Os dados são estáveis, ou seja, são realizadas poucas operações de inserção e remoção, e a aplicação exige grande quantidade de pesquisas. Sabe-se que vários algoritmos permitem realizar pesquisas em um conjunto de dados, sendo alguns mais eficientes que outros. Nesse sentido, a pesquisa binária apresenta características bem definidas para se encontrar um dado procurado.

Considerando o contexto apresentado, desenvolva um algoritmo de pesquisa binária, de forma imperativa ou orientada a objetos, utilizando qualquer notação em português estruturado ou em uma linguagem de alto nível, como o Pascal, C ou Java.

1. Introdução teórica

1.1. Estruturas de dados: arrays

As linguagens de programação costumam prover recursos para a organização eficiente dos dados, como as estruturas de dados estáticas.

As estruturas de dados estáticas apresentam um espaço fixo em memória. Esse espaço é fixado pelo programador no momento em que o programa está sendo escrito (tempo de compilação). Logo, o programador tem como prever exatamente o tamanho de memória que será consumido pelo programa. Além disso, esse profissional tem como saber os limites de todas as estruturas de memória antecipadamente e de forma bastante simples.

Como o tamanho dessas estruturas é fixo, não é possível alocar mais espaço em memória caso se tenha essa necessidade. Isso significa que o programador deve saber quanto espaço de memória deve ser necessário alocar para o programa durante a sua escrita ou antes da sua compilação. Se o programador não tiver como saber a quantidade de memória necessária com antecedência, sua única alternativa será alocar, preventivamente, grande quantidade de espaço, até muito mais do que o necessário, o que pode gerar desperdício de memória, mas com a garantia do funcionamento do programa.

Em muitas linguagens de programação, como na linguagem C, um array é implementado como um conjunto de variáveis (ou posições de memória) de um mesmo tipo, armazenadas sequencialmente.

⁹Questão Discursiva 3 – Enade 2014.

Essas variáveis formam uma espécie de “reticulado”, posições de memórias vizinhas e com o mesmo tamanho. Dessa forma, a declaração de um array envolve a definição de um tipo de variável, o estabelecimento de um tamanho máximo e o conhecimento da posição de memória inicial. Os elementos são acessados por meio de um índice, normalmente iniciado por zero.

Como todos os elementos são do mesmo tipo, dizemos que o array é uma estrutura de dados homogênea. Como os dados são acessados por um índice e de forma sequencial, também dizemos que o array é uma estrutura linear. Como o tamanho de um array é definido em tempo de compilação, seu tamanho é fixo e dizemos que ele é uma estrutura estática.

Uma forma gráfica de visualizarmos um array na linguagem C é por meio de um conjunto de células (ou posições de memória) vizinhas, como ilustrado na figura 1. Nessa imagem, foi criado um array chamado “Array”, com N elementos, no qual cada elemento é um número inteiro que ocupa um tamanho fixo na memória, dependendo da plataforma, e é acessado por meio de índices que iniciam da posição 0 vão até a posição N-1. Observe que isso totaliza N elementos.

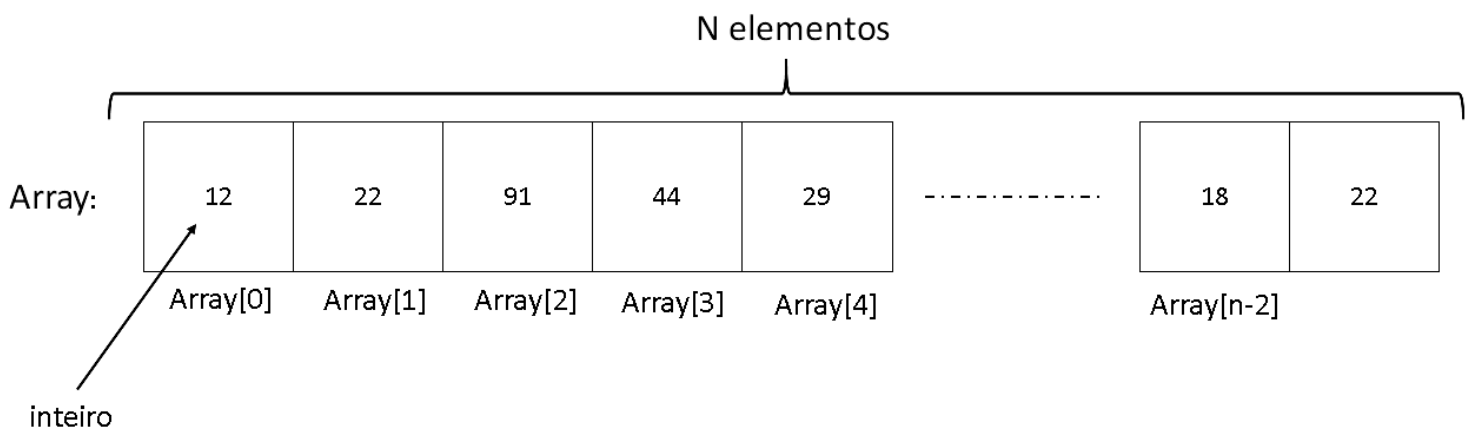


Figura 1. Um array de números inteiros.

Normalmente, as linguagens de programação também disponibilizam outras estruturas de dados mais poderosas, como os vetores. No caso da linguagem C++, os vetores têm uma notação similar à notação dos arrays, mas apresentam um mecanismo de alocação dinâmica de memória (eles podem “crescer” se adicionarmos novos elementos, não sendo necessário saber o seu tamanho no tempo de compilação).

1.2. Busca binária

Quando trabalhamos com um array previamente ordenado, a busca da informação pode ser feita de forma muito mais rápida. Em um array ordenado, a posição de cada elemento depende da sua relação com os demais e, assim, podemos utilizar essa informação para encontrar elementos mais rapidamente.

A ideia básica consiste em dividir a pesquisa em duas partições. Cada vez que testamos um elemento do array, dividimos o array em duas partes: os elementos posteriores e os elementos anteriores ao elemento atual. Como o array está ordenado, todos os elementos posteriores devem ser maiores do que o elemento atual e todos os elementos anteriores devem ser menores do que o elemento atual. Então, comparamos o valor desejado e verificamos quais das duas partes devemos atacar na próxima tentativa. Independentemente de estarmos buscando elementos maiores ou menores, podemos eliminar metade do array a cada teste. Isso implica grande ganho computacional, uma vez que o tempo de pesquisa é da ordem de $O(\log N)$ tanto para o caso médio quanto para o pior caso, o que é bem melhor do que uma busca sequencial típica, com tempo $O(N)$, em que N é o tamanho do array.

Em um algoritmo iterativo, utilizamos dois índices para a busca: um índice superior e um índice inferior. Inicialmente, o índice superior aponta para o final do array e o índice inferior aponta para a primeira posição. A cada iteração, calcula-se também um ponto médio, a posição intermediária entre os índices superiores e inferiores. O conteúdo dessa posição é comparado à chave, ou seja, ao elemento que estamos querendo encontrar. Se a chave for menor do que o elemento intermediário, atualizamos o índice superior para o valor imediatamente anterior a chave atual e mantemos o índice inferior (essencialmente descartando metade do array). Se a chave for maior, atualizamos o índice inferior para o elemento posterior a chave e mantemos o índice superior. Se a chave for igual ao elemento atual, significa que encontramos o elemento no array e, assim, retornamos o seu índice.

Nesse processo, estamos sempre atualizando um dos índices e o algoritmo continua até que o índice inferior se torne maior do que o índice superior. Nesse caso, o elemento não está presente no array e devemos retornar essa informação de alguma forma, por exemplo, retornando um índice negativo, como -1.

2. Resolução da questão

O código da solução, em C, parcialmente retirado de Shaffer (2011), pode ser visto a seguir.

```

1      #include<stdio.h>
2      int buscaBinaria(int A[], int n, int K){
3          int l = -1;
4          int r = n;
5          while( l+1 != r )
6          {
7              int i = (l+r)/2;
8              if( K < A[i] )
9              {
10                 r = i;
11             }
12             if( K == A[i])
13             {
14                 return i;
15             }
16             if (K > A[i])
17             {
18                 l = i;
19             }
20         }
21         return -1;
22     }
23     int main()
24     {
25         int novo = 0;
26         int vet[]={2,5,13,35,44,51,78,99,122,201};
27         int tamVet = 10;
28         int chave = 44;
29         int resp = 0;
30         printf("Iniciando a busca: \n");
31         resp = buscaBinaria(vet, tamVet, chave);
32         if( resp > -1 )
33         {
34             printf("Encontrou... %d \n",resp);
35         }else{
36             printf("Nao encontrou \n");
37         }
38         printf("---Fim --- \n");
39         return 0;
40     }

```

O algoritmo corresponde à função buscaBinaria, definida entre as linhas 2 e 22. A função “main” serve como exemplo de utilização da busca, definindo um vetor de 10 elementos e uma chave igual a 44. Como exercício, altere o programa para fazer com que o usuário seja capaz de entrar com o vetor e com a chave de busca e teste o programa para diferentes chaves.

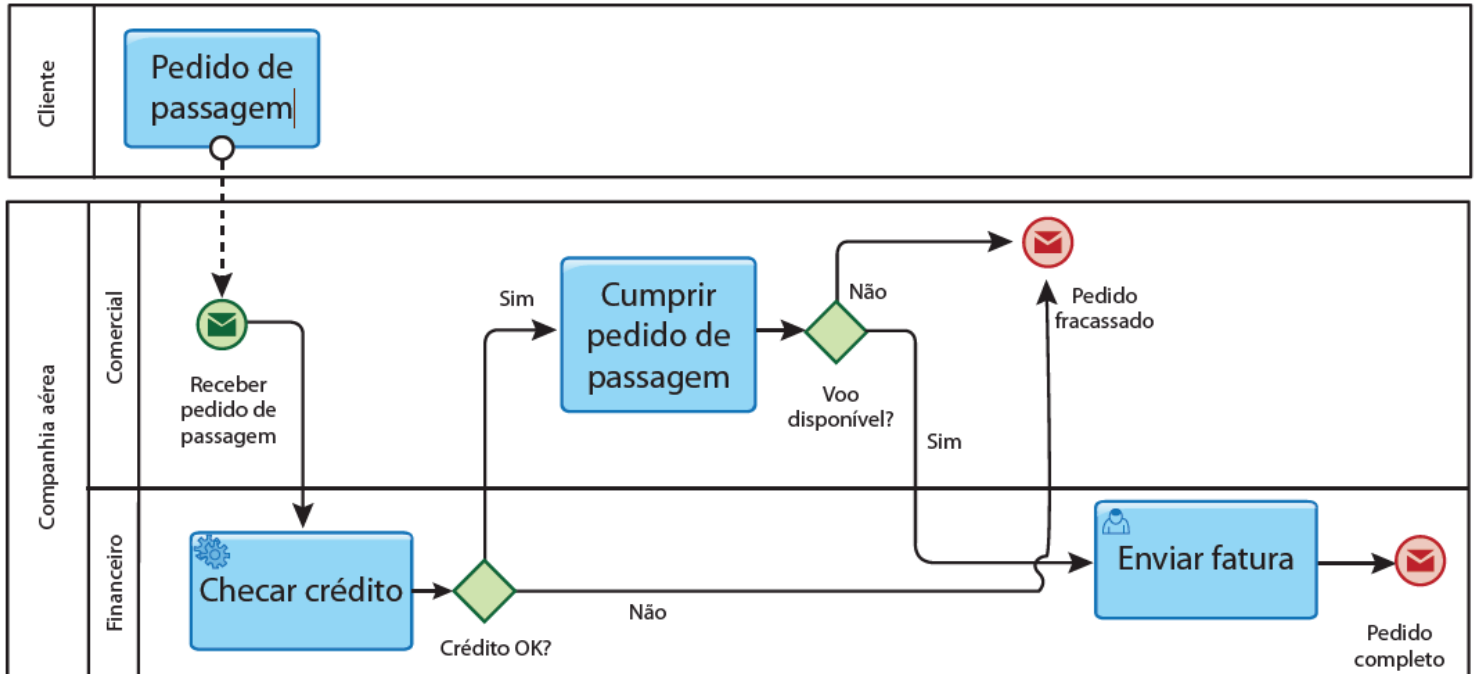
3. Indicações bibliográficas

- CELES, W.; CERQUEIRA, R.; RANGEL, J. R. *Introdução à estrutura de dados*. Rio de Janeiro: Campus, 2004.
- PAHUJA, S. *A Practical Approach to Data Structures and Algorithms*. New Delhi: New Age International, 2007
- RICHARDSON, D. R. *The book on Data Structures*. Lincoln: iUniverse, 2002, v. 1.
- SHAFFER, C. A. *Data Structures and Algorithm Analysis in C++*. Mineola: Dover, 2011.

Questão 10

Questão 10.¹⁰

Considere um cenário de venda de passagens aéreas cujo processo de negócio apresenta-se modelado no diagrama abaixo, usando-se BPMN (*Business Process Modeling and Notation*).



Para que essa companhia aérea obtenha mais retorno, será necessário:

- A. retirar o evento intermediário de tempo colocado após a tarefa "Pedido de passagem".
- B. adicionar uma tarefa para oferecimento de outro voo na raia "Comercial".
- C. modelar a tarefa "Enviar fatura" como tarefa de serviço (*service task*).
- D. transformar o gateway "Crédito OK?" em um gateway inclusivo.
- E. transformar a tarefa "Checar crédito" em um subprocesso.

1. Introdução teórica

1.1. Business Process Modeling and Notation (BPMN)

Em geral, as empresas trabalham com um conjunto sofisticado de processos, que devem ser convenientemente gerenciados pela administração. Para que isso possa ser feito, é preciso que exista efetivo entendimento de como esses processos funcionam, inclusive para a detecção de eventuais erros e para a proposta de melhorias.

¹⁰Questão 32 – Enade 2014.

Nessa perspectiva, o Business Process Modeling and Notation (BPMN) é um padrão de modelagem que pode ser aplicada para o entendimento e para o gerenciamento dos processos em uma empresa. O BPMN é composto por um único diagrama, chamado de Business Process Diagram (BPD) ou Diagrama de Processos de Negócio.

O escopo do BPMN é totalmente focado em processos (como o próprio nome já indica), sendo que outros tipos de modelagens, também úteis para uma organização, não são contemplados. Por exemplo, modelos da estrutura organizacional, modelos de dados e modelos de regras de negócio não são o foco da BPMN.

A BPMN apresenta três tipos básicos de processos, conforme segue.

- Processos privados: ocorrem dentro da organização.
- Processos abstratos: também chamados de processos públicos, ocorrem entre diferentes entidades de negócio.
- Processos de colaboração: modelam a interação entre dois ou mais processos de negócio.

1.2 Elementos gráficos da BPMN

O Diagrama de Processos de Negócio da BPMN é composto por essencialmente quatro elementos, ilustrados na figura 1:

- Atividades.
- Eventos.
- *Gateway* (decisões).
- Sequência de fluxo.



Figura 1. Elementos básicos da BPMN.
VALLE e de OLIVEIRA, 2013.

No BPMN, uma atividade é utilizada para representar algum tipo de trabalho que precisa ser feito para o funcionamento do processo que está sendo modelado.

Existem três tipos de atividades: tarefas, subprocesso e processo. Como o objetivo do diagrama é representar processos, essas atividades são representadas por um conjunto de símbolos. Contudo, um processo pode ser composto por outros subprocessos, que também podem ser complexos e podem ser representados por um quadro com um símbolo “+”, indicando que parte da sua estrutura não está sendo mostrada, como mostrado na figura 2. Uma tarefa é representada por um retângulo simples e é utilizada para representar processos que não têm mais de uma camada de detalhe.

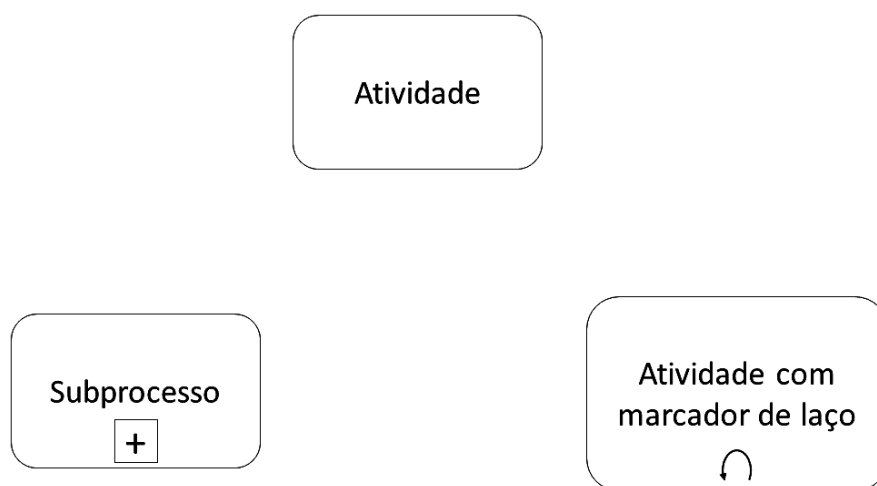


Figura 2. Tipos de atividades do BPMN.
VALLE e DE OLIVEIRA, 2013.

Além das atividades, também queremos representar eventos, que podem preceder, ser intermediários ou finalizar um processo. Eventos são representados por círculos, de acordo com o seu tipo, como representado na figura 3.



Figura 3. Tipos de evento do BPMN.
VALLE e DE OLIVEIRA, 2013.

Processos também envolvem decisões, que são representadas por um diamante. As decisões podem ser baseadas em dados ou eventos. Há também um símbolo para decisões que permitem a execução de múltiplas escolhas simultâneas. Esse o fluxo de execução posterior ao gateway pode envolver múltiplos caminhos. Os símbolos para gateways estão ilustrados na figura 4.



Figura 4. Tipos de Gateway.
VALLE e DE OLIVEIRA, 2013.

Para indicar a sequência de fluxo entre as atividades e os gateways, o BPMN utiliza conectores. Eles podem indicar a direção da sequência de fluxo, a direção do fluxo de mensagens e associação de elementos, como mostrado na figura 5.

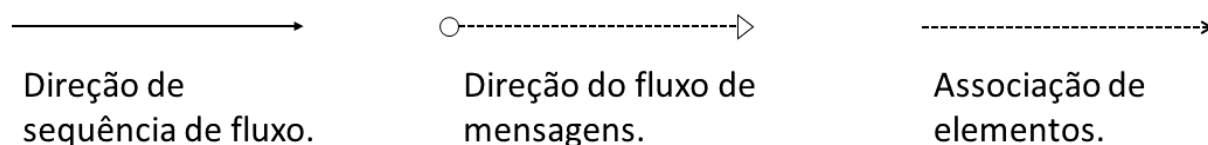


Figura 5. Tipos de conectores.
VALLE e DE OLIVEIRA, 2013.

Além dos elementos citados, o BPMN apresenta outros elementos, como os swimlanes, utilizados para dividir e organizar atividades, e os artefatos, empregados para mostrar informações complementares.

2. Análise das alternativas

A - Alternativa incorreta.

JUSTIFICATIVA. A retirada desse evento não faz sentido e não vai aumentar o retorno, uma vez que o evento é necessário.

B - Alternativa correta.

JUSTIFICATIVA. Ao adicionar essa tarefa, pode-se evitar que o pedido seja fracassado e abrir uma oportunidade de fazer outra venda, o que aumenta o retorno da empresa.

C - Alternativa incorreta.

JUSTIFICATIVA. É provável que a tarefa de enviar fatura necessite de algum envolvimento no caso da empresa do enunciado, talvez para algum tipo de checagem ou autorização em

algum sistema. Nesse caso, a tarefa deve ser modelada como user task. Evidentemente, se for possível a sua completa automação sem requerer nenhuma intervenção humana, essa tarefa pode tornar-se uma service task, mas isso depende do tipo de empresa e do tipo de negócio.

D - Alternativa incorreta.

JUSTIFICATIVA. Isso não pode ser feito, uma vez que o pedido só pode ser realizado se existir crédito, o que elimina a possibilidade de um gateway inclusivo.

E - Alternativa incorreta.

JUSTIFICATIVA. Isso não vai afetar o retorno da empresa, apenas a forma como o modelo foi construído.

3. Indicações bibliográficas

- DAVENPORT, T. *Reengenharia de processos*. Rio de Janeiro: Campus, 1994
- VALLE, R.; DE OLIVEIRA, S. B. *Análise e modelagem de processos de negócio: foco na notação BPMN*. São Paulo: Atlas, 2013.

ÍNDICE REMISSIVO

Questão 1	Modelagem de processos de negócios. Arquitetura de processos.
Questão 2	Gerência de projetos. Avaliação.
Questão 3	Gerência de projetos. Processos decisórios.
Questão 4	Gerência de projetos. Qualidade de processo e produto. Melhoria de software.
Questão 5	Redes. Protocolos de roteamento. Roteamento.
Questão 6	Complexidade de algoritmos. Notação O-Grande. Complexidade temporal.
Questão 7	UML. Diagramas de classe. Modelagem de software.
Questão 8	UML. Diagramas de classe. Modelagem de software.
Questão 9	Técnicas de programação. Arrays. Busca Binária.
Questão 10	BPMN. Modelagem de processos. Diagrama de processos de negócio.