EDUCAÇÃO, POR SUA ESCOLHA

CIÊNCIA DA COMPUTAÇÃO

MATERIAL INSTITUCIONAL ESPECÍFICO

TOMO 13

CQA - COMISSÃO DE QUALIFICAÇÃO E AVALIAÇÃO

CIÊNCIA DA COMPUTAÇÃO

MATERIAL INSTRUCIONAL ESPECÍFICO

TOMO 13

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.

Tiago Guglielmeti Correale

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

Questão 1

Questão 1.[1]

Considere os seguintes requisitos para desenvolvimento de uma solução para uma rede de restaurantes fast-food.

Quando o status de um pedido é atualizado, todos os dispositivos dos envolvidos devem receber a informação. Os sistemas a serem atualizados incluem os acessados pelo entregador, pela linha de produção e pela central de atendimento. Espera-se ainda que outros sistemas possam ser incluídos futuramente (por exemplo, sistema de pedido on-line do cliente), devendo se comportar da mesma forma.

Considerando esse contexto, avalie as asserções e a relação proposta entre elas.

I. O requisito apresentado pode ser implementado com a utilização do padrão de projeto Observer.

PORQUE

II. O padrão de projeto Observer realiza o estilo arquitetural cliente-servidor, no qual o servidor é responsável por enviar notificações aos clientes sempre que houver a atualização em alguma informação de interesse.

A respeito dessas asserções, assinale a opção correta.

A. As asserções I e II são proposições verdadeiras, e a asserção II justifica a I.

B. As asserções I e II são proposições verdadeiras, e a asserção II não justifica a I.

C. A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.

D. A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.

E. As asserções I e II são proposições falsas.

1. Introdução teórica

Padrões de projeto: Observer

A utilização de padrões de projeto auxilia a equipe de desenvolvimento a trabalhar com soluções que sejam previamente conhecidas e documentadas e cujo funcionamento seja coerente e adequado. Dessa forma, os desenvolvedores não "perdem o seu tempo reinventando a roda", ou seja, tendo que descobrir soluções que na realidade já existem.

Dado que projetos de software são normalmente bastante complexos, os tipos de problemas encontrados são variados e, muitas vezes, surgem em contextos particulares.

[1]Questão 10 – Enade 2017.

Mesmo assim, há similaridades com relação aos propósitos da sua aplicação. Gamma et al. (2007) encontram três grandes categorias ou propósitos de padrões de projeto:

- padrões de criação;
- padrões estruturais;
- padrões comportamentais.

Os padrões comportamentais procuram trabalhar com problemas ligados aos aspectos dinâmicos de um programa, enquanto os estruturais estão mais relacionados aos aspectos estáticos, ou, como o próprio nome diz, à estrutura do programa. Os padrões de criação trabalham com a forma como objetos de um programa são criados.

O padrão Observer é um padrão comportamental que funciona como uma espécie de mecanismo de inscrição. Para entender o seu funcionamento, vamos imaginar uma situação hipotética com pessoas, diferente de um programa de computador.

Imagine uma sala que tenha um representante de turma. Os seus colegas estão interessados em saber informações sobre o curso (por exemplo, a mudança de uma data de prova ou de entrega de um trabalho) e devem passar o seu e-mail para o representante de sala. Dessa forma, é como se cada aluno se "cadastrasse" com o representante de sala. Se o representante tiver alguma informação para ser transmitida para os seus colegas, ele mandará um e-mail para todos os alunos que se cadastraram inicialmente com ele.

Essa situação descreve o mecanismo básico do padrão Observer. No contexto do enunciado, existem diversos tipos de equipamentos (e provavelmente usuários) que devem se "cadastrar" para receber atualizações sobre o pedido. Os objetos que recebem essas atualizações costumam ser chamados de observers (observadores em português) ou subscribers (assinantes, em português). Já o objeto que notifica os demais da mudança de estado é chamado de subject, ou, mais apropriadamente, de publisher.

2. Análise das asserções

I – Asserção verdadeira.

JUSTIFICATIVA. O requisito de notificação de vários objetos é uma clara aplicação do padrão de projetos Observer.

II – Asserção falsa.

JUSTIFICATIVA. Não devemos ver o padrão de projetos Observer como a realização da arquitetura cliente-servidor, já que os dois conceitos são diferentes. Sabemos que, na

arquitetura cliente-servidor, vários clientes se comunicam e fazem requisições a um servidor, que atende a essas requisições. É possível utilizar o padrão Observer em uma arquitetura cliente servidor, mas isso não significa que esse padrão implique a utilização dessa arquitetura.

Alternativa correta: C.

3. Indicações bibliográficas

- FOWLER, M. *Padrões de arquitetura de aplicações corporativas*. Porto Alegre: Bookman, 2018.
- FREEMAN, E. et al. *Head first design patterns*. 2. ed. Sebastopol: O'Reilly, 2014.
- GAMMA, E. et al. *Padrões de projetos: soluções reutilizáveis de software orientados a objetos*. Porto Alegre: Bookman, 2000.
- LARMAN, C. *Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objetos e ao desenvolvimento iterativo*. 3. ed. Porto Alegre: Bookman, 2007.
- SHVETS, A. *Observer*. 2020. Disponível em <https://refactoring.guru/design-patterns/observer>. Acesso em 19 mar. 2026.

Questão 2

Questão 2.[2]

O encapsulamento é um mecanismo da programação orientada a objetos no qual os membros de uma classe (atributos e métodos) constituem uma caixa preta. O nível de visibilidade dos membros pode ser definido pelos modificadores de visibilidade "privado", "público" e "protegido".

Com relação ao comportamento gerado pelos modificadores de visibilidade, assinale a opção correta.

A. Um atributo privado pode ser acessado pelos métodos privados da própria classe e pelos métodos protegidos das suas classes descendentes.
B. Um atributo privado pode ser acessado pelos métodos públicos da própria classe e pelos métodos públicos das suas classes descendentes.
C. Um membro público é visível na classe à qual ele pertence, mas não é visível nas suas classes descendentes.
D. Um método protegido não pode acessar os atributos privados e declarados na própria classe.
E. Um membro protegido é visível na classe à qual pertence e em suas classes descendentes.

1. Introdução teórica

Modificadores de visibilidade e o conceito de encapsulamento

Um dos conceitos fundamentais da orientação a objetos é o conceito de encapsulamento. A ideia básica desse conceito é a de que não devemos conhecer os detalhes de implementação de um objeto específico ou o seu mecanismo de funcionamento. Isso precisa ser feito pela independência entre as várias classes de uma aplicação, que devem ser projetadas para "esconder" o seu funcionamento, expondo apenas uma interface (preferencialmente mínima).

Para auxiliar no encapsulamento, as linguagens de programação orientadas a objetos costumam oferecer modificadores de visibilidade. Esses modificadores permitem reduzir a exposição de métodos e atributos de uma classe às demais, controlando o grau máximo de acoplamento entre os diversos objetos.

[2]Questão 11 – Enade 2017.

Ainda que as diferentes linguagens de programação ofereçam diferentes tipos de modificadores de acesso (e algumas linguagens não oferecem nenhum desses modificadores), eles são bastante comuns na maioria das linguagens orientadas a objetos.

Vamos analisar o caso da linguagem C#. Nessa linguagem, o caso mais "aberto", ou seja, aquele que protege menos o atributo, ou método, é chamado de *public*, ou público. Quando um método ou atributo é público, ele pode ser visto por objetos de qualquer outra classe. O oposto da visibilidade pública é a privada, ou *private.* Nesse caso, a visibilidade restringe-se apenas à classe na qual o método, ou atributo, foi definido. Uma visibilidade intermediária é a chamada de protegida, ou *protected.* Nesse caso, a visibilidade inclui as classes filhas, além da própria classe em que o método, ou atributo, foi definido.

2. Análise das alternativas

A e B – Alternativas incorretas.
JUSTIFICATIVA. Um atributo privado não pode ser acessado por métodos de outras classes, mas pode acessar métodos (e atributos) que foram declarados apenas na própria classe na qual ele foi definido.

C – Alternativa incorreta.
JUSTIFICATIVA. Um membro público também é visível para as classes descendentes da classe na qual ele foi definido.

D – Alternativa incorreta.
JUSTIFICATIVA. Tanto os métodos definidos como protegidos quanto aqueles definidos como privados podem acessar os atributos privados definidos na própria classe.

E – Alternativa correta.
JUSTIFICATIVA. A definição apresentada na alternativa refere-se ao membro protegido.

3. Indicações bibliográficas

- FOWLER, M. *Padrões de arquitetura de aplicações corporativas*. Porto Alegre: Bookman, 2018.
- GAMMA, E. et al. *Padrões de projetos: soluções reutilizáveis de software orientados a objetos*. Porto Alegre: Bookman, 2000.
- LARMAN, C. *Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objetos e ao desenvolvimento iterativo*. 3. ed. Porto Alegre: Bookman, 2007.
- LAU, Y. *The art of objects: object-oriented design and architecture*. Upper Saddle River: Pearson, 2001.
- LIBERTY, J.; MACDONALD, *B. Learning C# 2005*. 2. ed. Sebastopol: O'Reilly, 2006.
- TROELSEN, A. *Pro C# 2005 and the .NET 2.0 Platform*. 3. ed. New York: Apress, 2005.
- TROELSEN, A.; JAPIKSE, P. *Pro C# 7: with .NET and .NET Core*. 8. ed. New York: Apress, 2017.

Questão 3

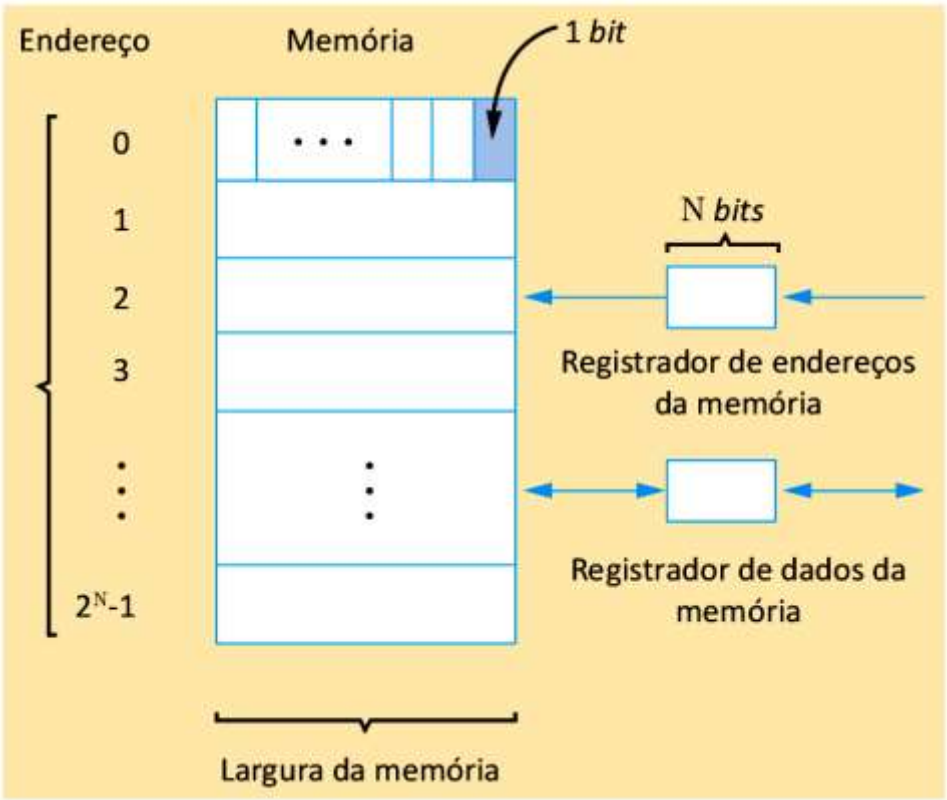
Questão 3.[3]

Leia o texto a seguir.

Em um computador, a memória é a unidade funcional que armazena e recupera operações e dados. Tipicamente, a memória de um computador usa uma técnica chamada acesso aleatório, que permite o acesso a qualquer uma de suas posições (células). As memórias de acesso aleatório são divididas em células de tamanho fixo, estando cada célula associada a um identificador numérico único chamado endereço. Todos os acessos à memória referem-se a um endereço específico e deve-se sempre buscar ou armazenar o conteúdo completo de uma célula, ou seja, a célula é a unidade mínima de acesso.

SCHNEIDER, G. M.; GERSTING, J. L. *An Invitation to computer science*. 6. ed. Boston: MA: Course Technology, Cengage Learning, 2009 (com adaptações).

A figura a seguir apresenta a estrutura de uma unidade de memória de acesso aleatório.



Considerando o funcionamento de uma memória de acesso aleatório, avalie as afirmativas.

I. Se a largura do registrador de endereços da memória for de 8 bits, o tamanho máximo dessa unidade de memória será de 256 células.

II. Se o registrador de dados da memória tiver 8 bits, será necessária mais do que uma operação para armazenar o valor inteiro 2024 nessa unidade de memória.

III. Se o registrador de dados da memória tiver 12 bits, é possível que a largura da memória seja de 8 bits.

[3]Questão 12 – Enade 2017.

É correto o que se afirma em

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

1. Introdução teórica

Arquitetura de computadores: memória RAM

Na computação, memória RAM é uma abreviação para *random access memory,* ou memória de acesso aleatório. O motivo desse nome peculiar está na ideia de que podemos acessar qualquer posição da memória em uma sequência arbitrária, e o tempo de acesso individual seria idealmente o mesmo. Na prática, o tempo de acesso não é absolutamente igual, podendo variar por vários fatores. Contudo, se compararmos uma memória RAM com um disco rígido, veremos uma diferença de comportamento muito grande. Em um disco rígido, o acesso a elementos que estão fisicamente próximos uns dos outros e em posições sequenciais é muito mais rápido que o tempo de acesso de elementos "espalhados" pelo disco. O fato de essa diferença ser muito menor em uma memória RAM levou ao seu nome.

De uma forma bem abstrata, podemos pensar que uma memória RAM funciona como uma tabela formada por duas colunas:

- na primeira coluna, temos o endereço de cada célula de memória;
- na segunda coluna, temos o conteúdo associado a esse endereço.

Não devemos confundir a quantidade de bits utilizada para especificar determinada posição de memória (o seu endereço) com a quantidade armazenada nessa posição. Esses valores podem ser bem diferentes. O primeiro está relacionado à quantidade de posições de memória disponíveis. O segundo está relacionado à quantidade de bits armazenada em cada célula de memória, o que é normalmente chamado de largura da memória.

O computador acessa as células de memória utilizando dois registradores:

- o registrador de endereços de memória, que identifica o endereço da célula que se quer acessar;
- o registrador de dados de memória, que contém o valor associado.

Contudo, é possível que o tamanho do registrador de dados de memória e a largura da memória não sejam iguais. Isso acontece porque um dado pode ocupar várias células de memória. Por exemplo, as células de memória podem conter (ou ter uma largura de) 8 bits. Mas o registrador de dados de memória pode ter 16 bits, sendo que cada dado vai ocupar 2 células de memória.

É natural que o tamanho do registrador de dados de memória seja um múltiplo da largura de memória, visto que isso simplifica o processo de manipulação de memória e evita desperdícios.

2. Análise das afirmativas

I – Afirmativa correta.
JUSTIFICATIVA. Considerando 8 bits para codificar os endereços da memória, sabemos que é possível obter 2^8=256 combinações possíveis, que correspondem às 256 células de memória.

II – Afirmativa correta.
JUSTIFICATIVA. Se considerarmos apenas uma representação binária de um número inteiro, sem sinal, sabemos que 2024 na base 10 é igual a 11111101000 na base 2. Observe que, nesse caso, em uma representação simples, precisamos de 11 bits para representar o número 2024, que é maior do que os 8 bits do registrador de dados da memória citados na afirmativa. Dessa forma, é necessária mais uma operação para armazenar esse número, pois não há bits suficientes em uma única operação. Outras representações, incluindo o sinal, vão necessitar de ainda mais bits para serem armazenados.

III – Afirmativa incorreta.
JUSTIFICATIVA. O registrador de dados da memória normalmente tem um tamanho que é um múltiplo da largura de memória, o que não é o caso da afirmativa.

Alternativa correta: C.

3. Indicações bibliográficas

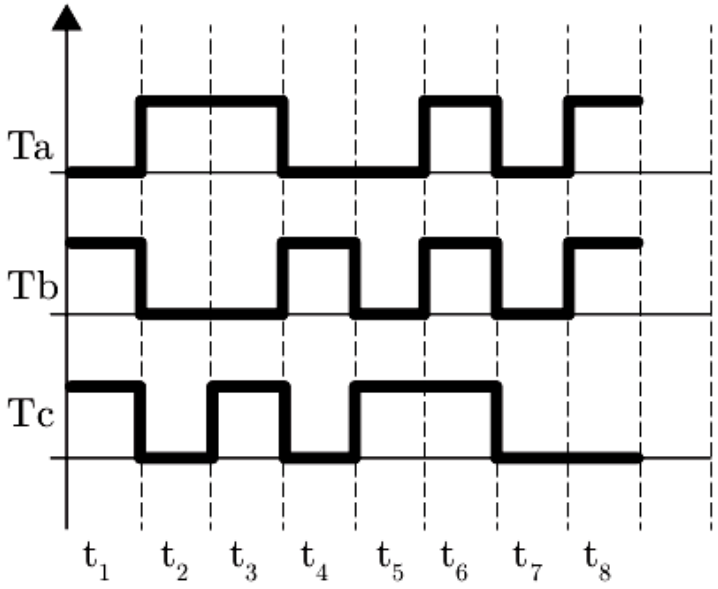
- HENNESSY, J. L.; PATTERSON, D. A. *Computer architecture: a quantitative approach*. 5. ed. Amsterdam: Elsevier, 2012.
- MONTEIRO, M. A. *Introdução à organização de computadores*. 5. ed. Rio de Janeiro: LTC, 2007.
- SCHNEIDER, G. M.; GERSTING, J. L. *An invitation to computer science*. 6. ed. Boston: Cengage Learning, 2009.
- STALLINGS, W. *Arquitetura e organização de computadores*. 5. ed. São Paulo: Pearson, 2002.
- TANENBAUM, A. S.; AUSTIN. T. *Organização estruturada de computadores*. 6. ed. São Paulo: Pearson, 2013.

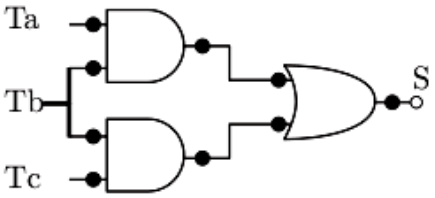
Questão 4

Questão 4.[4]

Os sistemas de refrigeração de piscinas de combustível em usinas nucleares evitam que a temperatura desses tanques exceda o limite de segurança. O circuito representado na figura a seguir atende aos requisitos necessários para o controle da ativação do sistema de resfriamento quando a temperatura está próxima de seu ponto crítico.

O diagrama de tempo ilustrado na figura apresenta uma amostra das temperaturas lidas desde o momento t_1 ao t_8. Os sinais de entrada Ta, Tb e Tc são de termômetros que medem a temperatura da piscina em diferentes pontos ao longo do dia e S é o terminal de acionamento do sistema.





Nesse contexto, assinale a opção em que são apresentados os momentos em que o sistema foi acionado.

A. t_1, t_4 e t_8.

B. t_1, t_6 e t_8.

C. t_2, t_4 e t_6.

D. t_2, t_6 e t_8.

E. t_3, t_5 e t_7.

[4]Questão 13 – Enade 2017.

1. Introdução teórica

Sistemas digitais: lógica combinatória

Os computadores que utilizamos na atualidade são chamados de máquinas digitais. Todo o seu processamento é baseado em operações lógicas simples feitas em um sistema binário.

O computador moderno é um exemplo do que é possível fazer utilizando circuitos digitais. Contudo, é importante termos em mente que nem todos os circuitos digitais são genéricos e flexíveis como um computador. Alguns são muito simples, realizam funções específicas. Além disso, um computador é construído pela integração de diversos circuitos digitais com diferentes finalidades.

Esses circuitos digitais mais especializados são construídos, em sua maioria, a partir de elementos ainda mais simples, as portas lógicas. Essas portas costumam implementar funções lógicas (booleanas) muito simples, como a inversão (porta lógica NOT), a conjunção (a porta lógica E, no inglês AND) e a disjunção (porta lógica OU, no inglês OR), entre outras operações similares.

Um dos tipos de circuitos digitais mais rudimentares que podemos construir são aqueles nos quais a saída é uma função booleana direta das entradas, sem a existência de nenhum tipo de dependência (memória) dos estados ou das entradas anteriores. Isso é chamado de lógica combinatória ou combinacional. Se existir uma dependência de estados anteriores, temos uma situação chamada de lógica sequencial.

2. Análise das alternativas

O circuito da figura pode ser expresso pela seguinte expressão booleana:

$$S = (T_aT_b)+(T_bT_c)$$

Assim, a saída S estará ativada apenas quando Ta e Tb forem iguais a 1 (ao mesmo tempo) ou quando Tb e Tc forem iguais a 1 (também ao mesmo tempo).

A – Alternativa incorreta.

JUSTIFICATIVA. Em t_4, $S=(T_aT_b)+(T_bT_c)=0+0=0$, o que significa que o sistema não foi acionado.

B – Alternativa correta.

JUSTIFICATIVA. Da figura, podemos observar que, em t_1, $S=(T_aT_b)+(T_bT_c)=0+1=1$. Em t_6, $S=(T_aT_b)+(T_bT_c)=1+1=1$. Em t_8, $S=(T_aT_b)+(T_bT_c)=1+0=1$. Nos demais momentos, $S=0$.

C e D – Alternativas incorretas.

JUSTIFICATIVA. Da figura, podemos observar que, em t_2, $S=(T_aT_b)+(T_bT_c)=0+0=0$, e o sistema não é acionado.

E – Alternativa incorreta.

JUSTIFICATIVA. Observe que, em t_3, $S=(T_aT_b)+(T_bT_c)=0+0=0$, e o sistema não é acionado.

3. Indicações bibliográficas

- CAPUANO, F. G. *Sistemas digitais: circuitos combinacionais e sequenciais*. São Paulo: Érica, 2008.
- FLOYD, T. L. *Sistemas digitais: fundamentos e aplicações*. 9. ed. Porto Alegre: Bookman, 2007.
- TOCCI, R. J.; WIDMER, N. S. *Sistemas digitais: princípios e aplicações.* 7. ed. Rio de Janeiro: LTC, 2000.

Questão 5

Questão 5.[5]

Leia o texto a seguir.

Uma das técnicas de ataques em ambientes virtuais é denominada "homem no meio" (man in the middle), cujo objetivo é associar o endereço MAC do intruso ao endereço IP de um outro nó da rede – nesse caso, o ponto de acesso (Access Point - AP) wi-fi da rede. Como o AP é o gateway padrão dessa sub-rede sem fio, todo o tráfego originalmente direcionado ao ponto de acesso pode ser interceptado pelo intruso. Esse ataque explora deficiências conhecidas no projeto de segurança do IEEE 802.11 wi-fi.

COULOURIS, G. et al. *Sistemas distribuídos: conceitos e projeto*. Porto Alegre: Bookman, 2013 (com adaptações).

Considerando um ataque virtual pela técnica "homem no meio", por meio de Address Resolution Protocol (ARP) spoofing, avalie as afirmativas.

I. O problema do compartilhamento de chave presente no projeto de segurança do AP pode ser resolvido com a utilização de um protocolo baseado em chave pública para negociar chaves individuais, como é feito no Transport Layer Security(TLS)/ Secure Sockets Layer (SSL).

II. O problema do desvio de tráfego causado pelo ataque de homem no meio pode ser evitado com a configuração de um firewall nos pontos de acesso que filtram tráfego entre clientes de uma mesma sub-rede.

III. O problema da falta de autenticação dos pontos de acesso sem fio pode ser contornado, obrigando-se o ponto de acesso a fornecer um certificado que possa ser autenticado pelo uso de uma chave pública obtida de terceiros.

IV. A vulnerabilidade das chaves de 40 bits ou 64 bits a ataques de força bruta pode ser evitada utilizando-se um AP que permita chaves de 128 bits e limitando-se o tráfego a dispositivos compatíveis com chaves de 128 bits.

É correto apenas o que se afirma em

A. I e II.

B. I e III.

C. II e IV.

D. I, III e IV.

E. II, III e IV.

[5]Questão 15 – Enade 2017.

Introdução teórica

Segurança de redes: ataque man in the middle

O ataque man in the middle, traduzido frequentemente como "homem no meio" (ainda que uma possível tradução mais adequada seria "homem intermediário"), envolve a estratégia em que um terceiro (um atacante) intercepta a comunicação entre duas máquinas e se faz passar por uma delas.

Uma situação típica na qual esse tipo de ataque pode ser utilizado é pelo protocolo ARP (Address Resolution Protocol). Esse protocolo foi desenvolvido para resolver um problema de mapeamento entre a camada de rede e a camada de enlace.

Em uma rede que utiliza a pilha de protocolos TCP/IP, sabemos que a camada de rede (no caso, chamada de camada internet) associa um endereço lógico, chamado de endereço IP, para cada máquina. Mas a pilha especifica também uma camada mais inferior, a camada de enlace de dados.

As máquinas têm um endereço físico, que, no caso de uma rede que utiliza tecnologia Ethernet: trata-se do endereço MAC (media access control). Esse endereço é utilizado em uma camada diferente do endereço IP, sendo, portanto, um número distinto. Contudo, é necessário que exista uma correspondência entre esses endereços, a fim de que a pilha de protocolos possa funcionar.

Nesse sentido, o protocolo ARP serve para fazer o mapeamento entre o endereço IP (que é também chamado de endereço lógico) e o endereço MAC (chamado de endereço físico). O endereço MAC é chamado de físico, pois está direta e permanentemente associado a um dispositivo físico (uma placa de rede, por exemplo, sendo que esse cenário não é obrigatório). O endereço IP, por sua vez, é associado a outra camada e pode ser facilmente alterado.

A natureza do protocolo ARP e a forma como foi feita a sua definição permitem a existência de ataques chamados de ARP spoofing, ou falsificação de ARP. Nesse caso, um atacante tenta manipular a associação do endereço lógico com o endereço físico. O fato de o protocolo ARP ser um protocolo sem estado (no inglês, stateless) facilita esse tipo de ataque, pois uma máquina que está tentando "enganar" outra pode enviar respostas falsas de requisição do endereço MAC para outras, tentando manipular sua tabela de mapeamento entre endereços lógicos e físicos.

Com base em técnicas de ARP spoofing, é possível fazer ataques man in the middle. Nesse caso, o objetivo é "enganar" duas ou mais máquinas que estão se comunicando em uma rede, tentando capturar ou alterar o tráfego entre essas máquinas com o mapeamento feito pelo protocolo ARP. O problema torna-se consideravelmente mais grave em uma rede sem fio com o padrão IEEE 802.11, em que o ponto de acesso (AP ou access point) também é o gateway da rede. O atacante pode tentar se fazer passar pelo gateway, interceptando a comunicação de várias máquinas.

2. Análise das afirmativas

I e III – Afirmativas corretas.
JUSTIFICATIVA. A utilização de protocolos TSL/SSL requer que o servidor ofereça um certificado que pode ser checado pelo cliente e que "prove" a sua identidade, tornando ataques que tentam "enganar" a identidade de uma máquina muito mais difíceis e a rede mais segura. Contudo, é importante ter em mente que atacantes sempre podem tentar encontrar vulnerabilidade nas implementações desses protocolos.

II – Afirmativa incorreta.
JUSTIFICATIVA. A utilização de um firewall serve para aumentar a segurança entre a rede interna e a rede externa, mas não entre elementos dentro de uma rede local. Um ataque de ARP spoofing vai ser feito por algum atacante que tenha acesso ao segmento local da rede (LOCKHART, 2004, p.67), por isso está fora da ação do firewall.

IV – Afirmativa correta.
JUSTIFICATIVA. Chaves maiores e mais complexas tornam ataques por força bruta (aqueles nos quais o atacante tenta aleatoriamente ou sequencialmente encontrar a chave) muito mais longos. Limitar o uso de chaves evita que um atacante consiga utilizar esse tipo de ataque.

Alternativa correta: D.

3. Indicações bibliográficas

- COULOURIS, G. et al. *Sistemas distribuídos: conceitos e projeto*. 5. ed. Porto Alegre: Bookman, 2013.
- FOROUZAN, B. A.; FEGAN, S. C. *Data communications and networking*. 4. ed. Boston: McGrawHill, 2007.
- KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a Internet: uma abordagem top-down.* 6. ed. São Paulo: Pearson Education do Brasil, 2014.
- LOCKHART, A. *Network security hacks*. Sebastopol: O'Reilly, 2004.
- SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE. G. *Fundamentos de sistemas operacionais*. 4. ed. Rio de Janeiro: LTC, 2015.
- TANENBAUM, A. S.; WETHERALL, J. *Redes de computadores*. 5. ed. São Paulo: Pearson Education do Brasil, 2011.

Questão 6

Questão 6.[6]

Leia os textos a seguir.

A segurança da informação está diretamente relacionada com a proteção de um conjunto de informações, no sentido de preservar o valor que possuem para um indivíduo ou uma organização, tendo como propriedades básicas a confidencialidade, a integridade, a disponibilidade e a autenticidade.

LYRA, M. R. *Segurança e auditoria em sistemas de informação*. Rio de Janeiro: Ciência Moderna, 2008 (com adaptações).

A engenharia social é definida como o conjunto de técnicas utilizadas para reunir informações, explorando a tendência humana a ignorar os sistemas de segurança. Os ataques de engenharia social implicam interação com outros indivíduos, o que evidencia o aspecto psicológico da engenharia social.

MITNICK, K. D.; SIMON, W. L. *The art of deception: controlling the human element of security*. New York: Wiley, 2001 (com adaptações).

A ética normativa é o "certo" e o "errado" do comportamento social interpretado. A principal diferença entre essas duas perspectivas é a forma como um dilema moral é abordado, e não necessariamente as consequências disso.

Disponível em <http://www.ethicmorals.com>. Acesso em 18 jul. 2017 (com adaptações).

Em relação à segurança da informação, avalie as afirmativas.

I. As empresas sempre estão vulneráveis, pois o fator humano é o elo mais fraco da segurança da informação.

II. A segurança da informação não é um produto, e, sim, um processo.

III. A ética profissional é um importante fator a ser considerado na segurança da informação.

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.

1. Introdução teórica

Segurança da informação

A segurança da informação envolve diferentes aspectos, alguns de natureza técnica, como a configuração adequada de um servidor, outros de natureza psicológica

[6]Questão 16 – Enade 2017.

(comportamental), e até mesmo aspectos sociais e culturais. Ainda que o objetivo final seja o mesmo (preservar as informações de uma empresa, de uma corporação ou de um indivíduo), esses diferentes aspectos influenciam a forma como a segurança da informação será garantida.

Os aspectos técnicos e tecnológicos da segurança da informação são os mais óbvios. Existem tecnologias que são inerentemente mais seguras do que outras. Por exemplo, utilizar protocolos de criptografia para a comunicação é mais seguro do que usar um sistema que não utiliza nenhum tipo de criptografia. Alguns algoritmos de criptografia são mais seguros do que outros. Alguns produtos podem ter mais vulnerabilidades do que outros, e algumas dessas vulnerabilidades podem ser mais graves. Assim, a escolha de determinada tecnologia tem impacto bastante claro na garantia da segurança da informação.

Contudo, até um produto tecnicamente sofisticado, projetado com a incorporação de técnicas adequadas de segurança, pode ser configurado ou utilizado de forma errada. Se os profissionais da área de tecnologia que trabalham em uma empresa não souberem corretamente como utilizar as tecnologias disponíveis, mesmo utilizando os produtos mais avançados, o problema da segurança continua presente, apenas de outra forma.

Mas, além dessas questões focadas na tecnologia, também existem questões organizacionais. Empresas podem ter processos de funcionamento que estimulem ou mesmo induzam problemas de segurança. Por exemplo, empresas com estruturas organizacionais ambíguas, em que as responsabilidades por uma área não são claramente definidas, podem ter problemas inerentes de segurança da informação. Uma vez que não é claro quem é responsável por uma área, as informações podem ser compartilhadas por um grupo demasiadamente grande de funcionários, mesmo por aqueles que não deveriam ter acesso a determinado conteúdo. Situações como o treinamento adequado dos funcionários e a pressão para resultados imediatistas podem levar a sérios comprometimentos da segurança, especialmente no médio prazo e no longo prazo.

Como dissemos, outro ponto que também afeta a segurança está no âmbito psicológico. Mesmo em uma empresa em que os aspectos técnicos, tecnológicos e organizacionais sejam adequados, os aspectos psicológicos não podem ser desconsiderados. Qualquer organização que esteja comprometida com a segurança das suas informações precisa levar em conta esses componentes nas suas decisões.

Aspectos sociais e culturais, ainda que relacionados aos aspectos psicológicos, não são exatamente os mesmos. Em algumas culturas, a preocupação com a segurança pode ter

um caráter bastante diferente. A cultura local pode influenciar o funcionamento de uma empresa, tanto de forma positiva quanto de forma negativa.

Finalmente, a questão ética permeia todos os níveis de uma empresa ou organização, envolvendo desde a natureza das decisões estratégicas dos altos gestores até as ações de um funcionário mais humilde. A ética também se estende para fora de uma organização, compreendendo a interação entre ela e seus fornecedores, seus clientes e a sociedade em geral. Ainda que esse fator seja muito mais difícil de ser diretamente trabalhado, não devemos nos esquecer do impacto fundamental que ele tem na segurança da informação, dado que muitos incidentes são motivados por problemas de natureza ética.

2. Análise das afirmativas

I – Afirmativa correta.

JUSTIFICATIVA. Mesmo sistemas tecnicamente seguros podem ser violados por erros ou por ações humanas. Ainda que nem sempre esses erros sejam propositais, podendo ocorrer em virtude de limitações de conhecimento ou de ingenuidade, o aspecto humano tende a ser muito vulnerável. Mesmo problemas de aparente natureza tecnológica, como uma falha em um programa, podem ter a sua origem em um erro humano. Esse erro pode ter sido originado pelo programador, no seu projeto ou na comunicação dos seus requisitos.

II – Afirmativa correta.

JUSTIFICATIVA. Ainda que a segurança da informação certamente envolva o uso de muitos produtos (e requeira escolhas adequadas), apenas a existência e o uso desses produtos não garantem a segurança da informação de uma organização. Os fatores humanos são fundamentais para o sucesso ou para o fracasso da segurança da informação: por isso, tais fatores devem ser encarados como partes de um processo. Vale notar que é muito comum vermos empresas tentando vender produtos como sinônimo de segurança da informação, o que é equivocado. Esses produtos podem auxiliar ou dificultar os processos relacionados à segurança, mas eles não podem, isoladamente, garantir a segurança da informação, especialmente se não estiverem inseridos em um contexto adequado.

III – Afirmativa correta.

JUSTIFICATIVA. A ética profissional é fundamental para a segurança da informação, pois não é possível criar produtos ou processos que sejam completamente imunes aos fatores humanos.

Alternativa correta: E.

3. Indicações bibliográficas

- FONTES, E. *Segurança da informação: o usuário faz a diferença.* São Paulo: Saraiva, 2006.
- MENEZES, J. das C. *Gestão da segurança da informação*. Leme: Mizuno, 2006.
- SÊMOLA, M. *Gestão da segurança da informação: uma visão executiva*. Rio de Janeiro: Elsevier Brasil, 2014.

Questão 7

Questão 7.[7]

Leia o texto a seguir.

Grupos de cientistas e grandes corporações de todo o mundo têm buscado desenvolver sistemas computacionais inteligentes capazes de ajudar as pessoas a aprenderem. As possibilidades, os efeitos e as implicações éticas da aplicação da chamada Inteligência Artificial (IA) na educação são temas que vêm ganhando espaço nos debates na área de tecnologia educacional em todo o mundo.

Disponível em <http://www.revistaeducacao.com.br>. Acesso em 26 set. 2017 (com adaptações).

A respeito da adoção de técnicas de IA no processo educacional, avalie as asserções e a relação proposta entre elas.

I. Algoritmos de IA adaptativos podem auxiliar a experiência de aprendizado da pessoa de acordo com o seu perfil.

PORQUE

II. Os sistemas com algoritmos de IA adaptativos analisam respostas anteriores, buscando determinados padrões que possam indicar pontos de dificuldade ou facilidade da pessoa em relação a determinado assunto.

A respeito dessas asserções, assinale a opção correta.

A. As asserções I e II são proposições verdadeiras, e a asserção II justifica a I.
B. As asserções I e II são proposições verdadeiras, e a asserção II não justifica a I.
C. A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
D. A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
E. As asserções I e II são proposições falsas.

1. Introdução teórica

Inteligência artificial

Nos últimos anos, a área da Inteligência Artificial – IA (no inglês, *Artificial Intelligence* ou AI) tornou-se extremamente popular, aparecendo no cotidiano de noticiários e das discussões em todos os tipos de empresas. O aumento do poder computacional e a diminuição do custo associado ao hardware contribuíram para popularizar uma série de técnicas que, em um passado não muito distante, estavam disponíveis apenas em laboratórios de pesquisa avançados.

[7]Questão 17 – Enade 2017.

No entanto, um problema teórico que continua difícil de ser resolvido é como encontrar uma definição precisa de inteligência artificial, dado que isso representaria antes ter uma ideia bastante definida do próprio conceito de inteligência.

No texto introdutório clássico da área de inteligência artificial, Russel e Norvig (2013) discutem dois aspectos fundamentais que podem ser utilizados para a sua definição: do ponto de vista do pensamento e do ponto de vista do comportamento ou ação. Adicionalmente, esses dois pontos também podem ter duas perspectivas: uma comparação com o ser humano e uma comparação com o conceito de racionalidade. Assim, a inteligência artificial poderia ser conceituada de diversas formas: do ponto de vista de se projetar um sistema que simule o pensamento de um ser humano, ou que se comporte como um ser humano; do ponto de vista de um sistema que simule um pensamento racional ou do ponto de vista de um sistema que se comporte de forma racional.

Devido à explosão de pesquisas feitas nessa área nos últimos 30 anos, diversas técnicas foram desenvolvidas para resolver uma ampla gama de problemas de naturezas bastante diferentes e em condições específicas. Isso levou a uma ampla gama de paradigmas diferentes, muitas vezes difíceis de serem classificados.

Uma das primeiras classificações da área envolvia a diferenciação dos paradigmas simbólicos versus o conexionista. A abordagem simbólica pode ser considerada a forma tradicional de trabalho da ciência da computação, em que o foco está em representar problemas por um conjunto de símbolos e identificar regras (algoritmos) que manipulam esses símbolos de acordo com a solução de um problema específico. É essencialmente o que fazemos quando escrevemos um programa convencional.

No caso da abordagem conexionista, a ideia está na construção de sistemas que simulam o funcionamento de uma rede, frequentemente uma rede de inspiração biológica, como uma rede neural. O termo muitas vezes é utilizado como sinônimo do uso de redes neurais artificiais.

Além disso, o uso de abordagens de inspiração biológica na computação tornou-se tão grande e difundido que levou ao surgimento de toda uma área em separado, uma área chamada de computação natural.

A computação natural apresenta uma série de elementos que se sobrepõem à área de inteligência artificial e outros elementos diferentes, de forma que é difícil fazer uma separação precisa entre essas duas áreas. Alguns autores costumam subdividir a computação natural de acordo com a forma como a inspiração biológica (ou natural) afeta o modelo computacional, como nos casos listados a seguir:

- o objetivo pode ser construir um algoritmo de inspiração biológica, mas não necessariamente para simular um fenômeno natural;
- o objetivo pode ser simular um fenômeno natural;
- materiais naturais podem ser diretamente utilizados no processo de computação (por exemplo, na situação em que um computador utiliza diretamente moléculas de DNA, diferentemente de um computador comum, como o que utilizamos no dia a dia).

Seguindo como uma subdivisão dos sistemas que partem de uma inspiração natural para o seu funcionamento, Castro (2006) cita ainda diversas abordagens, como algoritmos evolutivos, redes neurais artificiais e inteligência de enxame, entre outras. Note que essa divisão não é exclusiva: a inteligência artificial também pode ser definida em termos similares, e as redes neurais são tipicamente vistas como uma técnica da inteligência artificial.

Recentemente, uma subárea da inteligência artificial que ganhou grande visibilidade é a área de aprendizado de máquina (no inglês, *Machine Learning*, frequentemente abreviado para ML). Nesse caso, o objetivo dos algoritmos dessa área está na construção de programas que possam modificar e melhorar o seu funcionamento, de forma autônoma e baseada no seu uso, simulando uma forma de aprendizado por parte do computador (por isso o nome da área).

Uma aplicação particularmente importante e que vem crescendo muito é a utilização da inteligência artificial na educação. A ideia é construir algoritmos que ajudem o aluno a aprender, com base nas avaliações das suas respostas e, também, nas avaliações das respostas de vários outros alunos. Diversos algoritmos podem ser usados para essa abordagem, especialmente aqueles que se utilizam das técnicas de aprendizado de máquina.

2. Análise das asserções

I – Asserção verdadeira.

JUSTIFICATIVA. Um dos objetivos do uso da inteligência artificial na educação é justamente adaptar o processo de aprendizado ao aluno, apontando oportunidades para auxiliá-lo a superar dificuldades e identificando seus pontos fortes, de forma autônoma e personalizada.

II – Asserção verdadeira.

JUSTIFICATIVA. Um exemplo de utilização de algoritmos adaptativos de inteligência artificial na educação pode ser feito por meio de perguntas e da avaliação das respostas obtidas.

Conforme um aluno responde às perguntas, o sistema pode avaliar não apenas o seu índice de acerto, mas o tipo de erro cometido. Com base nas respostas anteriores do aluno e em um estudo prévio de como essas respostas estão relacionadas com um tipo de erro, o sistema pode identificar falhas de aprendizado e sugerir pontos que devem ser reforçados. Além disso, o sistema pode orientar a seleção de questões de acordo com essas informações, coma intenção de orientar o aluno no seu processo de aprendizado.

Relação entre as asserções. Observe que a asserção II envolve um exemplo de aplicação do uso dos algoritmos adaptativos da inteligência artificial com o objetivo dado na asserção I: ajudar os alunos no processo de aprendizado. Dessa forma, a asserção II justifica a I.

Alternativa correta: A.

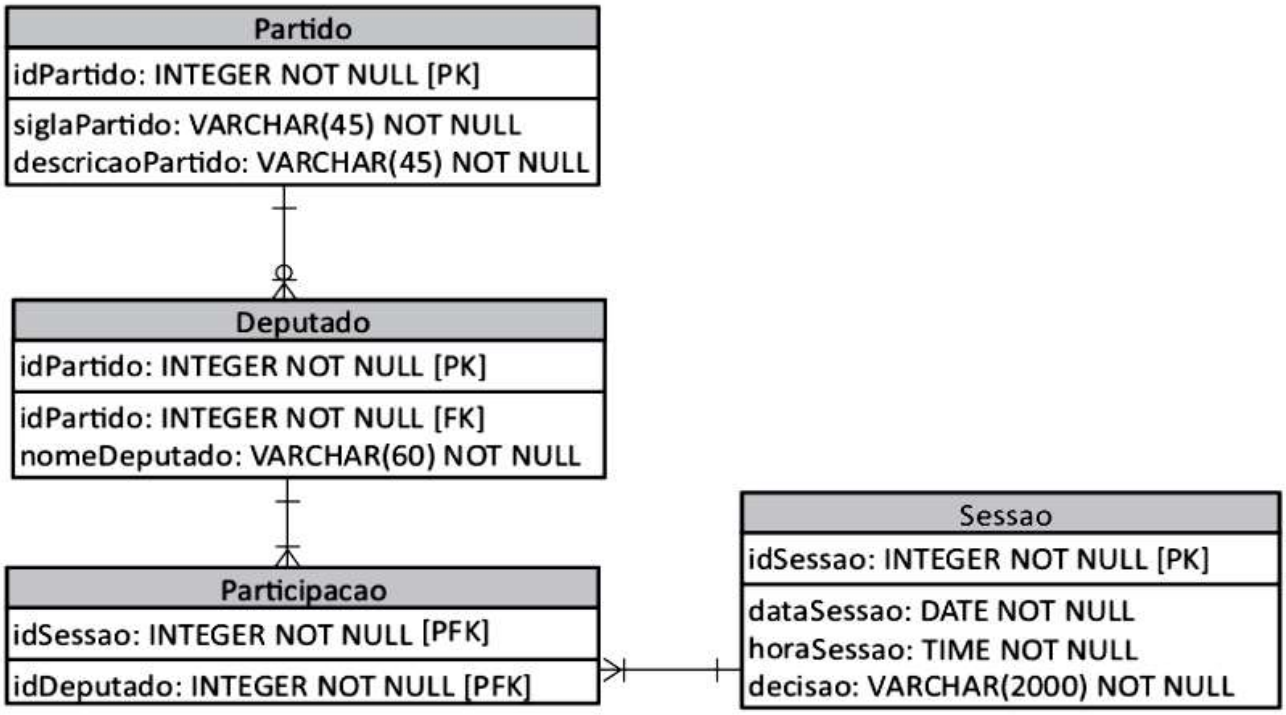
3. Indicações bibliográficas

- CASTRO, L. N. de. *Fundamentals of Natural Computing.* Boca Raton: Chapman & Hall/CRS, 2006.
- HERNANDEZ-ORALLO, J. et al. AI Paradigms and AI Safety: Mapping Artefacts and Techniques to Safety Issues. In: *European Conference on Artificial Intelligence - ECAI*, 24, 2020, Santiago de Compostela, Spain. Disponível em <https://ecai2020.eu/papers/1364_paper.pdf>. Acesso em 19 mar. 2026.
- LUGER, G. F. *Inteligência artificial.* 6. ed. São Paulo: Pearson Education do Brasil, 2013.
- MEDEIROS, L. F. de. *Inteligência artificial aplicada: uma abordagem introdutória.* Curitiba: InterSaberes, 2018.
- MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997.
- RUSSEL, S.; NORVIG, P. *Inteligência Artificial.* 3. ed. Rio de Janeiro: Elsevier, 2013.

Questão 8

Questão 8.[8]

Considere o diagrama Entidade-Relacionamento apresentado a seguir.



Qual código SQL exibe o nome de todos os deputados que compareceram a pelo menos uma sessão e as datas de cada sessão de que os deputados participaram?

A. SELECT Deputado.nomeDeputado, Sessao.dataSessao FROM Deputado, Participacao, Sessao WHERE Deputado.idDeputado=Participacao.idDeputado;
B. SELECT Deputado.nomeDeputado, Sessao.dataSessao FROM Deputado, Participacao, Sessao WHERE Deputado.idDeputado = Participacao. idDeputado OR Sessao.idSessao = Participacao.idSessao;
C. SELECT Deputado.nomeDeputado, Sessao.dataSessao FROM Deputado LEFT OUTER JOIN Participacao ON Deputado.idDeputado = Participacao.idDeputado LEFT OUTER JOIN Sessao ON Sessao.idSessao = Participacao.idSessao;
D. SELECT Deputado.nomeDeputado, Sessao.dataSessao FROM Deputado RIGHT OUTER JOIN Participacao ON Deputado.idDeputado = Participacao.idDeputado RIGHT OUTER JOIN Sessao ON Sessao.idSessao = Participacao.idSessao;
E. SELECT Deputado.nomeDeputado, Sessao.dataSessao FROM Deputado INNER JOIN Participacao ON Deputado.idDeputado = Participacao.idDeputado INNER JOIN Sessao ON Participacao.idSessao=Sessao.idSessao;

[8]Questão 19 – Enade 2017 (com adaptações).

1. Introdução teórica

Bancos de dados relacionais e consultas SQL

O uso de banco de dados relacionais é tradicionalmente feito por meio de uma linguagem específica, a linguagem SQL, no inglês *Structured Query Language.* Embora, nos últimos anos, venha ocorrendo a utilização de outras tecnologias que trabalham de forma diferente da trabalhada nos bancos de dados relacionais, sem a utilização da SQL (e sem a utilização do modelo relacional), a linguagem continua sendo muito popular e existe uma grande quantidade de sistemas que a utilizam.

A linguagem SQL costuma ser dividida em várias partes, às vezes chamadas de sublinguagens (ainda que a própria SQL costume ser chamada de uma sublinguagem em alguns textos). Essas partes estão relacionadas aos objetivos do conjunto de comandos oferecidos. Por exemplo, a parte da linguagem que lida com a criação, a modificação e a remoção de tabelas é chamada de DDL (*Data Definition Language*, ou linguagem de definição de dados). A parte da linguagem que foca na manipulação dos dados nas tabelas, como, por exemplo, a inserção de novos registros, sua remoção e sua alteração, é chamada de DML (*Data Manipulation Language* ou linguagem de manipulação de dados).

Para a consulta ou para a obtenção de dados do banco, utilizamos especialmente o comando `SELECT` da DML. Com esse comando, selecionamos registros vindos de uma tabela ou de várias tabelas. O resultado do comando `SELECT` também vai ser uma tabela, formada pelas colunas que especificamos na consulta, assim como os registros correspondentes (se existirem, ou `NULL`, caso contrário), respeitadas as condições impostas na consulta.

Um aspecto importante para termos em mente é que uma consulta SQL corresponde a diferentes operações da álgebra relacional: as operações de projeção, de produto cartesiano e de predicado de seleção.

Vamos nos lembrar de algumas dessas operações no que tange à álgebra relacional.

- A operação de projeção é representada pela letra grega pi maiúscula (Π) e permite que selecionemos quais atributos queremos retornar na relação resultante.
- O produto cartesiano, representado pelo sinal de multiplicação x, é utilizado com um significado similar ao de produto cartesiano da matemática: o produto cartesiano de duas tabelas t_1 x t_2 é o conjunto das combinações dessas duas tabelas.
- O predicado de seleção corresponde a uma condição e é associado com a operação de seleção, denotada pela letra grega sigma (σ). Por exemplo, suponha uma relação (ou

tabela) chamada de "cliente" e que possua o atributo saldo, um valor numérico de ponto flutuante. Se quisermos escrever a expressão da álgebra relacional que seleciona todos os clientes que possuem mais de R$100,00 em saldo, escrevemos $\sigma_{saldo>100,00}$(cliente).

Para criarmos uma consulta SQL típica, seguimos a seguinte forma geral:

```
SELECT <atributos> FROM <tabelas> WHERE <condição>
```

Na primeira parte da consulta, `SELECT <atributos>`, dizemos quais são os atributos que estamos interessados em obter. Podemos estar interessados em apenas um, em vários ou em todos os atributos das diversas tabelas envolvidas. Essa parte da consulta é similar à operação de projeção da álgebra relacional.

Na segunda parte da consulta, `FROM <tabelas>`, especificamos quais são as tabelas (ou relações) que estão envolvidas na consulta e fazemos o produto cartesiano dessas tabelas. A primeira parte da consulta nos permite filtrar os atributos resultantes desse produto cartesiano, especificado na segunda parte.

Finalmente, na parte `WHERE <condição>`, especificamos quais são as condições da seleção. Isso nos permite especificar valores ou faixa de valores para atributos que queremos selecionar, restringindo os registros que serão retornados. Em função desse comportamento, essa parte da consulta é entendida como o predicado de seleção da álgebra relacional.

Uma situação bastante comum no desenvolvimento de programas é aquela na qual queremos obter o resultado da correspondência entre registros vindos de diferentes tabelas. Por exemplo, um banco de dados pode ter uma tabela "cliente", com as colunas (ou os atributos) "ID" (sua chave primária) e uma coluna "nome". Uma segunda tabela, chamada de "telefone" tem duas colunas: uma coluna "numero_telefone" e outra coluna "ID_cliente", que é uma chave estrangeira para a tabela "cliente". Queremos identificar todos os clientes que possuam ao menos um telefone cadastrado e listar os seus telefones respectivos.

Para isso, podemos fazer uso de uma junção interna (`INNER JOIN`). Se simplesmente fizéssemos o produto cartesiano dessas duas tabelas, teríamos combinações de clientes e telefones que não fariam sentido. Contudo, ao fazermos uma junção interna com a condição que, nos registros retornados, o ID do cliente na tabela "nome" seja o mesmo que o ID_cliente da tabela "telefone", garantimos apenas combinações de nomes e telefones adequados. Na SQL, essa consulta é escrita como:

```
SELECT nome, numero_telefone
FROM cliente
INNER JOIN telefone ON cliente.ID = telefone.ID_cliente
```

O resultado dessa consulta retorna todos os registros da tabela "cliente" que possuam um registro associado na tabela "telefone". Clientes que não tenham telefones associados ou telefones que não tenham clientes associados não serão retornados. É interessante observar que, no modelo simplificado aqui proposto, um cliente pode ter zero, um ou mais telefones associados. Se tivéssemos dois clientes com o mesmo telefone, seria necessário ter dois registros em separado na tabela telefone, com ID_cliente diferentes, mas com o mesmo número de telefone cadastrado.

2. Análise das alternativas

A – Alternativa incorreta.
JUSTIFICATIVA. A cláusula `WHERE` não está completa, pois não envolve a tabela "Sessao", não havendo um `INNER JOIN` dessa tabela com a tabela "Participacao".

B – Alternativa incorreta.
JUSTIFICATIVA. A cláusula `WHERE` está incorreta para o problema proposto, pois está sendo feito um `OR` entre as condições da consulta. De acordo com o enunciado, queremos identificar todas as sessões das quais um deputado participou, com suas datas correspondentes. Para isso, devemos identificar registros das tabelas "Deputado", "Participacao" e "Sessao" que estejam associados pelas suas chaves estrangeiras.

C – Alternativa incorreta.
JUSTIFICATIVA. O uso de um `LEFT OUTER JOIN` como indicado na consulta da alternativa não retorna os registros desejados de acordo com o enunciado.

D – Alternativa incorreta.
JUSTIFICATIVA. O uso de um `RIGHT OUTER JOIN` como indicado na consulta da alternativa não retorna os registros desejados de acordo com o enunciado.

E – Alternativa correta.

JUSTIFICATIVA. A consulta seleciona corretamente as colunas "nomeDeputado" da tabela "Deputado" e "dataSessao" da tabela "Sessao". Além disso, a consulta utiliza corretamente a junção interna (`INNER JOIN`), retornando os registros da tabela "Deputado" que possuam registros associados na tabela "Participacao", e, adicionalmente, os registros da tabela "Sessao" associados a esses mesmos registros, pela chave estrangeira "idSessao" da tabela "Participacao".

3. Indicações bibliográficas

- DATE, C. J. *Introdução a sistemas de banco de dados.* 8. ed. Rio de Janeiro: Elsevier, 2003.
- MACHADO, F. N. R. *Banco de dados: projeto e implementação.* 4. ed. São Paulo: Érica, 2020.
- SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Sistemas de banco de dados.* 5. ed. Rio de Janeiro: Elsevier, 2006.

Questão 9

Questão 9.[9]

Em redes de computadores, a camada de transporte é responsável pela transferência de dados entre máquinas de origem e destino. Dois protocolos tradicionais para essa camada são o Transmission Control Protocol (TCP) e o User Datagram Protocol (UDP). Diferentemente do UDP, o TCP é orientado à conexão. Com relação a esses protocolos, avalie as afirmativas.

I. O UDP é mais eficiente que o TCP quando o tempo de envio de pacotes é fundamental.

II. O TCP é o mais utilizado em jogos on-line de ação para a apresentação gráfica.

III. O TCP é mais eficiente que o UDP quando a confiabilidade de entrega de dados é fundamental.

É correto o que se afirma em

A. II, apenas.

B. III, apenas.

C. I e II, apenas.

D. I e III, apenas.

E. I, II e III.

1. Introdução teórica

Redes de computadores: protocolos TCP e UDP

Em casos de aplicações que utilizam redes TCP/IP de forma intensa, um desenvolvedor pode ter que escolher qual será o protocolo da camada de transporte que vai ser utilizado em uma aplicação: TCP ou UDP. Vamos explorar alguns aspectos dessa escolha. Adicionalmente, é importante ressaltar a diferença que existe entre o termo "pilha de protocolos TCP/IP" e o protocolo TCP ou UDP especificamente.

A pilha de protocolos TCP/IP deve ser vista como um modelo em camadas, de uma forma similar (mas não igual) ao modelo OSI. Cada camada apresenta uma função e um conjunto de protocolos associados. O número de camadas do modelo de referência TCP/IP pode variar um pouco na literatura: alguns descrevem esse protocolo como tendo cinco camadas (aplicação, transporte, rede, enlace e física), enquanto outros descrevem apenas 4

[9]Questão 20 – Enade 2017.

(aplicação, transporte, internet e enlace). É possível encontrar uma correspondência entre algumas camadas do modelo OSI (outro modelo também utilizado na área de redes) e do modelo TCP/IP, ainda que esse último tenha menos camadas, agrupando algumas funcionalidades em uma mesma camada ou simplesmente ignorando camadas cujas capacidades não foram consideradas essenciais para o modelo, como a camada de sessão ou a camada de apresentação.

Cada uma das camadas é composta por vários protocolos. A camada de aplicação apresenta protocolos ligados a uma aplicação específica (por isso o seu nome). Por exemplo, quando acessamos uma página da internet via um endereço começando com "http", utilizamos o protocolo HTTP (que pertence à camada de aplicação). Esse protocolo é utilizado na comunicação entre os clientes (frequentemente navegadores) e os servidores web. Outro protocolo similar, no qual foi adicionada uma comunicação criptografada, é o protocolo HTTPS. Existem muitos outros protocolos na camada de aplicação, como o FTP, empregado na transferência de arquivos, e o NTP, usado na sincronização dos relógios de computadores, entre outros.

Abaixo da camada de aplicação, temos a camada de transporte. Assim como nos demais casos, existem vários protocolos disponíveis: o protocolo TCP, o protocolo UDP e o SCTP, por exemplo. A escolha do protocolo do transporte é uma etapa importante no projeto de um programa, que vai ter um impacto fundamental no desempenho de uma aplicação.

Uma das principais diferenças entre os protocolos TCP e UDP está no fato de que o primeiro é orientado à conexão, ao passo que o último não é. Algumas aplicações não necessitam do estabelecimento de uma conexão entre um cliente e o servidor, sendo que essa etapa pode impactar negativamente no seu desempenho. Por exemplo, vamos considerar o caso do sistema de nomes de domínio (DNS – Domain Name System), que é utilizado a cada momento em que devemos encontrar o endereço IP de um domínio baseado no seu nome, sendo o sistema de uso mais intenso da Internet. Nesse sistema, utiliza-se o protocolo UDP por uma questão de performance não só da aplicação, mas da rede e da Internet de forma geral. Se tivéssemos que abrir uma conexão, fazer uma requisição e fechá-la logo em seguida, a penalidade de tempo devido seria considerável. Por esse motivo, emprega-se o protocolo UDP, evitando essa sobrecarga.

Contudo, o protocolo TCP é mais indicado em situações nas quais devemos garantir a confiabilidade nas transferências de dados, incluindo o controle de fluxo. Se estivermos

interessados em transferir um arquivo de um computador para outro, a confiabilidade é fundamental. Todo o conteúdo original deve ser transferido de forma exata.

Há, entretanto, situações nas quais uma informação perde a sua utilidade após um certo tempo. Um exemplo típico ocorre em sistemas que implementam chamada de voz, similar à telefonia. Caso uma parte da conversa não seja transmitida em certo intervalo de tempo, é melhor simplesmente ignorar a informação. Se isso não for feito, e parte do áudio for entregue muito atrasado, o ouvinte pode perceber essa informação como uma espécie de ruído, o que piora a comunicação. Nesses casos, a utilização do protocolo UDP pode ser mais interessante, ainda que a escolha final do protocolo a ser usado dependa de outros fatores.

2. Análise das afirmativas

I – Afirmativa correta.
JUSTIFICATIVA. Nos casos em que o tempo de envio é fundamental (por exemplo, no caso de uma aplicação de comunicação de voz similar à da telefonia), a penalidade de tempo na abertura e no fechamento de conexões é desaconselhada, o que sugere o uso do protocolo UDP.

II – Afirmativa incorreta.
JUSTIFICATIVA. Na apresentação gráfica de um jogo que apresenta muita ação, ou seja, em que a tela sofre muitas alterações a cada segundo, o ideal é que o tempo de envio seja o menor possível (para evitar atrasos e travamentos na tela durante o jogo), o que indica o uso do protocolo UDP, e não do TCP, como dito na afirmativa.

III – Afirmativa correta.
JUSTIFICATIVA. O protocolo TCP foi projetado para incorporar confiabilidade na entrega de dados. Diversas das suas características, como o fato de ser orientado à conexão e ter controle de fluxo, sugerem a sua utilização em aplicações nas quais a confiabilidade de entrega é condição fundamental.

Alternativa correta: D.

3. Indicações bibliográficas

- FOROUZAN, B. A.; FEGAN, S. C. *Data communications and networking*. 4. ed. Boston: McGrawHill, 2007.
- KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a Internet: uma abordagem top-down.* 6. ed. São Paulo: Pearson Education do Brasil, 2014.
- NICBR. *O NTP.* Disponível em <https://ntp.br/ntp.php>. Acesso em 19 mar. 2026.
- ORACLE. *System administration guide: IP Services.* 2010. Disponível em <https://docs.oracle.com/cd/E23823_01/html/816-4554/ipov-6.html>. Acesso em 19 mar. 2026.
- TANENBAUM, A. S.; WETHERALL, J. *Redes de computadores*. 5. ed. São Paulo: Pearson Education do Brasil, 2011.

Questão 10

Questão 10.[10]

Considere o seguinte alfabeto:

$$\Sigma=\{(,),0,1,2,3,4,5,6,7,8,9,+,-\}.$$

Considere, ainda, uma linguagem L definida sobre esse alfabeto.

$$L = \{w|\ w \in \Sigma^*, \text{ para cada ocorrência de '(' em w, existe uma ocorrência de ')'}\}$$

Por exemplo, a cadeia x=(2+(3-4)) pertence a L, mas a cadeia y=(2+(3-4) não pertence a L.

Com relação à linguagem L, avalie as asserções e a relação proposta entre elas.

I. A linguagem L não pode ser considerada regular.

PORQUE

II. Autômatos finitos não têm mecanismos que permitam contar infinitamente o número de ocorrências de determinado símbolo em uma cadeia.

A respeito dessas asserções, assinale a opção correta.

A. As asserções I e II são proposições verdadeiras, e a asserção II justifica a I.
B. As asserções I e II são proposições verdadeiras, e a asserção II não justifica a I.
C. A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
D. A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
E. As asserções I e II são proposições falsas.

1. Introdução teórica

Linguagens formais: autômatos finitos e linguagens regulares

Uma das áreas mais importantes da teoria da computação é o estudo das linguagens formais. Nessa área, define-se linguagem como um conjunto de sequências de símbolos, estes últimos restritos a outro conjunto, chamado de alfabeto.

Essa definição costuma surpreender algumas pessoas, pois o termo "linguagem", especialmente na computação, está relacionado às linguagens de programação, como C, Java ou C#, e a ideia de que uma linguagem pode ser uma simples sequência de símbolos pode parecer um pouco surpreendente ou confusa. Por exemplo, podemos definir um

[10]Questão 23 – Enade 2017.

alfabeto composto de apenas dois símbolos, por exemplo, a letra 'a' e a letra 'b'. Depois, podemos definir uma linguagem composta das seguintes cadeias: ab, ba, abab, baab. Formalmente, diríamos que o alfabeto da linguagem Σ é {a,b} e que a linguagem é {ab, ba, abab, baab}.

Adicionalmente, é importante termos em mente que a definição de linguagem formal inclui linguagens como C e Java, mas é mais abrangente, abarcando outros conjuntos. Como outro exemplo de linguagem, poderíamos considerar a linguagem L de todas as sequências de bits que têm quantidade ímpar de uns e quantidade par de zeros. Nesse caso, o alfabeto Σ seria apenas o conjunto {0,1}, e poderíamos escrever:

$$L=\{w \mid w \in \Sigma^* \text{ e } w \text{ tem um número par de zeros e ímpar de uns}\}$$

A cadeia (sequência de símbolos) 11100 pertence à linguagem L, enquanto a cadeia 111000 não pertence à linguagem L.

No último exemplo, quando escrevemos Σ^*, estamos nos referindo ao fecho de Kleene, uma operação que pode ser feita sobre um conjunto. Essa operação corresponde a fazer todas as combinações possíveis de símbolos que pertencem a um conjunto, no caso, o alfabeto Σ. Isso inclui cadeias de todos os comprimentos, de zero até infinito. Assim, ao escrevermos $w \in \Sigma^*$, estamos dizendo que a cadeia w pertence ao conjunto Σ^*, que é o conjunto de todas as combinações possíveis de zeros e uns.

Mas observe que isso não quer dizer que w corresponde a todo o conjunto Σ^*. Existe outra condição: w deve ter quantidade par de zeros e quantidade ímpar de uns. Esse conjunto é um subconjunto de Σ^*, mas não é igual a Σ^*. Podemos notar isso facilmente, pois 11100 pertence à linguagem L e a Σ^*, mas 111000 não pertence à linguagem L. Além disso, ambas as cadeias pertencem a Σ^*.

Um avanço importante na teoria de linguagens foi a observação de que elas podem ser subdivididas em uma hierarquia, normalmente chamada de hierarquia de Chomsky, em homenagem ao linguista americano Noam Chomsky, que propôs o estudo de linguagens na década de 1950.

É interessante notarmos que linguagens formais podem ser estudadas sob dois pontos de vistas diferentes: em um deles, estamos interessados em como reconhecer uma linguagem. Dada uma cadeia (uma sequência de símbolos), queremos responder à pergunta: essa cadeia pertence a uma dada linguagem? Tal tarefa é feita por diversos tipos de máquinas ou autômatos. Aqui, a ideia de máquina é abstrata, não se referindo a uma máquina real específica.

Outro ponto de vista está relacionado a como podemos produzir uma cadeia. Dada uma linguagem e conhecidas as suas regras, queremos produzir cadeias que pertençam a essa linguagem específica. Essas regras, também chamadas de "regras de produção", fazem parte da gramática formal, associadas a uma linguagem específica.

Um avanço importante no estudo das linguagens formais foi que, ao colocarmos algum tipo de limite na máquina (ou autômato), como um limite de memória ou da forma como a memória é manipulada, também limitamos o conjunto de linguagens que podem ser reconhecidos por essa categoria de máquinas.

De forma similar, ao colocarmos restrições nas regras de produção de uma gramática, limitamos os tipos de linguagens que podem existir na gramática. Essas restrições também estão diretamente relacionadas às restrições colocadas sobre as máquinas que devem reconhecer uma linguagem.

As linguagens regulares são aquelas que podem ser reconhecidas por um autômato finito e que são geradas por gramáticas regulares. Costuma-se dizer que os autômatos finitos não têm memória, mas essa afirmação deve ser vista com cuidado: os autômatos finitos têm memória limitada, que corresponde aos seus estados. Como o número de estados de um autômato finito é finito, isso impõe um limite à sua memória. Essa limitação coloca restrições com relação às situações nas quais seria necessário contar a ocorrência de determinado símbolo, e nas quais não dispomos de nenhuma informação sobre qual é o limite máximo dessa contagem (esse não seria o caso se soubéssemos o limite de antemão, e, nesse caso, poderíamos pensar na utilização de um autômato finito).

É importante darmos muita atenção à ideia de contagem arbitrária apresentada anteriormente. Por exemplo, suponha um alfabeto composto de dois símbolos (por exemplo, zero e um) e uma linguagem que seja composta pelas cadeias que apresentam uma quantidade ímpar de zeros. Essa linguagem é regular, pois é possível construir um autômato finito que reconhece essa linguagem.

Alguém poderia concluir, erroneamente, que essa linguagem não pode ser regular, pois precisaríamos contar a quantidade de zeros e verificar se esse número é par ou ímpar. Contudo, isso não é necessário: sabemos que números pares e ímpares se alternam, e podemos ter apenas um estado correspondendo a um número par de zeros e outro correspondente a um número ímpar de zeros. Assim, mesmo sem saber o número exato de zeros, podemos saber se a quantidade total é par ou ímpar. Isso nos mostra que devemos ser cuidadosos ao interpretarmos o significado de quantidade finita de memória.

2. Análise das asserções

I – Asserção verdadeira.
JUSTIFICATIVA. A linguagem descrita é um exemplo clássico de uma linguagem que não é regular, o que pode ser provado utilizando o lema do bombeamento para linguagens regulares.

II – Asserção verdadeira.
JUSTIFICATIVA. Devido ao número finito de estados e à ausência de qualquer memória auxiliar, autômatos finitos não são capazes de "contar" e armazenar números com tamanho arbitrário.

Relação entre as asserções. Na linguagem descrita no enunciado, na qual devemos equiparar o número de parênteses abertos com o número de parênteses fechados, é necessário ter alguma forma de memorizar o número inicial de parênteses abertos. Como não sabemos quantos parênteses serão encontrados (a descrição da linguagem não apresenta nenhuma limitação), não podemos utilizar um autômato finito para fazer o reconhecimento dessa linguagem. Por esse motivo, podemos afirmar que a linguagem L do enunciado não é regular. Dessa forma, a asserção II justifica a I.

Alternativa correta: A.

3. Indicações bibliográficas

- AHO, A. V. et al. *Compiladores: princípios, técnicas e ferramentas.* 2. ed. São Paulo: Pearson Addison Wesley, 2007.
- BLAUTH, P. M. *Matemática discreta para computação e informática.* 4. ed. Porto Alegre: Bookman, 2013.
- COOPER, K. D.; TORCZON, L. *Construindo compiladores.* 2. ed. Rio de Janeiro: Campus-Elsevier, 2013.
- SIPSER, M. *Introdução à teoria da computação.* São Paulo: Cengage, 2005.

ÍNDICE REMISSIVO