



# ***UNIP***

---

## **ESTUDOS DISCIPLINARES** **Formação Específica**

**Profa. Me. Sandra Bozolan**

# 1) *JavaBeans*

---

- *JavaBeans* são componentes de *software* escritos em linguagem de programação Java. Segundo a especificação da Sun Microsystems os *JavaBeans* são componentes reutilizáveis de *software* que podem ser manipulados visualmente com a ajuda de uma ferramenta de desenvolvimento. Um *bean* também pode ser definido como uma classe Java que expõe propriedades, seguindo uma convenção de nomenclatura simples para os métodos *getter* e *setter*.
- O *JavaBean* é um Objeto Java que é serializável, possui um construtor nulo e permite acesso às suas propriedades por meio de métodos *getter* e *setter*.

# 1) *JavaBeans*

---

- O modelo de componentes *JavaBeans* foi desenvolvido para reutilizar componentes em outros componentes *JavaBeans* e *applets*, podendo rodar em um *web browser* usando uma *Java Virtual Machine (JVM)*.

# 1) *JavaBeans*

---

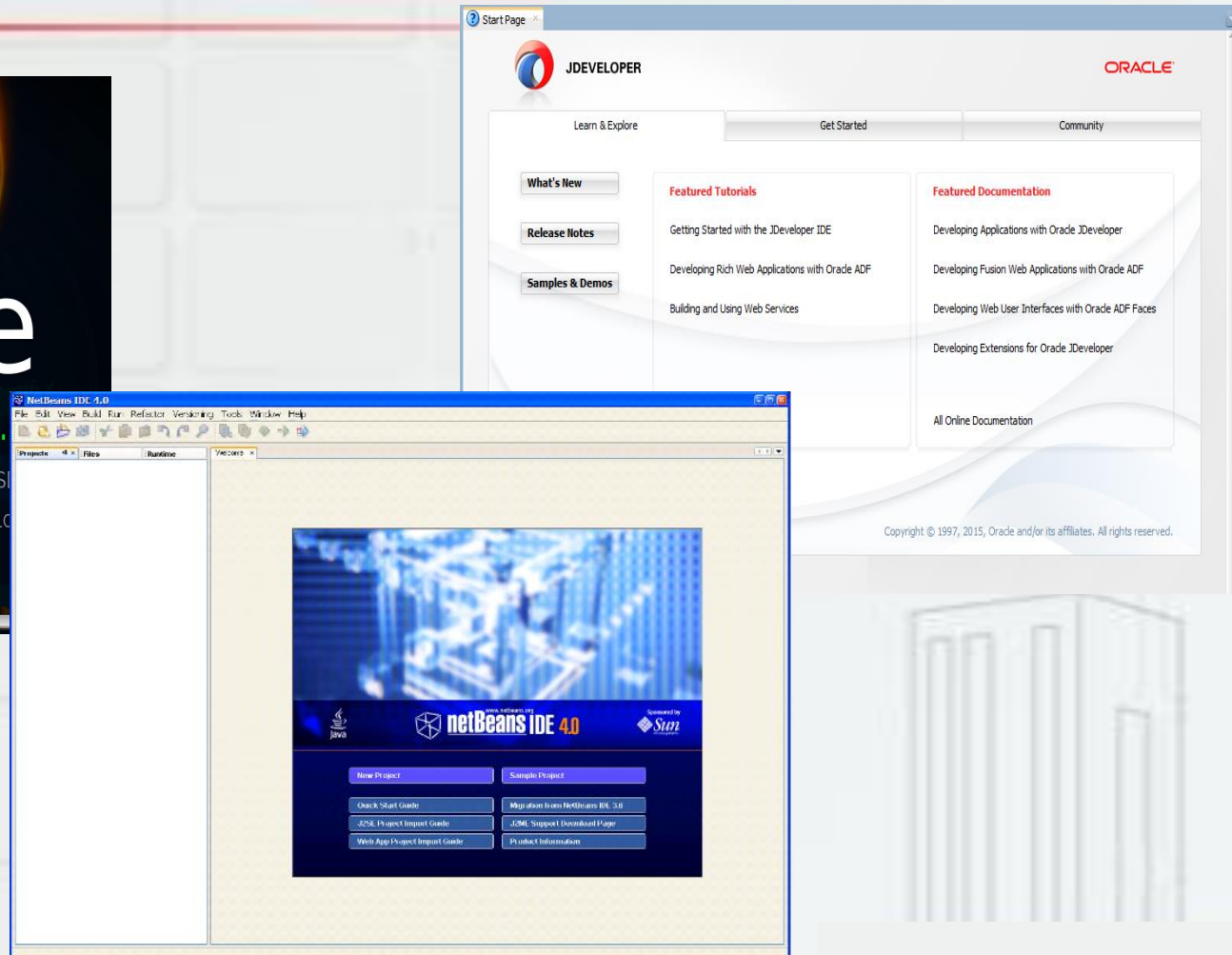
- A JVM é um programa que executa e carrega os aplicativos Java, converte o *bytecodes* (formato intermediário de código projetado para permitir que vários *hardwares* e *softwares* executem o mesmo código) em códigos de máquina executável e gerencia os aplicativos conforme são executados.
- Dessa forma, os programas em Java podem funcionar em qualquer plataforma de *software* ou *hardware* que tenha uma JVM instalada.

# 1) IDEs Java

---

- Eclipse, *Netbeans*, *Jdeveloper* são exemplos de ambientes integrados de desenvolvimento, chamados de IDE, que têm por objetivo facilitar o desenvolvimento de *softwares*, provendo maior produtividade e gerenciamento de projetos.
- A especificação *JavaBeans* foi criada para ser um padrão de desenvolvimento de componentes que possam ser facilmente usados por outros desenvolvedores em diferentes IDE.

# 1) Exemplos IDE Java (Eclipse, *NetBeans* e *Jdeveloper*)



Fontes: <https://bugs.eclipse.org/bugs/attachment.cgi?id=260996>  
[https://netbeans.org/competition/win-with-netbeans/get-started-with-nb\\_pt\\_BR.html](https://netbeans.org/competition/win-with-netbeans/get-started-with-nb_pt_BR.html)  
[https://docs.oracle.com/cd/E53569\\_01/tutorials/tut\\_rich\\_app\\_alta/](https://docs.oracle.com/cd/E53569_01/tutorials/tut_rich_app_alta/)

**UNIP**

# 1) IDEs Java

---

- Com relação ao tema, analise as asserções a seguir.
- Seja para o *Netbeans* ou para o Eclipse, é possível adquirir componentes de terceiros que facilitem a implementação do seu projeto.

# 1) IDEs Java

---

## Porque

- Como o código desses componentes está em linguagem intermediária, ou seja, independente da arquitetura de um computador real, só é necessário que a máquina virtual esteja instalada no computador onde o aplicativo será executado e a máquina virtual será a responsável pela interpretação do código para a linguagem de máquina do computador em execução.



# 1) IDEs Java

---

Acerca dessas asserções, assinale a alternativa correta.

- a) As duas asserções são proposições verdadeiras e a segunda é uma justificativa correta da primeira.
- b) As duas asserções são proposições verdadeiras e a segunda não é uma justificativa correta da primeira.
- c) A primeira asserção é uma proposição verdadeira e a segunda é uma proposição falsa.
- d) A primeira asserção é uma proposição falsa e a segunda é uma proposição verdadeira.
- e) Tanto a primeira quanto a segunda asserções são proposições falsas.

# 1) IDEs Java

Acerca dessas asserções, assinale a alternativa correta.

- a) As duas asserções são proposições verdadeiras e a segunda é uma justificativa correta da primeira.
- b) As duas asserções são proposições verdadeiras e a segunda não é uma justificativa correta da primeira.**
- c) A primeira asserção é uma proposição verdadeira e a segunda é uma proposição falsa.
- d) A primeira asserção é uma proposição falsa e a segunda é uma proposição verdadeira.
- e) Tanto a primeira quanto a segunda asserções são proposições falsas.

# 1) IDEs Java

---

- As duas asserções são proposições verdadeiras, mas a segunda não é uma justificativa correta da primeira porque é possível adquirir componentes de terceiros que facilitem a implementação de um projeto, com ganho de tempo e custo mais baixo do que ser desenvolvido internamente.

## 2) Exceções: *try* – *catch*

- Tratamento de erro e exceções, sempre que um método de alguma classe é passível de causar algum erro, podemos usar o método de tentativa, o *try*. Tudo que estiver dentro do bloco *try* será executado até que alguma exceção seja lançada, ou seja, até que algo dê errado.
- Quando uma exceção é lançada, ela sempre deve ser capturada. O trabalho de captura da exceção é executado pelo bloco *catch*.

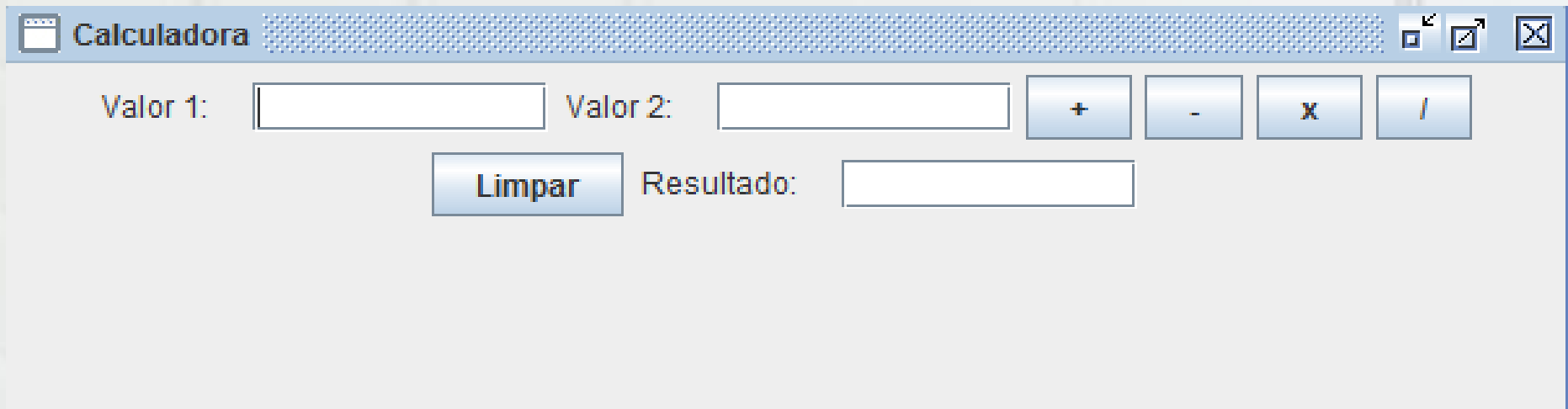
## 2) Exceções: *try* – *catch*

- Um bloco *try* pode possuir vários blocos de *catch*, dependendo do número de exceções que podem ser lançadas por uma classe ou método.

O bloco *catch* obtém o erro criando uma instância da exceção. Portanto, a sintaxe do bloco *try catch* é:

```
try
{ // código a ser executado }
catch (ClasseDeExceção instânciaDaExceção)
{ // tratamento da exceção }
```

## 2) Exceções: *try – catch* – Exemplo



A screenshot of a simple calculator application window titled "Calculadora". The window has a blue header bar with standard Windows window controls (minimize, maximize, close) on the right. The main area is light gray. It features two input fields labeled "Valor 1:" and "Valor 2:". To the right of these fields are four buttons for arithmetic operations: "+", "-", "x", and "/". Below the "Valor 1:" field is a button labeled "Limpar". To the right of the "Limpar" button is a label "Resultado:" followed by an empty output field.

Fonte: Própria (professor)

## 2) Exceções: *try – catch*

---

- Uma das utilidades da orientação a objetos do Java é a facilidade em tratar possíveis erros de execução (exceções). Sempre que um método de alguma classe é passível de cometer erro, podemos utilizar o método de tentativa *try/catch*.

## 2) Exceções: *try* – *catch*

Com relação às exceções, analise as afirmativas.

- I. O maior problema das exceções é que não existem métodos comuns entre todas as suas classes.
- II. O tratamento de exceções consiste em prever situações anormais que podem ocorrer e, a partir disso, buscar uma solução.
- III. Um bloco *try* pode ter vários blocos de *catch*, dependendo do número de exceções que podem ser lançadas por uma classe ou por um método.
- IV. Ao tentar conectá-las a um servidor inexistente, é impossível trabalhar com as classes de exceções do Java, pois o erro é físico.



## 2) Exceções: *try – catch*

---

É correto apenas o que afirma em:

- a) I e II.
- b) II e III.
- c) III e IV.
- d) I e III.
- e) II e IV.



**INTERVALO**



***UNIP***

## 2) Exceções: *try – catch*

---

É correto apenas o que afirma em:

a) I e II.

**b) II e III.**

c) III e IV.

d) I e III.

e) II e IV.

### 3) Concatenação de *Strings*

- Java tem uma versão do operador + para a concatenação de *strings* que concatena um *String* e um valor de outro tipo de dados (incluindo outra *String*); o resultado dessa operação é uma *String* nova (e normalmente mais longa).

Se assumirmos que soma contém o valor 117, a expressão é avaliada da seguinte maneira:

1. Java determina que os dois operandos do operador + (a string "A soma é." e o inteiro soma) são de tipos diferentes e um deles é um *String*.
2. *Java converte soma para um String.*
3. Java acrescenta a representação de soma sob a forma de *String* no fim de "A soma é. ", o que resulta na *String* "A soma é 117".

### 3) Concatenação de *Strings*

A linguagem JAVA2 EE apresenta operadores que realizam certas ações, como concatenação e aritmética de *strings*. Com base nos operadores utilizados nessa linguagem, a expressão `myJob += "Sr.";` é equivalente à:

- a) `myJob = myJob + 1.`
- b) `myJob = "Sr. + 1".`
- c) `myJob = "Sr." + 1.`
- d) `myJob = myJob + "Sr.."`
- e) `myJob = 1 + (myJob + Sr.).`

### 3) Concatenação de *Strings*

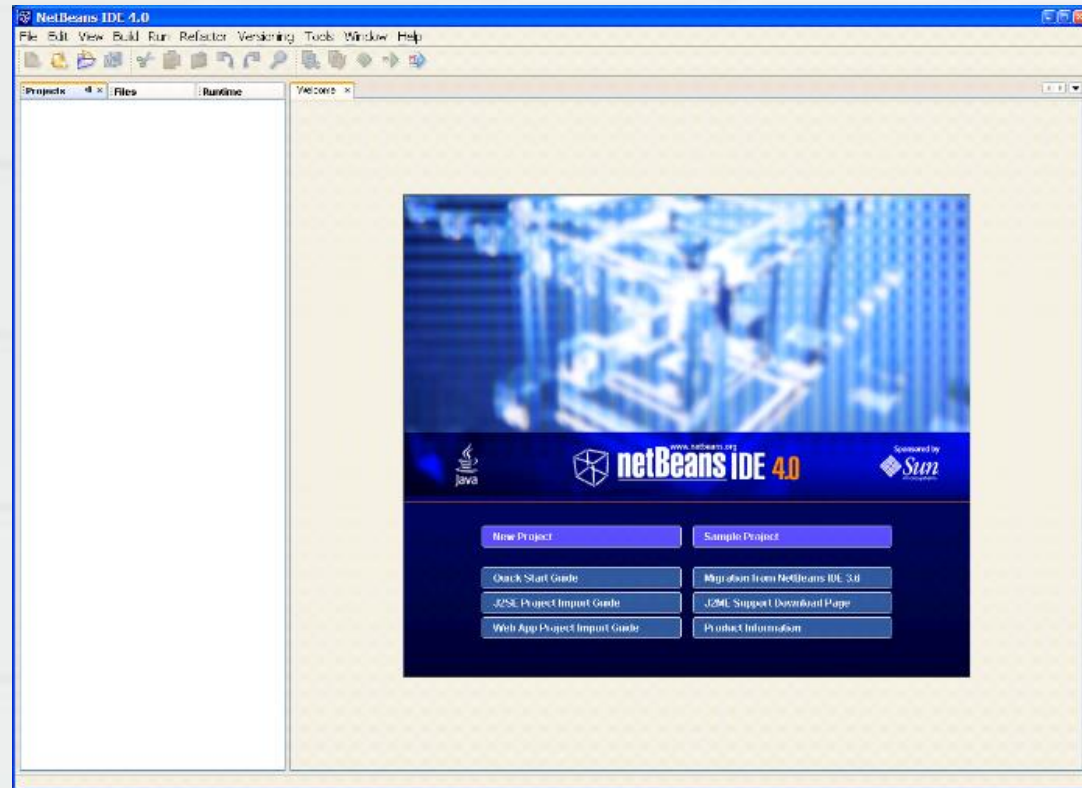
---

É correto apenas:

- a) `myJob = myJob + 1.`
- b) `myJob = "Sr. + 1".`
- c) `myJob = "Sr." + 1.`
- d) `myJob = myJob + "Sr."`
- e) `myJob = 1 + (myJob + Sr.).`

## 4) IDE – *NetBeans*

- O *NetBeans* IDE é um *software* que oferece suporte abrangente e de primeira classe para as tecnologias e melhorias de especificação Java mais recentes, antes de outros IDEs. É o primeiro IDE gratuito a oferecer suporte a JDK 8, JDK 7, Java EE 7.



Fonte: <https://netbeans.org/features/platform/index.html>

## 4) IDE – *NetBeans*

Com relação ao ambiente integrado de desenvolvimento *NetBeans*, assinale a alternativa correta:

- a) Uma de suas vantagens é o compartilhamento de biblioteca para outros usuários.
- b) Para criar arquivos Java não há necessidade de se criar um projeto anteriormente. Projetos e arquivos são conceitos tratados de forma distinta e não possuem relações.
- c) O *NetBeans* é um *software* que é utilizado especificamente em ambiente *Windows*.
- d) Na distribuição Java EE, Subprojetos não são tratados pelo *NetBeans*.
- e) Os projetos não podem ser executados por meio do *NetBeans*, somente por meio de linha de comando.



## 4) IDE – *NetBeans*

---

É correto apenas:

- a) Uma de suas vantagens é o compartilhamento de biblioteca para outros usuários.
- b) Para criar arquivos Java não há necessidade de se criar um projeto anteriormente. Projetos e arquivos são conceitos tratados de forma distinta e não possuem relações.
- c) O NetBeans é um *software* que é utilizado especificamente em ambiente *Windows*.
- d) Na distribuição Java EE, Subprojetos não são tratados pelo *NetBeans*.
- e) Os projetos não podem ser executados por meio do *NetBeans*, somente por meio de linha de comando.

## 5) Proposições

---

José discutindo futebol com João declarou que se o time X venceu o último jogo, então o time Y é campeão. João depois de algumas horas descobriu que a declaração era falsa. Em consequência, teve certeza de que é verdade que:

- a) O time X não venceu o último jogo e o time Y é campeão.
- b) O time X venceu o último jogo e o time Y não é campeão.
- c) Ou o time X não venceu o último jogo ou o time Y é campeão.
- d) O time X não venceu o último jogo e o time Y não é campeão.
- e) Se o time Y não é campeão, então o time X não venceu o último jogo.

## 5) Proposições

---

Segundo de Souza (2008), uma proposição “é uma sentença declarativa que pode ser interpretada como verdadeira ou falsa”. O autor ainda adiciona que uma proposição não deve ser ambígua e não deve permitir mais de uma única interpretação.

De acordo com Alencar Filho (2002), proposição é “todo o conjunto de palavras ou símbolos que exprimem um pensamento de sentido completo”.

Essencialmente, dizemos que uma proposição é uma frase para a qual podemos atribuir um valor único de verdadeiro ou falso. Temos, a seguir, alguns exemplos de proposições.

- Júpiter é um planeta do sistema solar (proposição verdadeira).
- A Terra é um planeta do sistema solar (proposição verdadeira).
- A Lua é um planeta (proposição falsa).

## 5) Proposições

- Segundo Alencar Filho (2002), a proposição condicional é representada por  $p$  e  $q$ , cujo valor lógico é falso (F) no caso em que  $p$  é verdadeira e  $q$  é falsa e verdadeiro (V) nos demais casos.
- Costuma-se expressar a proposição condicional pelo símbolo  $\rightarrow$ , com a tabela verdade ilustrada na tabela 1.

**Tabela 1.** Tabela verdade para a proposição condicional.

<b>p</b>	<b>q</b>	<b><math>p \rightarrow q</math></b>
V	V	V
V	F	F
F	V	V
F	F	V

## 5) Proposições

- A questão pode ser enunciada no formato  $p \rightarrow q$ , sendo p a afirmação sobre o time X e q a afirmação sobre o time Y.
- Se observarmos a segunda linha da tabela 1, que é a tabela verdade para a proposição condicional, ela só é falsa quando p for verdadeira e q for falsa. Assim, podemos afirmar que o time X venceu o último jogo e o time Y não é campeão.
- **Alternativa correta: B.**

**INTERVALO**



***UNIP***

## 6) Marco Civil da Internet

*A Lei Nº 12.965, de 23 de abril de 2014, originalmente denominada Marco Civil da Internet, estabelece princípios, garantias, direitos e deveres para o uso da Internet no Brasil, contribuindo principalmente nas relações de consumo e segurança da informação pessoal para com os prestadores de serviço de acesso à Internet.*

BRASIL. Lei Nº 12.965, de 23 de abril de 2014.

- **Suponha que o cliente de determinado provedor de acesso à Internet contrate um serviço com velocidade de conexão de 5 Megabits/s (Mbps), mas tenha continuamente enfrentado problemas de perda de conexão e, adicionalmente, não obtém a velocidade contratada, apesar de o cliente realizar todos os pagamentos em dia.**

**Nessa situação, avalie as seguintes asserções e a relação proposta entre elas:**

## 6) Marco Civil da Internet

---

- I. A manutenção da qualidade contratada da conexão à Internet é uma obrigação prevista no Marco Civil da Internet.

Porque

- II. Não pode haver suspensão da conexão à Internet, salvo por débito diretamente decorrente de sua utilização.



## 6) Marco Civil da Internet

- a) As asserções I e II são proposições verdadeiras e a II é uma justificativa da I.
- b) As asserções I e II são proposições verdadeiras e a II não é uma justificativa correta da 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.

## 6) Marco Civil da Internet

- O artigo primeiro do Capítulo I da Lei Nº 12.965/2014 estabelece os princípios do Marco Civil da Internet, conforme indicado a seguir.

*Esta lei estabelece princípios, garantias, direitos e deveres para o uso da Internet no Brasil e determina as diretrizes para atuação da União, dos Estados, do Distrito Federal e dos Municípios em relação à matéria.*

## 6) Marco Civil da Internet

- No Capítulo II da Lei Nº 12965/2014, é abordada a questão dos direitos e garantias dos usuários. O artigo 7º desse capítulo está reproduzido a seguir.

*Art. 7º O acesso à Internet é essencial ao exercício da cidadania, e ao usuário são assegurados os seguintes direitos:*

*I - Inviolabilidade da intimidade e da vida privada, sua proteção e indenização pelo dano material ou moral decorrente de sua violação;*

*II - inviolabilidade e sigilo do fluxo de suas comunicações pela Internet, salvo por ordem judicial, na forma da lei;*

*III - inviolabilidade e sigilo de suas comunicações privadas armazenadas, salvo por ordem judicial;*

*IV - Não suspensão da conexão à Internet, salvo por débito diretamente decorrente de sua utilização;*

*V - Manutenção da qualidade contratada da conexão à Internet;*

*VI - informações claras e completas constantes dos contratos de prestação de serviços, com detalhamento sobre o regime de proteção aos registros de conexão e aos registros de acesso a aplicações de Internet, bem como sobre práticas de gerenciamento da rede que possam afetar sua qualidade.*

## 6) Marco Civil da Internet

---

### I. Asserção correta.

- Justificativa. De acordo com o item V do artigo 7º do Capítulo II da Lei Nº 12.965/2014, os provedores devem garantir a qualidade do serviço contratado.

### II. Asserção correta.

- Justificativa. De acordo com o item IV do artigo 7º do Capítulo II da Lei Nº 12.965/2014, não deve haver suspensão arbitrária da conexão. A desconexão é aplicável apenas no caso de o cliente não estar em dia com os pagamentos do serviço contratado, o que não é a situação proposta na questão.

## 6) Marco Civil da Internet

---

- Ainda que ambas as asserções estejam corretas, não existe relação de causalidade entre elas. Essas asserções dizem respeito a aspectos diferentes da legislação: a asserção I refere-se à obrigação dos provedores de serviço com relação à qualidade e a asserção II refere-se às condições que podem levar à suspensão do serviço.
- **Alternativa correta: B.**

## 7) Barramentos

---

- O barramento é o elemento de conexão entre todos os componentes do computador, como memória, CPU e dispositivos de entrada e saída. O barramento de dados é o meio por onde serão trafegados os dados. O barramento de endereços transporta a informação do endereço do dispositivo que poderá acessar o barramento de dados; e o barramento de controle serve para determinar o sentido do fluxo de dados (se os dados são de entrada ou saída da CPU), e se os dados devem ser destinados à memória ou aos dispositivos de I/O e também para controlar o *clock* no barramento.

## 7) Barramentos

Considerando um computador com um barramento de dados de 4 bits e barramento de endereços de 3 bits, ele poderá endereçar, respectivamente, quantas posições de memória e quantos dispositivos de I/O?

- a) 4 e 4.
- b) 4 e 8.
- c) 8 e 3.
- d) 8 e 4.
- e) 8 e 8.

## 7) Barramentos

---

- Um computador é um dispositivo eletrônico formado por diversos componentes que se comunicam de forma rápida e eficiente. Podemos dizer que seus elementos principais são a CPU, as memórias e os dispositivos de entrada e saída. É necessário que esses dispositivos se comuniquem uns com os outros e a forma como isso é realizado tem efeitos no custo e no desempenho de um equipamento. Isso leva à questão da interconexão entre os diversos elementos que constituem um computador.



## 7) Barramentos

- Para compreendermos o problema da interconexão em uma situação na qual dispomos de vários dispositivos (por exemplo, 5 elementos), vamos pensar na elaboração de um sistema em que todos os dispositivos possam se comunicar uns com os outros. Uma forma de fazer isso é conectar diretamente todos os elementos entre si, como indicado na figura 1.

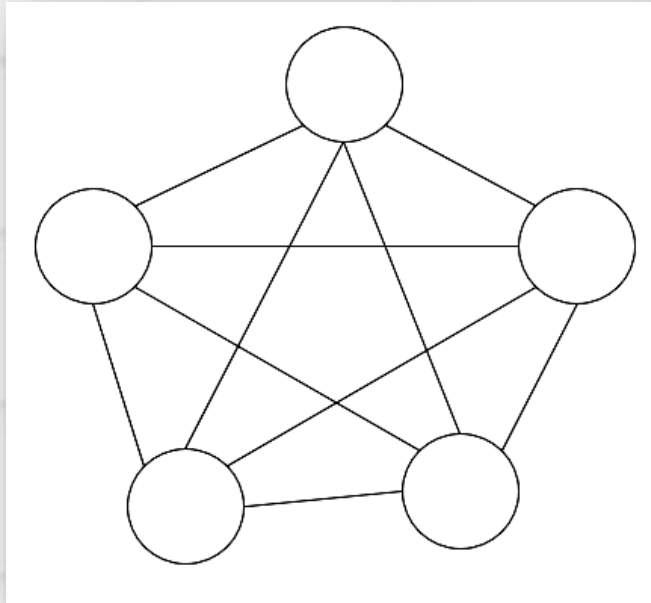


Figura 1. Interconexão de vários elementos.

## 7) Barramentos

---

- O problema dessa abordagem é que o número de interconexões aumenta de forma muito rápida com o crescimento do número de elementos a serem conectados, o que torna essa abordagem inviável para dispositivos complexos, como um computador.
- Uma outra abordagem consiste em se construir um barramento, uma espécie de “rodovia” que interconecta todos os elementos, como indicado na figura 2. Dessa forma, ao conectarmos novos elementos no sistema, não precisamos conectar o novo elemento a todos os demais; podemos conectá-lo apenas ao barramento.

## 7) Barramentos

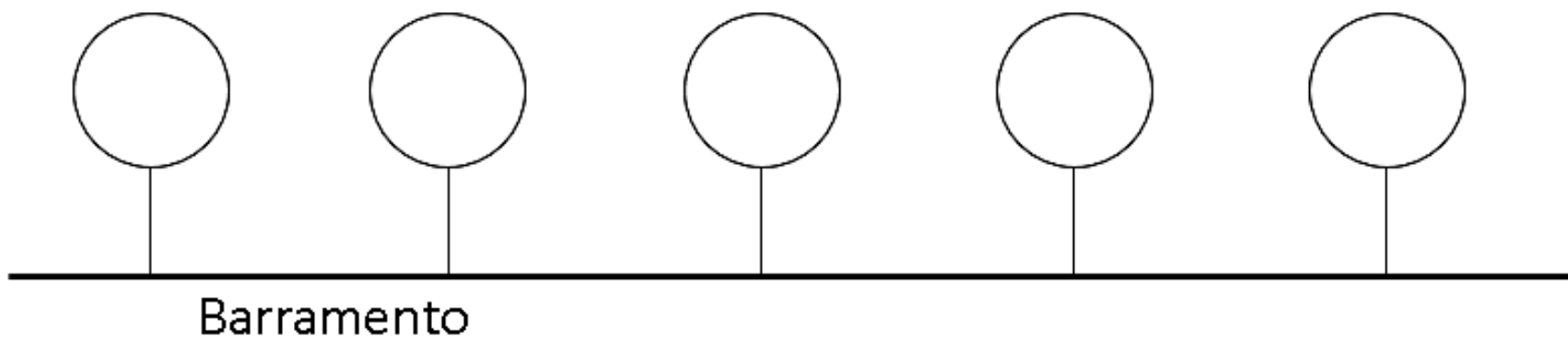


Figura 2. Interconectando vários elementos através de um barramento

## 7) Barramentos

- Considerando que o barramento de endereços atua tanto para o endereçamento da memória quanto para o atendimento aos dispositivos de entrada e saída, e sabendo que esse barramento tem largura de 3 bits (valor dado na questão), ele deve ser capaz de endereçar  $2^3=8$  posições diferentes. Na arquitetura do enunciado, o barramento de controle vai definir quando esses bits endereçam posições de memória ou dispositivos de entrada e saída, sendo que os barramentos de dados serão utilizados para ambos os casos, endereçando 8 posições de memória diferentes ou 8 dispositivos diferentes.
- Alternativa correta: E.

## 8) Função recursiva

- Uma função é denominada recursiva quando ela é chamada novamente dentro de seu corpo. Implementações recursivas tendem a ser menos eficientes, porém facilitam a codificação e seu entendimento.

Considere a função recursiva `f()`, a qual foi escrita em linguagem C:

```
1 int f( int v[], int n){
2     if(n == 0)
3         return 0;
4     else{
5         int s;
6         s = f(v, n-1);
7         if( v[n-1] > 0 ) s = s + v[n-1];
8         return s;
9     }
10 }
```

## 8) Função recursiva

Suponha que a função  $f()$  é acionada com os seguintes parâmetros de entrada:

$f(\{2,-4,7,0,-1,4\},6);$

Nesse caso, o valor de retorno da função  $f()$  será:

- a) 8.
- b) 10.
- c) 13.
- d) 15.
- e) 18.

## 8) Função recursiva

---

Em um programa computacional, quando temos uma função (ou procedimento) chamando a si mesma, chamamos esse mecanismo de recursividade. À primeira vista, o fato de uma função chamar a si mesma pode parecer estranho e talvez até mesmo errado: se uma função chama a si mesma de forma contínua, quando o processo irá parar?

- Ao criar uma função recursiva, o programador deve evitar situações em que o programa nunca termine, com uma função chamando a si mesma sem nunca se estabelecer um critério de parada. Dessa forma, deve existir uma condição na qual ocorra recursividade e outra condição na qual a função retorne algum valor.

## 8) Função recursiva

- Para a resolução do problema, podemos construir uma tabela na qual simulamos a execução do programa. Observe que essa tabela será construída na ordem em que a função  $f(v,n)$  retorna aos valores, e não na ordem em que ela é chamada. Isso acontece porque essa é uma função recursiva e o primeiro valor a ser retornado é 0, na linha 3, quando  $n$  é igual a 0. A função  $f(v[0], 0)$  retorna a  $s=0$  na linha 6. Como  $v[0]=2>0$ , sabemos que  $s=s+2=0+2=2$ . Sendo assim,  $f(v,1)=2$ . Esse valor é novamente retornado à linha 6. Com  $f(v[1],2)$ , mas  $v[1]=-4<0$ , portanto,  $f(v,2)=2$ . Esse processo é repetido até que se chegue ao valor  $f(v,6)=13$ , conforme tabela 1.



## 8) Função recursiva

Tabela 1. Resultado de  $f(v,n)$  para a execução do programa.

N	$f(v,n)$
0	0
1	2
2	2
3	9
4	9
5	9
6	13

- Alternativa correta C.

**INTERVALO**



***UNIP***

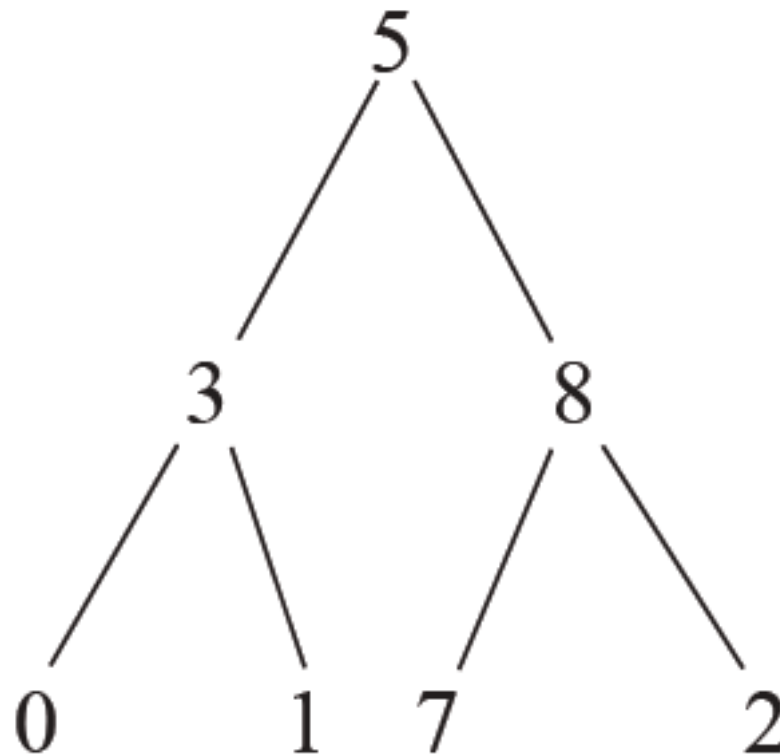
## 9) Árvores binárias

- Existem várias maneiras de se percorrer uma árvore binária. A função a seguir, escrita em pseudocódigo, percorre uma árvore na ordem esquerda-raiz-direita, conhecida por varredura e-r-d recursiva. A função `erd()` recebe por parâmetro a raiz `r` de uma árvore e faz uso de seus elementos `esq`, `dir` e `cont`, que representam, respectivamente, ponteiros para uma subárvore à esquerda de `r`, uma subárvore à direita de `r` e o conteúdo de `r`.

```
função erd( árvore r)
{
    se( r != NULO )
    {
        erd( r->esq);
        escreva(r->conteúdo);
        erd(r->dir);
    }
}
```

## 9) Árvores binárias

Considere a árvore binária a seguir:



## 9) Árvores binárias

---

A sequência correta de exibição do conteúdo da árvore utilizando a função `erd()` é:

- a) 5,3,8,0,1,7,2.
- b) 0,1,7,2,3,8,5.
- c) 0,3,5,1,7,8,2.
- d) 0,3,1,5,7,8,2.
- e) 2,7,8,5,0,3,1.

## 9) Árvores binárias

---

- Listas ligadas, pilhas e vetores são estruturas de dados muito úteis para representação de vários tipos de informações em programas computacionais. Contudo, nem sempre conseguimos representar informações utilizando esses tipos de estruturas. Isso ocorre à medida que o relacionamento entre os nós começa a se tornar mais complexo, com mais possibilidades de interligação.
- Uma das formas de estrutura de dados mais comuns para a representação de informações hierárquicas é a árvore (CELES, CERQUEIRA e RANGEL, 2004). Na computação, uma árvore corresponde a uma estrutura que contém um nó raiz (desenhado no topo) e uma série de nós filhos (que correspondem aos ramos).

## 9) Árvores binárias

---

- Existem vários tipos de árvores, cada uma com uma finalidade específica. Um dos tipos mais comuns e úteis é a chamada árvore binária. Nesse caso, cada nó pode ter zero, um ou dois nós filhos. Os últimos elementos da árvore, normalmente desenhados na sua porção inferior, são chamados de nós folhas.
- Uma das operações importantes que sempre devemos ser capazes de executar em uma estrutura de dados é chamada de percurso. Por exemplo, frequentemente queremos percorrer uma lista, visitando cada um dos seus elementos. Ou então percorrer um vetor, lendo cada um dos seus elementos ou fazendo outra operação com esses elementos. Também podemos percorrer uma árvore binária visitando cada um dos seus elementos.

## 9) Árvores binárias

---

- É conveniente utilizar funções recursivas para situações em que queremos percorrer estruturas de dados como árvores binárias e listas ligadas. Devido a essas estruturas terem referências (ou ponteiros) para outros elementos da mesma classe (ou tipo), a utilização de funções que seguem essas referências é bastante conveniente.



## 9) Árvores binárias

---

- Para resolvermos a questão, enumeramos três pontos importantes da função “erd”, mostrados na figura 1. Devemos observar que um número só é impresso quando a função “escreva” é chamada. Quando r for Nulo, a função “erd” retorna sem efetuar nenhuma ação. Caso contrário, a função “erd” é chamada de forma recursiva nos pontos 1 e 3.

## 9) Árvores binárias

```
função erd( árvore r)
{
  se( r != NULO )
  {
    ① erd( r->esq);
    ② escreva(r->conteúdo);
    ③ erd(r->dir);
  }
}
```

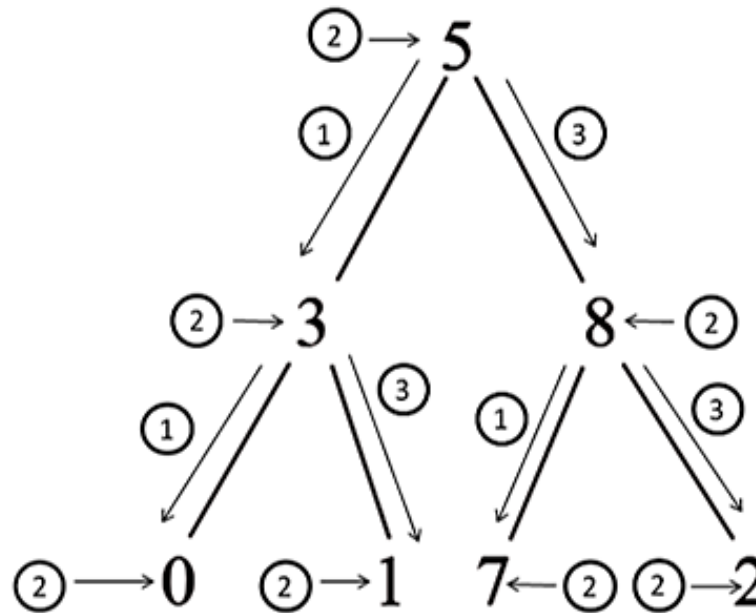


Figura 1. Representação gráfica de uma árvore binária ao lado da função “erd” utilizada para percorrer os elementos de uma árvore.

## 9) Árvores binárias

- Para que a função “escreva” seja chamada, a função “erd” chamada no ponto 1 deve retornar, o que ocorre apenas quando r for igual a NULO. Logo, a primeira impressão deve ser a de um dos nós folhas da árvore. Nós folhas são os últimos nós de uma árvore computacional. Observe que, em computação, a árvore é desenhada “de cabeça para baixo”, com a raiz no topo do desenho e as folhas na parte de baixo.

Na figura 2, temos o desenho da mesma árvore do enunciado, com a ordem de impressão dos elementos dentro de quadrados ao lado dos nós da árvore. Compare essa figura e a figura 1 e observe os sentidos das setas. O número dentro das circunferências na figura 1 mostra o ponto no código em que o programa toma determinada decisão.

Observe que:

- no ponto 1, sempre percorremos o ramo esquerdo da árvore;
- no ponto 2, sempre imprimimos um elemento;
- no ponto 3, sempre percorremos o ramo direito.

## 9) Árvores binárias

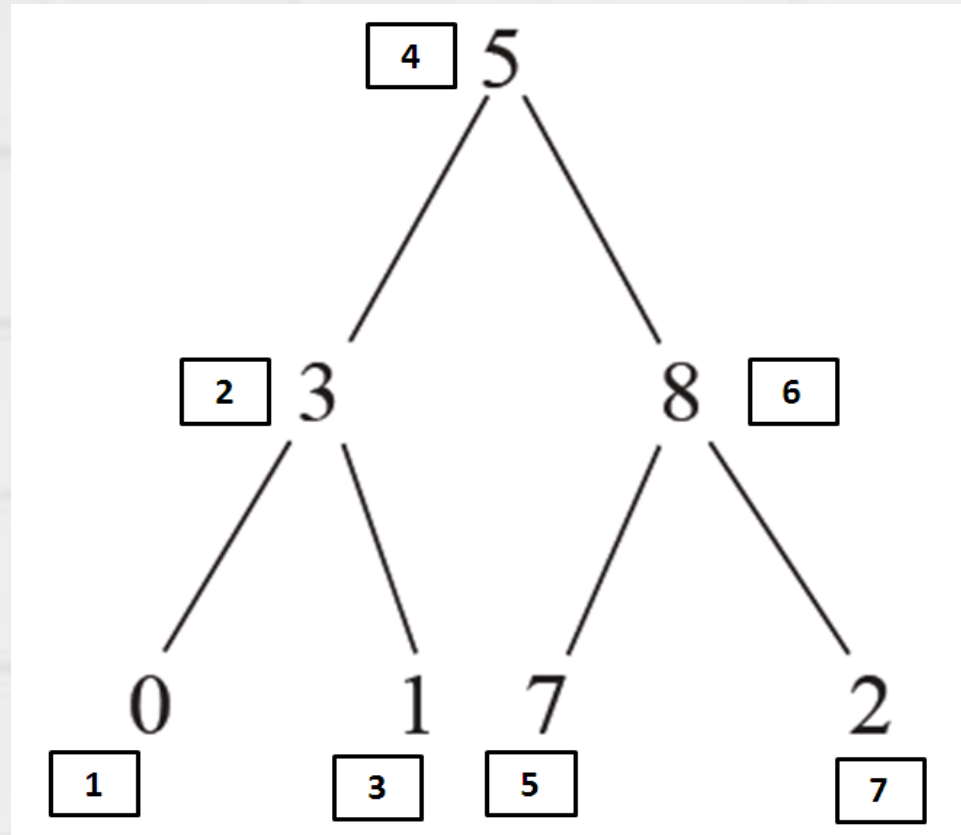


Figura 2. Ordem de impressão dos elementos da árvore binária.

- Alternativa correta: D.

**ATÉ A PRÓXIMA!**



***UNIP***



# ***UNIP***

---

## **ESTUDOS DISCIPLINARES**

### **Formação Específica**

**Profa. Me. Sandra Bozolan**

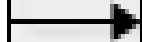
## 10) Programação OO e Web

---

Em um sistema de cartões de crédito, existe a seguinte funcionalidade:

### Consultar fatura do cliente

Tela – Consultar  
Fatura



Informar dados do  
cartão do cliente



Exibir informações da  
fatura do cliente

## 10) Programação OO e Web

---

- a) Para que os dados sejam exibidos após uma consulta ao banco de dados, é preciso desenvolver classes em XML que sejam capazes de acessar o banco de dados remotamente.
- b) Desenvolver, inicialmente, classe/método a fim de efetuar a instância do método criado e garantir o desenvolvimento parcial da consulta ao banco de dados.
- c) Desenvolver arquivos REF que irão acessar o banco de dados de forma dinâmica.
- d) Atribuir parametrização de valores dentro do XML para que a consulta ao banco de dados seja efetuada com sucesso.
- e) Desenvolver classes atribuindo *switch/case* para acessar a consulta ao banco de dados.



## 10) Programação OO e Web

---

- a) Para que os dados sejam exibidos após uma consulta ao banco de dados, é preciso desenvolver classes em XML que sejam capazes de acessar o banco de dados remotamente.
- b) Desenvolver, inicialmente, classe/método a fim de efetuar a instância do método criado e garantir o desenvolvimento parcial da consulta ao banco de dados.**
- c) Desenvolver arquivos REF que irão acessar o banco de dados de forma dinâmica.
- d) Atribuir parametrização de valores dentro do XML para que a consulta ao banco de dados seja efetuada com sucesso.
- e) Desenvolver classes atribuindo *switch/case* para acessar a consulta ao banco de dados.

# 11) Programação OO e Web

---

- Uma locadora de veículos pretende aumentar os ganhos de locação em, pelo menos, 30% ao mês. Porém, para atingir essa meta, precisa efetuar melhorias no sistema atual. Uma consultoria de *software* foi contratada para analisar e propor sugestões para atender à meta do cliente. Após a análise efetuada, foi levantado que, no sistema atual, o código desenvolvido encontra-se totalmente desorganizado, ou seja, é de difícil entendimento e não tem documentação de apoio.

Como sugestão inicial para o cliente, foi proposto que o código fosse revisado e otimizado, aplicando conceitos de programação orientada a objetos. Para que o código seja estruturado de maneira clara, pode-se:

# 11) Programação OO e Web

---

- a) Trabalhar, inicialmente, com a construção de métodos, entendendo-se por método uma sequência nomeada de instruções que atenderia à necessidade inicial de entendimento do projeto.
- b) Redefinir todas as variáveis de sistema, a fim de garantir a sequência de desenvolvimento do projeto.
- c) Reescrever todos os arquivos XML que se encontram no sistema, pois, assim, o sistema teria o aumento de performance desejado pelo cliente.
- d) Utilizar conceitos de LINQ, pois o LINQ garantiria o aumento de performance desejado pelo cliente.
- e) Reescrever todas as dll's do sistema, atendendo às expectativas iniciais do projeto.

# 11) Programação OO e Web

---

- a) **Trabalhar, inicialmente, com a construção de métodos, entendendo-se por método uma sequência nomeada de instruções que atenderia à necessidade inicial de entendimento do projeto.**
- b) **Redefinir todas as variáveis de sistema, a fim de garantir a sequência de desenvolvimento do projeto.**
- c) **Reescrever todos os arquivos XML que se encontram no sistema, pois, assim, o sistema teria o aumento de performance desejado pelo cliente.**
- d) **Utilizar conceitos de LINQ, pois o LINQ garantiria o aumento de performance desejado pelo cliente.**
- e) **Reescrever todas as dll's do sistema, atendendo às expectativas iniciais do projeto.**

## 12) Programação OO e Web

---

Um desenvolvedor foi acionado após a implantação de sistemas que visavam à melhoria de alguns códigos, como a criação de novas instâncias e de novas classes. Quando o operador efetuou uma ação de consulta no sistema, foi apresentada a seguinte mensagem:

“Erro na chamada do método consulta dados cliente”

- Com o SLA de 1h na análise, seria apontado que:

## 12) Programação OO e Web

---

- a) A classe `consulta_dados_cliente` está com excesso de parâmetros, gerando sobrecarga no sistema, e, conseqüentemente, a mensagem de erro.
- b) A classe `consulta_dados_cliente` está tráfegando parâmetros dos tipos inteiro, *string* e *float* e, por isso, gerou a mensagem de erro.
- c) O método `consulta_dados_cliente` é um método privado e não pode ser acessado, ocasionando erro em sua chamada.
- d) A instância criada não faz referência ao método `consulta_dados_cliente`, gerando a mensagem de erro.
- e) A instância criada não faz referência à classe `consulta_dados_cliente`, gerando a mensagem de erro.

## 12) Programação OO e Web

---

- a) A classe `consulta_dados_cliente` está com excesso de parâmetros, gerando sobrecarga no sistema, e, conseqüentemente, a mensagem de erro.
- b) A classe `consulta_dados_cliente` está tráfegando parâmetros dos tipos inteiro, *string* e *float* e, por isso, gerou a mensagem de erro.
- c) O método `consulta_dados_cliente` é um método privado e não pode ser acessado, ocasionando erro em sua chamada.
- d) A instância criada não faz referência ao método `consulta_dados_cliente`, gerando a mensagem de erro.
- e) A instância criada não faz referência à classe `consulta_dados_cliente`, gerando a mensagem de erro.

## 13) Programação OO e Web

Considere o seguinte documento XML:

```
<?xml version="1.0"?>
<hamburgeres>
  <hamburger baixacaloria="Nao">
    <nome>CowBurger</nome >
    <descricao>Hamburger      de      carne      de
vaca.</descricao>
    <preco>2</preco>
  </hamburger>
</hamburgeres >
```

Fonte: FVG, 2013



# 13) Programação OO e Web

Considere também o seguinte código XSL:

```
<xsl:stylesheet                                version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
<h2>My CD Collection</h2>
<table border="1">
<tr bgcolor="#9acd32">
<th>Title</th>
<th>Artist</th>
</tr>
<xsl:for-each select="catalog/cd">
<tr>
<td><xsl:value-of select="title"/></td>
<td><xsl:value-of select="artist"/></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

## 13) Programação OO e Web

---

**Assinale a alternativa que descreve a saída da execução do código XSLT, tendo como entrada o documento XML mostrado anteriormente.**

- a) Será gerado um documento HTML.**
- b) Não é possível aplicar esse código XSLT sobre o documento XML apresentado.**
- c) O documento XML apresentado não é válido.**
- d) Em XSL/XSLT não existe o elemento template.**
- e) Será gerado um arquivo WSDL.**

Fonte: FVG, 2013

## 13) Programação OO e Web

---

Assinale a alternativa que descreve a saída da execução do código XSLT, tendo como entrada o documento XML mostrado anteriormente.

- a) **Será gerado um documento HTML.**
- b) Não é possível aplicar esse código XSLT sobre o documento XML apresentado.
- c) O documento XML apresentado não é válido.
- d) Em XSL/XSLT não existe o elemento template.
- e) Será gerado um arquivo WSDL.

Fonte: FVG, 2013

## 14) Programação OO e Web

O CSS (*cascading style sheets*) tem a finalidade de separar o estilo de uma página web a partir de sua estrutura, facilitando sua manutenção. Em relação ao CSS, assinale a alternativa correta.

- a) O CSS é exibido normalmente em todos os navegadores, inclusive nas versões mais antigas.
- b) Com o CSS, é possível especificar o posicionamento dos elementos em uma página.
- c) Uma de suas desvantagens é não permitir especificar a cor de uma fonte.
- d) Na estrutura do CSS, o seletor é o componente de uma regra de estilo que tem a função de especificar como fica o conteúdo descrito pela marcação.
- e) Um seletor único inclui apenas uma declaração.

**INTERVALO**



***UNIP***

## 14) Programação OO e Web

---

O CSS (*cascading style sheets*) tem a finalidade de separar o estilo de uma página web a partir de sua estrutura, facilitando sua manutenção. Em relação ao CSS, assinale a alternativa correta.

- a) O CSS é exibido normalmente em todos os navegadores, inclusive nas versões mais antigas.
- b) **Com o CSS, é possível especificar o posicionamento dos elementos em uma página.**
- c) Uma de suas desvantagens é não permitir especificar a cor de uma fonte.
- d) Na estrutura do CSS, o seletor é o componente de uma regra de estilo que tem a função de especificar como fica o conteúdo descrito pela marcação.
- e) Um seletor único inclui apenas uma declaração.

## 15) Programação OO e Web

---

Os tipos primitivos são utilizados pelo JavaScript quando se desejam armazenar dados simples. Um dos tipos primitivos utilizados pelo JavaScript é o:

- a) Varchar.
- b) Char.
- c) Date.
- d) Undefined.
- e) Blob.

Fonte: FUNIVERSA, 2014

## 15) Programação OO e Web

---

Os tipos primitivos são utilizados pelo JavaScript quando se desejam armazenar dados simples. Um dos tipos primitivos utilizados pelo JavaScript é o:

- a) Varchar.
- b) Char.
- c) Date.
- d) Undefined.**
- e) Blob.

Fonte: FUNIVERSA, 2014



## 16) Programação OO e Web

Com relação aos atributos contidos no trecho de HTML apresentado a seguir, assinale a alternativa correta:

```

```

- a) O atributo `img` indica a origem da figura a ser acrescentada à página.
- b) O atributo `src` tem a função específica de disponibilizar determinado objeto para *download*.
- c) O atributo `alt` fornece um texto alternativo à imagem para a exibição em navegadores que leem somente texto.
- d) O atributo *width* define a altura da linha, o que pode ser feito em percentual da tela.
- e) O atributo *height* define a largura da linha, o que pode ser feito em *pixels* (número absoluto).

## 16) Programação OO e Web

Com relação aos atributos contidos no trecho de HTML apresentado a seguir, assinale a alternativa correta:

```

```

- a) O atributo `img` indica a origem da figura a ser acrescentada à página.
- b) O atributo `src` tem a função específica de disponibilizar determinado objeto para *download*.
- c) O atributo `alt` fornece um texto alternativo à imagem para a exibição em navegadores que leem somente texto.
- d) O atributo `width` define a altura da linha, o que pode ser feito em percentual da tela.
- e) O atributo `height` define a largura da linha, o que pode ser feito em *pixels* (número absoluto).

## 17) Programação OO e Web

---

Na *string* de conexão de uma base de dados SQL Server 2012, o parâmetro *Trusted\_Connection* indica se:

- a) A conexão deve ser criptografada utilizando SSL/TSL.
- b) A conexão com a base de dados é confiável, isto é, se possui certificado de autenticidade emitido.
- c) O banco de dados deve confiar nas credenciais fornecidas pelo Windows para realizar a autenticação.
- d) O banco de dados não requer qualquer tipo de autenticação para ser acessado.
- e) Deve ser utilizada uma conexão existente com o banco de dados.

## 17) Programação OO e Web

Na *string* de conexão de uma base de dados SQL Server 2012, o parâmetro *Trusted\_Connection* indica se:

- a) A conexão deve ser criptografada utilizando SSL/TSL.
- b) A conexão com a base de dados é confiável, isto é, se possui certificado de autenticidade emitido.
- c) O banco de dados deve confiar nas credenciais fornecidas pelo Windows para realizar a autenticação.
- d) O banco de dados não requer qualquer tipo de autenticação para ser acessado.
- e) Deve ser utilizada uma conexão existente com o banco de dados.

## 18) Programação OO e Web

Em um programa escrito na linguagem HTML, inseriu-se um texto entre as marcações <PRE> e </PRE>. Após esse programa ser aberto por um navegador típico que é o suporte, é correto afirmar que o texto será apresentado:

- a) De acordo com a formatação presente no último parágrafo exibido.
- b) De acordo com o formato definido no arquivo index.htm.
- c) De acordo com um estilo predefinido no navegador.
- d) De acordo com uma formatação predefinida no início do programa.
- e) Exatamente da forma como foi digitado.

## 18) Programação OO e Web

Em um programa escrito na linguagem HTML, inseriu-se um texto entre as marcações <PRE> e </PRE>. Após esse programa ser aberto por um navegador típico que é o suporte, é correto afirmar que o texto será apresentado:

- a) De acordo com a formatação presente no último parágrafo exibido.
- b) De acordo com o formato definido no arquivo index.htm.
- c) De acordo com um estilo predefinido no navegador.
- d) De acordo com uma formatação predefinida no início do programa.
- e) **Exatamente da forma como foi digitado.**

## 19) Programação OO e Web

- Em um formulário de uma página HTML5 há o seguinte campo
- CPF: `<input type="text" name="cpf" required>`

No Google Chrome, para que o formulário seja submetido somente se o CPF for digitado no formato XXXXXXXXX-XX, em que X representa qualquer número decimal de 0 a 9, é necessário colocar nesse campo o atributo:

- a) `placeholder="XXXXXXXXXX-XX"`
- b) `paterna="^\d{9}\-\d{2}$"`
- c) `pattern="[0-9]{11,3}"`
- d) `placeholder="^[0-9]\-\d{2}$"`
- e) `format="^[0-9]\-\d{2}$"`

## 19) Programação OO e Web

- Em um formulário de uma página HTML5 há o seguinte campo
- CPF: `<input type="text" name="cpf" required>`

No Google Chrome, para que o formulário seja submetido somente se o CPF for digitado no formato XXXXXXXXX-XX, em que X representa qualquer número decimal de 0 a 9, é necessário colocar nesse campo o atributo:

- a) **placeholder="XXXXXXXXXX-XX"**
- b) `paterna="^\d{9}\-\d{2}$"`
- c) `pattern="[0-9]{11,3}"`
- d) `placeholder="^[0-9]\-\d{2}$"`
- e) `format="^[0-9]\-\d{2}$"`



## 20) Programação OO e Web

Considere  
XML a seguir:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE empresa[
<!ELEMENT empresa (entrega+,produto+)>
<!ELEMENT entrega (tempo?)>
<!ATTLIST entrega meioEntrega ID #REQUIRED>
<!ELEMENT produto (#PCDATA)>
<!ATTLIST produto formaEntrega IDREF #IMPLIED>
<!ELEMENT tempo (#PCDATA)>
]>
<empresa>
<entrega meioEntrega="Motoboy">
<tempo> 3 dias </tempo>
</entrega>
<entrega meioEntrega="Correios">
<tempo> 1 dia – São Paulo </tempo>
<tempo> 5 dias - outros estados </tempo>
</entrega>
<produto formaEntrega="Correios">Playstation</produto>
<produto>Geladeira</produto>
<produto formaEntrega="Correios">Tablet</produto>
</empresa>
```

## 20) Programação OO e Web

O arquivo não é válido, de acordo com a Document Type Definition-DTD, porque:

- a) Todo elemento produto precisa ter, obrigatoriamente, um atributo formaEntrega, mas um deles não tem.
- b) Deve haver, pelo menos, um produto cujo atributo formaEntrega possua o valor "Motoboy", o que não ocorre.
- c) O elemento entrega pode conter nenhuma ou uma ocorrência do elemento tempo, mas há duas ocorrências em um dos elementos entrega.
- d) O elemento tempo não pode conter números, devido à definição de tipo de dado PCDATA, mas contém o número de dias para entrega dos produtos.
- e) O elemento empresa só pode ter uma ocorrência do elemento entrega, mas há duas ocorrências.

## 20) Programação OO e Web

O arquivo não é válido, de acordo com a Document Type Definition–DTD, porque:

- a) Todo elemento produto precisa ter, obrigatoriamente, um atributo formaEntrega, mas um deles não tem.
- b) Deve haver, pelo menos, um produto cujo atributo formaEntrega possua o valor "Motoboy", o que não ocorre.
- c) **O elemento entrega pode conter nenhuma ou uma ocorrência do elemento tempo, mas há duas ocorrências em um dos elementos entrega.**
- d) O elemento tempo não pode conter números, devido à definição de tipo de dado PCDATA, mas contém o número de dias para entrega dos produtos.
- e) O elemento empresa só pode ter uma ocorrência do elemento entrega, mas há duas ocorrências.

**INTERVALO**



***UNIP***

## 21) Empreendedorismo

---

- O plano de negócios, mais do que um documento de elaboração das ações de implementação de um novo empreendimento, serve como documento que estabelece o relacionamento entre empreendedores e investidores. O conhecimento de características dos atores envolvidos nessa relação interfere diretamente na elaboração do plano de negócios. Considerando os papéis do empreendedor, do investidor e de conceitos de fatores envolvidos na elaboração do plano de negócios, assinale a opção correta.

## 21) Empreendedorismo

---

- a) O verdadeiro empreendedor cria um negócio diante de uma oportunidade e procura, o mais breve possível, vendê-lo para um grupo de investidores.
- b) Investidores inteligentes consideram, ao analisar onde investir, que projeções financeiras mês a mês para um período maior que um ano constituem um dos fatores que garante o sucesso de um novo empreendimento.
- c) O empreendedor é uma pessoa à procura de riscos, que diante de uma nova oportunidade de empreendimento transfere todos os riscos para si.
- d) As pessoas, as oportunidades, o contexto e as possibilidades de riscos e recompensas são quatro fatores fundamentais, que devem ser considerados para o sucesso de um novo empreendimento.
- e) Um plano de negócios deve ser criado seguindo uma fórmula de sucesso preestabelecida apresentada em livros da área **UNIP** administração e implementada em aplicativos.

## 21) Empreendedorismo

---

Uma empresa é criada para sobreviver no longo prazo. Para isso, algumas ações são fundamentais, como as citadas a seguir.

- Inovar para crescer.
- Investir em qualificação.
- Fazer um planejamento estratégico.
- Manter a qualidade de produtos e serviços.
- Gerenciar o negócio.
- Controlar os custos.
- Manter a base de clientes.

## 21) Empreendedorismo

---

- O plano de negócios é o documento que estrutura as ideias e as opções que o empreendedor analisará para decidir sobre a viabilidade do seu negócio. Deve conter informações sobre o negócio, as previsões e as projeções financeiras, a análise do mercado, a previsão de fluxo de caixa e as necessidades de capital.
- O empreendedor é aquele que inicia novos negócios e consegue escolher entre várias alternativas. Ele apresenta determinadas habilidades e competências para criar, abrir e gerir um negócio, gerando resultados positivos.



## 21) Empreendedorismo

---

- É impossível afastar o elemento risco do negócio, por isso ele tem que ser gerenciado e acompanhado, com o objetivo de minimizá-lo. Assumir riscos é a principal característica do empreendedor.
- Não existem fórmulas mágicas para ser um empreendedor de sucesso. Há exemplos de inúmeros negócios que prometiam sucesso e fracassaram. Igualmente, temos também inúmeros negócios que nasceram por acaso e hoje são completos sucessos.

## 21) Empreendedorismo

---

### A – Alternativa incorreta.

- Justificativa. A criação do negócio é voltada para o longo prazo e para a permanência da empresa.

### B – Alternativa incorreta.

- Justificativa. As projeções financeiras devem abranger vários períodos de análise (mensais, semestrais, anuais e acima de um ano). Quanto mais distante o prazo, menos precisas serão. Ao projetar um investimento, o investidor deve levar em conta o período necessário para que o capital seja remunerado, o que pode levar vários anos.

## 21) Empreendedorismo

---

### C – Alternativa incorreta.

- Justificativa. Embora seja correto afirmar que o empreendedor assume riscos, ele não os procura (os riscos são inerentes ao negócio). Muito menos transfere todos para si mesmo.

### D – Alternativa correta.

- Justificativa. Os fatores listados estão corretos. O negócio tem que ser avaliado em várias dimensões para que não sejam cometidos erros graves que o inviabilizem.

## 21) Empreendedorismo

---

E – Alternativa incorreta.

- Justificativa. Não existe uma fórmula de sucesso preestabelecida. Sucessos passados não garantem a viabilidade futura de um negócio similar. Cada um tem que ser analisado como um evento singular, mesmo que sejam comparados com negócios ou mercados similares.

**INTERVALO**



***UNIP***

## 22) Serviço DNS (*Domain Name System*)

---

- O serviço DNS (*Domain Name System*) traduz nomes alfanuméricos de *hosts* em endereços numéricos, de acordo com o protocolo IP (*Internet Protocol*). Essa ação é comumente chamada de resolução de endereço.
- TANENBAUM, A. S. Redes de Computadores. Rio de Janeiro: Campus, 2003 (com adaptações).
- Considere um conjunto de computadores conectados em uma rede local, os quais têm à sua disposição um servidor DNS capaz de resolver endereços, sejam eles internos ou externos.

## 22) Serviço DNS (*Domain Name System*)

---

Nesse contexto, avalie as afirmativas a seguir.

- I. O servidor DNS também executa funções de cliente DNS quando não é autoritativo para determinado endereço.
- II. A adoção do IPv6 (*Internet Protocol*, versão 6) dispensará serviços de DNS, pois suas funções serão incorporadas pelo próprio protocolo IP.
- III. O cache DNS permite que determinada requisição do cliente DNS possa ser resolvida sem que seja necessário recorrer a outro serviço DNS.
- IV. O protocolo DNS depende de um banco de dados distribuído.

## 22) Serviço DNS (*Domain Name System*)

---

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.



## 22) Serviço DNS (*Domain Name System*)

---

- As máquinas presentes em uma rede que usa o protocolo TCP/IP têm ao menos um endereço numérico, chamado de endereço IP. O protocolo IPv4 especifica que o endereço IP deve possuir 32 *bits*, mas devido à grande expansão da internet após o final da década de 1990, esse tamanho tornou-se muito restritivo. Foi criado, então, o IPv6, de 128 *bits*, aumentando também a quantidade disponível de endereços. Contudo, a implantação do IPv6 vem ocorrendo de forma gradual e ainda não está completa.

## 22) Serviço DNS (*Domain Name System*)

- A ideia de duas máquinas comunicarem-se utilizando endereços IP é bastante conveniente para os computadores, mas não necessariamente para seres humanos. Por exemplo, para acessar o *site* do Google ([www.google.com.br](http://www.google.com.br)) pelo endereço IP (v4, de 32 *bits*), deveríamos digitar o número 200.196.224.129. Obviamente, navegar por *sites* utilizando números grandes é bastante inconveniente para a maioria dos usuários, sendo muito mais simples utilizar o nome [www.google.com.br](http://www.google.com.br). Contudo, as máquinas continuam utilizando o endereço IP para a comunicação. Logo, é necessário que exista alguma tecnologia para “traduzir” o nome do endereço (no caso, [www.google.com.br](http://www.google.com.br)) para o endereço IP 200.196.224.129.

## 22) Serviço DNS (*Domain Name System*)

---

- O serviço de DNS faz precisamente essa “tradução”. Dessa forma, ao digitarmos a URL de um *síte* no navegador, o servidor de DNS procura qual é o endereço IP correspondente a [www.google.com.br](http://www.google.com.br) e retorna essa informação para o computador cliente, que vai agora utilizar o endereço IP para a comunicação. Para o usuário, essa é uma operação transparente: ele apenas deve saber a URL do *síte* ([www.google.com.br](http://www.google.com.br)) e o computador, automaticamente, solicita o endereço IP para o servidor de DNS.

## 22) Serviço DNS (*Domain Name System*)

---

- Devido ao grande número de *sites* que existem (um número em constante aumento), o tamanho do banco de dados de DNS é muito grande. Tal informação é importante, pois, por exemplo, a associação do endereço IP de um banco ou de uma instituição financeira à sua URL é alvo de ataques maliciosos em busca de vulnerabilidades. Logo, servidores de DNS devem ser cuidadosamente protegidos.

## 22) Serviço DNS (*Domain Name System*)

---

- Além disso, o número de clientes requisitando endereços IP de um servidor de DNS é muito grande, uma vez que o número de máquinas na internet é gigantesco. Assim, é interessante que exista um grande número de servidores de DNS para distribuir a carga da consulta entre diferentes máquinas, preferencialmente as que estejam próximas aos clientes. A fim de facilitar esse processo, existem três tipos de servidores de DNS: os servidores raiz, os servidores autoritários e os servidores intermediários.

## 22) Serviço DNS (*Domain Name System*)

---

- Os servidores autoritários contêm a informação original que associa um endereço IP a uma URL. Quando um servidor de DNS intermediário precisa identificar um endereço IP, ele entra em contato com um dos servidores DNS raiz para identificar qual servidor autoritário contém a informação sobre o endereço IP. Com essa informação, o servidor intermediário pode, então, contatar o servidor autoritário e identificar e receber a informação do endereço IP, que vai ser passada para o cliente do servidor de DNS (BOURKE, 2001).

## 22) Serviço DNS (*Domain Name System*)

---

- Com a finalidade de agilizar o processo de consulta, os servidores intermediários de DNS podem manter um conjunto de dados temporários, vindo das consultas anteriores, chamado de cache local. Dessa forma, se o servidor intermediário já contiver a informação, não é necessário contatar um servidor raiz ou um servidor autoritário, aliviando a carga nesses servidores (existe um número muito maior de servidores intermediários). No entanto, essa informação é mantida apenas por certo período de tempo, uma vez que pode haver atualizações nas informações (que vão ocorrer inicialmente nos servidores autoritários).

## 22) Serviço DNS (*Domain Name System*)

---

### I. Afirmativa correta.

- Justificativa. Quando um servidor não é autoritário para um endereço, significa que ele não tem, em seu banco de dados, os registros originais que associam determinado domínio a um endereço IP. Ele pode possuir os dados em cache, caso já tenha sido feita uma pesquisa para dado domínio, mas o conteúdo dessa cache é originado em um servidor remoto. Assim, quando o servidor não é autoritativo para um endereço, ele deve buscar o endereço em outros servidores DNS.



## 22) Serviço DNS (*Domain Name System*)

---

### II. Afirmativa incorreta.

- Justificativa. A adoção do IPv6 vai mudar o tamanho do endereço IP armazenado pelo servidor de DNS, de 32 *bits* para 128 *bits*. Contudo, o servidor de DNS, que fica na camada de aplicação, continua existindo.

## 22) Serviço DNS (*Domain Name System*)

### III. Afirmativa correta.

- Justificativa. O cache de um servidor DNS permite que uma informação que tenha sido obtida por uma consulta prévia seja reaproveitada em consultas similares subsequentes. Dessa forma, consultas similares não necessitam gerar novamente tráfego de rede aos servidores autoritários, além de serem mais rápidas. Consultas em cache podem ficar obsoletas, se a informação armazenada na cache mudar. Por isso, uma das informações armazenadas no banco de dados do servidor DNS é o campo “Time\_to\_live”, que registra em quanto tempo (em segundos) o registro deve ser atualizado. Dessa forma, o servidor de DNS pode saber por quanto tempo a informação armazenada no seu cache permanece válida.

## 22) Serviço DNS (*Domain Name System*)

### IV. Afirmativa correta.

- Justificativa. A internet é uma vasta rede, de alcance mundial e que envolve milhões de máquinas. Se o serviço de DNS fosse centralizado, teríamos uma série de problemas, tanto de desempenho (milhares de computadores no mundo todo acessando uma única central de informações) e de segurança (a estrutura central se tornaria alvo de vários tipos de ataques), quanto de confiabilidade (se o servidor estivesse fora do ar, a rede mundial não funcionaria). Para resolver esses problemas, o serviço de DNS depende de um banco de dados distribuído, ou seja, várias máquinas com autoridade local que dividem a responsabilidade por zonas virtuais; a falha de um servidor de DNS pode afetar no máximo uma pequena fração das máquinas na internet.
- **Alternativa correta: D.**

**ATÉ A PRÓXIMA!**



***UNIP***