

CAMPUS CHARQUEADAS  
**INSTRUÇÕES GERAIS**

- 1 - Este caderno de prova é constituído por 40 (quarenta) questões objetivas.
- 2 - A prova terá duração máxima de 04 (quatro) horas.
- 3 - Para cada questão são apresentadas 04 (quatro) alternativas (a – b – c – d). *APENAS UMA* delas constitui a resposta CORRETA.
- 4 - Após conferir os dados contidos no campo “Identificação do Candidato” no Cartão de Resposta, assine no espaço indicado.
- 5 - As alternativas assinaladas deverão ser transcritas para o Cartão de Resposta, que é o único documento válido para correção eletrônica.
- 6 - Marque o Cartão de Resposta conforme o exemplo abaixo, com caneta esferográfica azul ou preta, de ponta grossa:  


- 7 - Em hipótese alguma haverá substituição do Cartão de Resposta.
- 8 - Não deixe nenhuma questão sem resposta.
- 9 - O preenchimento do Cartão de Resposta deverá ser feito dentro do tempo previsto para esta prova, ou seja, 04 (quatro) horas.
- 10 - Serão anuladas as questões que tiverem mais de uma alternativa marcada, emendas e/ou rasuras.
- 11 - O candidato só poderá retirar-se da sala de prova após transcorrida 01 (uma) hora do seu início.
- 12 - Não é permitido o uso de calculadora.

***BOA PROVA!***



01. Considere a relação "emprestimo" representada abaixo:

cod_emprestimo	cod_cliente	valor
101	231	900
102	364	800
208	540	1500
104	105	20000
405	920	100

Segundo S. Silberschats, H. Korth e S. Sudarshan a operação da álgebra relacional  $\pi_{cod\_emprestimo,valor}(emprestimo)$  intitula-se

- união.
- projeção.
- seleção.
- produto cartesiano.

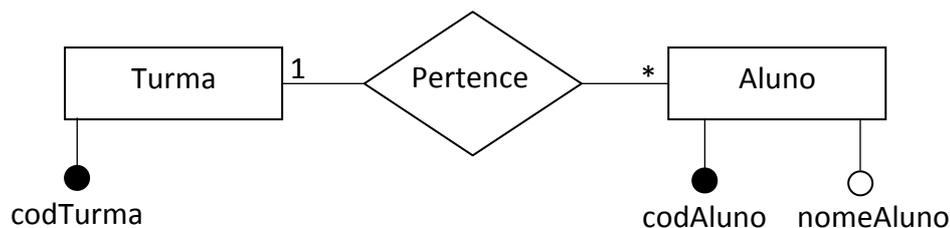
02. A linguagem de definição de dados (DDL) permite a especificação não só de um conjunto de relações, mas também de informações sobre cada relação, tais como:

- O conjunto dos índices a serem mantidos para cada relação.
- A estrutura de armazenamento físico de cada relação no disco.
- As restrições de integridade.

Está(ao) correta(s) a(s) afirmativa(s)

- II apenas.
- I e II apenas.
- I e III apenas.
- I, II e III.

03. Observe o diagrama entidade-relacionamento representado abaixo.



Considerando que as chaves primárias aparecem sublinhadas, qual o conjunto de relações que representam as tabelas estritamente necessárias para implementá-lo?

- Turma(codTurma)  
Aluno(codAluno, nomeAluno)
- Turma(codTurma, cod\_aluno, nomeAluno)  
Aluno(codAluno, nomeAluno)
- Turma(codTurma)  
Aluno(codAluno, nomeAluno, codTurma)
- Turma(codTurma, cod\_aluno)  
Aluno(codAluno, nomeAluno)

04. Considere as seguintes tabelas em um banco de dados relacional:

**produtos**

codproduto	nomeproduto	preco
100	bicicleta	200
200	escada	50
300	cadeira	100
400	mesa	400

**clientes**

codcliente	nomecliente	unidadefederativa
10	Joao	SP
20	Maria	RS
30	Luiza	RS

**compra**

codpedido	codcliente	codproduto	quantidade
1000	10	300	4
2000	20	100	2
3000	20	300	1
4000	10	400	3
5000	20	400	4
6000	30	300	1

A partir dessas tabelas, qual a consulta SQL correta para gerar o resultado abaixo?

cliente	uf	t
Joao	SP	1600
Maria	RS	2100

- a) SELECT clientes.nomecliente AS cliente, clientes.unidadefederativa AS uf, sum(produtos.preco \* compra.quantidade) as t FROM produtos, compra, clientes WHERE produtos.codproduto=compra.codproduto AND clientes.codcliente=compra.codcliente and T > 1000 GROUP BY clientes.nomecliente, clientes.unidadefederativa ORDER BY T
- b) SELECT clientes.nomecliente AS cliente, clientes.unidadefederativa AS uf, sum(produtos.preco \* compra.quantidade) as t FROM produtos, compra, clientes WHERE produtos.codproduto=compra.codproduto AND clientes.codcliente=compra.codcliente GROUP BY clientes.nomecliente, clientes.unidadefederativa HAVING SUM(produtos.preco \* compra.quantidade) > 1000 ORDER BY T
- c) SELECT clientes.nomecliente AS cliente, clientes.unidadefederativa AS uf, sum(produtos.preco \* compra.quantidade) as t FROM produtos, compra, clientes WHERE produtos.codproduto=compra.codproduto AND clientes.codcliente=compra.codcliente and T > 1000 GROUP BY clientes.nomecliente, clientes.unidadefederativa ORDER BY T DESC
- d) SELECT clientes.nomecliente AS cliente, clientes.unidadefederativa AS uf, sum(produtos.preco \* compra.quantidade) as t FROM produtos, compra, clientes WHERE produtos.codproduto=compra.codproduto AND clientes.codcliente=compra.codcliente GROUP BY clientes.nomecliente, clientes.unidadefederativa HAVING SUM(produtos.preco \* compra.quantidade) > 1000 ORDER BY T DESC

**05.** Dizemos que uma relação  $R$  está na primeira forma normal (1FN), quando

- a) os domínios de todos os atributos de  $R$  são atômicos.
- b) apenas o domínio da chave primária é atômico.
- c) pelo menos os elementos que não são chave são atômicos.
- d) um conjunto de atributos pode ser definido como chave primária.

**06.** Com relação às funções agregadas do SQL, é correto afirmar que

- I. o SQL não permite o uso de *distinct* com o *count(\*)*.
- II. é obrigatório o uso da cláusula *group by* quando do uso da função *avg*.
- III. tomam uma coleção de valores e retornam um único valor.

Está(ao) correta(s) apenas as afirmativa(s)

- a) I e II apenas.
- b) II e III apenas.
- c) III apenas.
- d) I, II e III.

**07.** Considere que as relações abaixo foram criadas com o uso da linguagem SQL em um banco de dados relacional.

```
CREATE TABLE fornecedor (  
    codfornec          INTEGER NOT NULL,  
    nomefornec        VARCHAR(50) NOT NULL,  
    PRIMARY KEY (codfornec)  
);  
CREATE TABLE remessa (  
    codremessa        INTEGER NOT NULL,  
    data              DATE NOT NULL,  
    codfornec         INTEGER NOT NULL,  
    PRIMARY KEY (codremessa)  
);  
  
ALTER TABLE remessa ADD constraint "fornec_fkey" FOREIGN KEY  
(codfornec) REFERENCES fornecedor ON UPDATE CASCADE;
```

Sobre o uso da linguagem SQL em um banco de dados relacional, é **INCORRETO** afirmar que

- a) a relação "remessa" possui uma chave estrangeira que referencia a relação "fornecedor", especificamente o atributo "codfornec".
- b) a existência da chave estrangeira "codfornec" na relação "remessa" não permite que operações de delete sejam realizadas sobre fornecedores que possuem entradas na relação "remessa".
- c) quando o atributo "codfornec" da relação "fornecedor" sofrer um update, essa atualização será automaticamente refletida na relação "remessa".
- d) quando o atributo "codfornec" da relação "fornecedor" sofrer um update, essa atualização irá gerar um problema de integridade referencial.

**08.** Segundo C. J. Date, as transações em um banco de dados devem possuir quatro propriedades principais, que são chamadas de ACID (Atomicidade, Correção, Isolamento e Durabilidade).

Considerando essas quatro propriedades, qual definição abaixo está correta?

- a) A propriedade de Correção indica que as transações transformam um estado correto do banco de dados em outro estado correto sem, necessariamente, preservar a correção em todos os pontos intermediários.
- b) A propriedade de Durabilidade indica que as transações são invisíveis umas às outras, ou seja, as atualizações de qualquer transação dada são ocultas de todas as outras até o COMMIT dessa transação.
- c) A propriedade Isolamento se refere ao fato de que, uma vez completado o COMMIT de uma transação, as informações sobrevivem no banco de dados mesmo que haja uma queda subsequente do sistema.
- d) A propriedade de Atomicidade prevê que todas as relações em um banco de dados relacional tenham uma ou mais tuplas que definam uma chave primária.

**09.** Considere a seguinte definição de dados descrita com o uso da linguagem SQL em um banco de dados relacional:

```
CREATE TABLE agencia (  
    nome_agencia      CHAR(15),  
    cidade_agencia    CHAR(30),  
    ativo             NUMERIC(16,2),  
    PRIMARY KEY (nome_agencia)  
);
```

Qual das consultas encontra os nomes de todas as agências que têm ativos maiores do que pelo menos uma agência localizada em Charqueadas?

- a) `SELECT nome_agencia FROM agencia WHERE ativo > ALL (SELECT ativo FROM agencia WHERE cidade_agencia='Charqueadas')`
- b) `SELECT nome_agencia FROM agencia WHERE ativo > SOME (SELECT ativo FROM agencia WHERE cidade_agencia='Charqueadas')`
- c) `SELECT DISTINCT nome_agencia FROM agencia WHERE agencia.ativo > agencia.ativo AND agencia.cidade_agencia='Charqueadas';`
- d) `SELECT nome_agencia FROM agencia WHERE agencia.ativo > agencia.ativo AND agencia.cidade_agencia='Charqueadas';`

**10.** Considere o código em linguagem PHP apresentado abaixo:

```
1. <?php
2. function mais()
3.     {
4.         global $total;
5.         $total++;
6.     }
7. function menos()
8.     {
9.         static $val = 100;
10.        $val-= 10;
11.        return $val;
12.    }
13. $total = 100;
14. $r = 0;
15. for($i = 10; $i > 0; $i--)
16.     $r+= $i % 2 ? mais() : menos();
17. echo $r."-".$total;
18. ?>
```

O que será impresso pelo comando *echo*, da linha 17, do código-fonte acima?

- a) 450-100
- b) 450-105
- c) 350-100
- d) 350-105

**11.** Considere o código em linguagem PHP apresentado abaixo:

```
<?php
$arr[] = 10;
$arr[] = 20;
$arr["100"] = 30;
$arr[] = 40;
?>
```

No código acima, quais serão os índices do vetor *\$arr*, respectivamente?

- a) 0, 1, 100, 2
- b) 0, 1, 100, 101
- c) 1, 2, 100, 101
- d) 1, 2, 100, 3

**12.** Considere o código em linguagem PHP apresentado abaixo:

1. <?php
2. \$a = 100;
3. \$b = "200 + 200";
4. \$a+= \$b;
5. echo "\$a | \$b";
6. ?>

O que será impresso pelo comando *echo*, da linha 5, no código-fonte acima?

- a) 300 | 200 + 200
- b) 100 | 200 + 200
- c) \$a | \$b
- d) 100200 + 200 | 200 + 200

**13.** Qual a função exercida pelas aspas invertidas ( ` ` ) na linguagem PHP?

- a) Delimitar cadeias de dados alfanuméricos.
- b) Ser uma das formas para comentar trechos de código.
- c) Ser um caractere de escape para integração com linguagem HTML.
- d) Executar chamadas ao sistema operacional.

**14.** Quanto aos comandos *include* e *require* na linguagem PHP é **INCORRETO** afirmar que

- a) a instrução *require* permite a inclusão de apenas um arquivo.
- b) o comando *include*, quando utilizado com comandos condicionais, deve sempre aparecer entre chaves, pois geralmente uma *include* executa mais de um comando.
- c) uma das funcionalidades da instrução *require* é a possibilidade de incluir arquivos dinamicamente.
- d) o comando *include* é reavaliado a cada chamada, ou seja, uma mesma instrução *include* pode incluir vários arquivos.

**15.** Considerando os métodos de submissão de formulários GET e POST, é correto afirmar que o método

- a) POST exibe toda a informação submetida na URL.
- b) POST não possui limitação quanto à quantidade de dados a ser submetida.
- c) GET envia os dados no conteúdo das informações HTTP.
- d) GET não permite o envio de informações por meio de *links*.

**16.** Considere o código em linguagem PHP apresentado abaixo:

1. <?php
2. \$x = 10;
3. \$y = 20;
4. \$str = "\$x papagaios|";
5. \$str.= '\$y gatos|';
6. \$str = \$str.(\$x + \$y)." cachorros";
7. echo \$str;
8. ?>

O que será impresso pelo comando *echo*, da linha 7, no código-fonte acima?

- a) \$x papagaios|\$y gatos|30 cachorros
- b) 10 papagaios|20 gatos|30 cachorros
- c) 10 papagaios|\$y gatos|30 cachorros
- d) 10 papagaios|\$y gatos|1020 cachorros

**17.** O PHP 5 introduz o conceito de interceptação em operações realizadas por objetos. Nesse contexto, podemos citar, corretamente, como exemplos de métodos de interceptação (*magic methods*):

- a) \_\_set(), \_\_get() e \_\_call()
- b) \_\_set(), \_\_get() e \_\_show()
- c) \_\_write(), \_\_read() e \_\_call()
- d) \_\_write(), \_\_read() e \_\_show()

**18.** Considerando o código em linguagem PHP apresentado abaixo:

```

1. <?php
2. class Pessoa {
3.     private $nome;
4.     static $qtd = 0;
5.     function __construct($n)
6.     {
7.         $this->nome = $n;
8.         _____;
9.     }
10. }
11. $p1 = new Pessoa("Joao");
12. $p2 = new Pessoa("Maria");
13. $p3 = new Pessoa("Pedro");
14. echo "Quantidade total de pessoas = "._____;
15. ?>

```

As duas instruções necessárias para permitir

- incrementar em 1 (um) a propriedade "static \$qtd" no interior do método construtor, na linha 8 em destaque (área sublinhada) e
- apresentar a quantidade total de objetos Pessoa, acessando a propriedade "static \$qtd", na linha 14 em destaque (área sublinhada).

São, respectivamente,

- a) this::\$qtd++ e \_\_Pessoa::\$qtd
- b) self::\$qtd++ e \_\_Pessoa::\$qtd
- c) self::\$qtd++ e Pessoa::\$qtd
- d) this::\$qtd++ e Pessoa::\$qtd

**19.** Em se tratando do uso da JQuery para manipular requisições Ajax, qual atributo HTML deve ser adicionado as *tags* dos elementos de formulário, para que os valores destes elementos sejam serializados, a partir do método `serialize()`?

- a) id.
- b) serialize.
- c) name.
- d) class.

**20.** Qual evento Ajax é associado apenas ao término de uma requisição bem sucedida?

- a) ajaxComplete.
- b) ajaxSuccess.
- c) ajaxStatus.
- d) ajaxStop.

**21.** A função `$.ajax` carrega uma página remota usando uma requisição HTTP.

Qual tipo de requisição **NÃO** é válida?

- a) GET.
- b) POST.
- c) GETSCRIPT.
- d) GETHTML.

**22.** Em requisições AJAX com o uso da JQuery, qual o método que anula a ação do método `live()`?

- a) `stop()`.
- b) `close()`.
- c) `kill()`.
- d) `dead()`.

**23.** Considere a linha de código na linguagem JavaScript abaixo.

```
document.write("<p>"+6+9+"</p>");
```

Qual é o código HTML gerado após sua execução?

- a) 15.
- b) `<p>69</p>`.
- c) `<p>15</p>`.
- d) Ocorreria um erro durante a execução da linha, não apresentando resultado.

**24.** Considerando o código JavaScript apresentado abaixo.

```
1. <script type="text/javascript" language="JavaScript">
2.     var x;
3.     alert(x);
4. </script>
```

Qual seria a informação apresentada pelo `alert` da linha 3?

- a) null
- b) undefined
- c) 0
- d) ""

**25.** Na linguagem JavaScript, qual o escopo de uma variável cuja declaração foi omitida?

- a) Global.
- b) Local.
- c) De bloco.
- d) Não é possível omitir a declaração de uma variável JavaScript.

**26.** Observe o seguinte trecho de código na linguagem JavaScript:

```
<script type="text/javascript" language="JavaScript">
  var x="3";
  var y=2;
  var z=1;
  y+=z;
  if(x<y)
    alert("A");
  else if (x==y)
    alert("B");
  else
    alert("C");
</script>
```

Esse código apresentará um `alert` com conteúdo igual a

- a) A
- b) B
- c) C
- d) Não apresentará `alert`.

**27.** Suponha-se o seguinte código na linguagem HTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>QUESTAO</title>
</head>
<body>
  <form name="formulario">
    <div>
      <span>NOME:</span>
      <input type="text" name="nome" size="15"/>
    </div>
    <div>
      <span>SENHA:</span>
      <input type="password" name="senha" size="15"/>
    </div>
  </form>
</body>
</html>
```

Para acessar o campo **<input type="password" name="senha" size="15"/>** do formulário, a partir do uso do objeto *Form* da linguagem JavaScript, é válida a instrução:

- a) document.forms.fields[1]
- b) document.forms[0].fields[1]
- c) document.forms.elements[1]
- d) document.forms[0].elements[1]

**28.** Qual o símbolo utilizado na linguagem CSS para representação de um seletor universal?

- a) &
- b) \*
- c) #
- d) .

29. Considere o seguinte código-fonte de uma página web:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>QUESTAO</title>
  <style type="text/css">
    h1{
      background-color: #FF0000;
      color:#FFFFFF;
    }
    p{
      background-color: #0000FF;
      color:#FFFFFF;
    }
  </style>
  <script type="text/javascript">
function carregado(){
  x=document.getElementById('titulo');
  x=document.getElementById('texto');
  x.style.backgroundColor="#00FF00";
}
</script>
</head>
<body onload="carregado()">
  <h1 id="titulo">Minha P&aaacute;gina</h1>
  <p id="texto">Conte&uacute;do</p>
</body>
</html>
```

Após a carga da página acima e as respectivas execuções de código JavaScript associados a ela, teremos a seguinte formatação de cor de fundo para o elemento **<h1 id="titulo">Minha P&aaacute;gina</h1>** e para o elemento **<p id="texto">Conte&uacute;do</p>**, respectivamente,

- a) #FF0000 e #0000FF
- b) #FFFFFF e #FFFFFF
- c) #00FF00 e #00FF00
- d) #FF0000 e #00FF00

**30.** No seguinte trecho de código de uma folha de estilos CSS:

```
#x
{
color:#FF0000;
}

.y
{
color:#00FF00;
}

p
{
color:#0000FF;
}
```

Caso a página HTML associada a essa folha de estilos possua um elemento

```
<p class="y" id="x" >texto</p>
```

Esse elemento terá cor de fonte igual a

- a) #FF0000
- b) #00FF00
- c) #0000FF
- d) #000000

**31.** Com respeito ao uso de DOCTYPE na linguagem XHTML, qual destas declarações **NÃO** admite elementos que, segundo a recomendação do W3C, estejam em desuso?

- a) <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
- b) <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
- c) <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
- d) A declaração do DOCTYPE não abrange restrições a elementos que, de acordo com W3C, estejam em desuso.

**32.** Quando montamos formulários com a linguagem HTML, qual será o atributo *type* indicado, caso seja necessário criar um campo *input* onde os caracteres digitados não são legíveis?

- a) <input type="text"/>
- b) <input type="hidden"/>
- c) <input type="submit"/>
- d) <input type="password"/>

33. Considere o código na linguagem HTML abaixo:

```

1. <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
2. "http://www.w3.org/TR/html4/strict.dtd">
3. <html>
4.     <head>
5.         <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
6.         <title>QUESTAO</title>
7.         <style type="text/css">
8.             div.meudiv > p[class~=num]{
9.                 border: solid #000000 1px;
10.            }
11.        </style>
12.    </head>
13.    <body>
14.        <div class="meudiv">
15.            <p class="num" id="x">A</p>
16.            <p class="enum" id="num">B</p>
17.            <p class="numero" id="num">C</p>
18.                <div class="numero">
19.                    <p class="num" id="num">D</p>
20.                </div>
21.        </div>
22.    </body>
23. </html>

```

Qual dos elementos HTML **NÃO** será apresentado com borda?

- a) <p class="num" id="x">A</p> (Linha 15)
- b) <p class="enum" id="num">B</p> (Linha 16)
- c) <p class="numero" id="num">C</p> (Linha 17)
- d) <p class="num" id="num">D</p> (Linha 19)

34. Considerando a linguagem XHTML, qual das estruturas de *tags* abaixo é válida?

- a) <DIV><P class="paragrafo" >A<BR /></P></DIV>
- b) <div><p class="paragrafo" >B<br></p></div>
- c) <div><p class="paragrafo" >C<br></br></p></div>
- d) <div><p class=paragrafo>D</p></div>

35. Qual será a *meta tag* utilizada para fazer com que o navegador permaneça 5 (cinco) segundos na página e depois seja redirecionado para a página do Instituto Federal Sul-Rio-Grandense (IFSul) (<http://www.ifsul.edu.br>)?

- a) <meta http-equiv="refresh" content="5" url="http://www.ifsul.edu.br">
- b) <meta refresh="5" url="http://www.ifsul.edu.br">
- c) <meta http-equiv="refresh" content="5;url=http://www.ifsul.edu.br">
- d) <meta http-equiv="http://www.ifsul.edu.br" content="5">

36. Para modificar o posicionamento de um elemento HTML, deslocando-o a partir de sua posição inicial em 50px para baixo, qual classe definida com a linguagem CSS está correta?

- a) .a{position: absolute; top:50px;}
- b) .b{position: absolute; bottom:50px;}
- c) .c{position: relative; top:50px;}
- d) .d{position: relative; bottom:50px;}

- 37.** A arquitetura genérica do J2ME prevê quatro camadas. Partindo da camada superior até a inferior temos, respectivamente, as camadas
- Perfil, Configuração, Máquina Virtual Java, Sistema Operacional Hospedeiro.
  - Configuração, Perfil, Máquina Virtual Java, Sistema Operacional Hospedeiro.
  - Perfil MID, Configuração, Sistema Operacional Hospedeiro, Máquina Virtual K.
  - Configuração, Perfil MID, Sistema Operacional Hospedeiro, Máquina Virtual K.
- 38.** Existem três aspectos principais que distinguem a linguagem Java para CLDC (*Connected, Limited Device Configuration*) em relação ao definido na Especificação da Linguagem Java. Qual aspecto **NÃO** se apresenta como uma dessas distinções?
- Suporte a ponto flutuante.
  - Finalização.
  - Tratamento de erros.
  - Herança.
- 39.** Quais são os quatro componentes básicos de uma aplicação, segundo a definição da arquitetura *Android*?
- Profiles* (Perfis), *Services* (Serviços), *Broadcast and Intent Receivers* (Receptores de Transmissão), *Content Providers* (Fornecedores de Conteúdo)
  - Activities* (Atividades), *Services* (Serviços), *Broadcast and Intent Receivers* (Receptores de Transmissão), *Content Providers* (Fornecedores de Conteúdo)
  - Activities* (Atividades), *Lifecycle* (Ciclo de Vida), *Broadcast and Intent Receivers* (Receptores de Transmissão), *Content Providers* (Fornecedores de Conteúdo)
  - Profiles* (Perfis), *Services* (Serviços), *Receivers* (Receptores), *Providers* (Provedores)
- 40.** Qual o arquivo XML que especifica as funcionalidades e requisitos de uma aplicação *Android*?
- AndroidSettings.xml
  - AndroidMain.xml
  - AndroidManifest.xml
  - AndroidIni.xml