



**GOVERNO DO DISTRITO FEDERAL**

**GDF SECRETARIA DE ESTADO DE PLANEJAMENTO E GESTÃO**

CONCURSO PÚBLICO

**PROFESSOR DE EDUCAÇÃO BÁSICA**

**CARGO 9**

Aplicação: 16/11/2008

# CADERNO DE PROVAS – PARTE II

## CONHECIMENTOS ESPECÍFICOS

### ÁREA 1 COMPONENTE CURRICULAR: INFORMÁTICA

#### **ATENÇÃO!**

- » Leia atentamente as instruções constantes na capa da Parte I do seu caderno de provas.
- » Nesta parte do seu caderno de provas, que contém os itens relativos à prova objetiva de **Conhecimentos Específicos**, confira o número do seu cargo, o número de sua área e o nome do seu componente curricular transcritos acima e no rodapé de cada página numerada desta parte do caderno de provas.

#### **AGENDA (datas prováveis)**

- I **18/11/2008**, após as 19h (horário de Brasília) – Gabaritos oficiais preliminares das provas objetivas: Internet — [www.cespe.unb.br](http://www.cespe.unb.br).
- II **19 a 21/11/2008** – Recursos (provas objetivas): exclusivamente no Sistema Eletrônico de Interposição de Recurso, Internet, mediante instruções e formulários que estarão disponíveis nesse sistema.
- III **16/12/2008** – Resultado final das provas objetivas e convocação para a entrega da documentação para a avaliação de títulos: Diário Oficial do Distrito Federal e Internet.
- IV **17 a 19/12/2008** – Entrega da documentação para a avaliação de títulos, em locais e horários a serem informados na respectiva convocação.

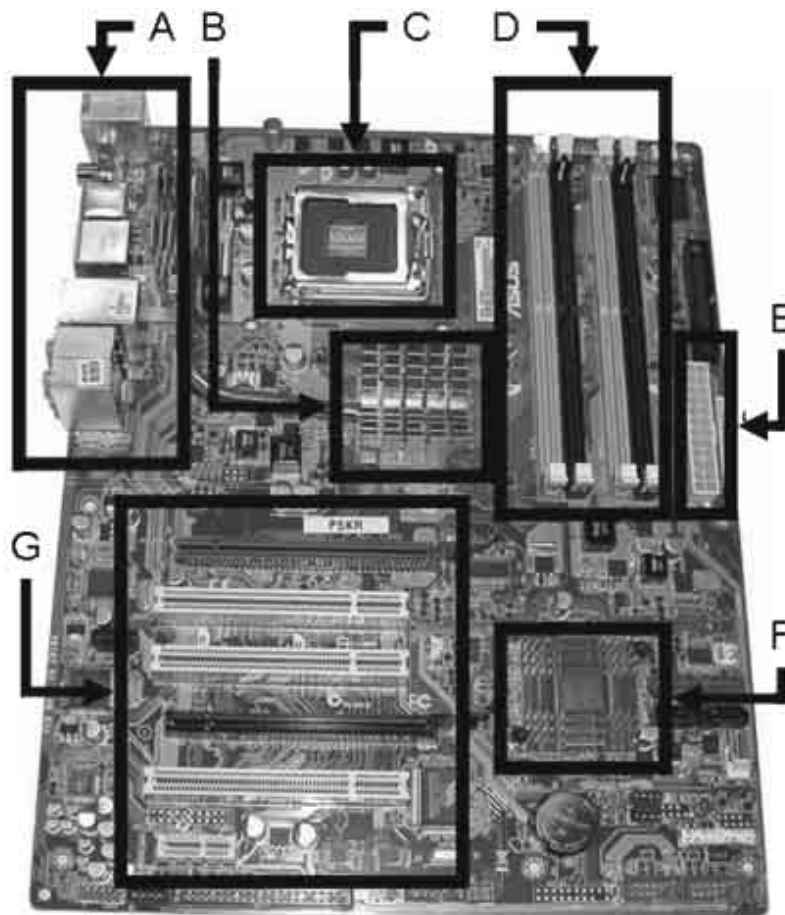
#### **OBSERVAÇÕES**

- Não serão objeto de conhecimento recursos em desacordo com o item 10 do Edital n.º 1 - SEPLAG/PROF, de 15/9/2008.
- Informações adicionais: telefone 0(XX) 61 3448-0100; Internet – [www.cespe.unb.br](http://www.cespe.unb.br).
- É permitida a reprodução deste material apenas para fins didáticos, desde que citada a fonte.

De acordo com o comando a que cada um dos itens de 51 a 120 se refira, marque, na **folha de respostas**, para cada item: o campo designado com o código **C**, caso julgue o item **CERTO**; ou o campo designado com o código **E**, caso julgue o item **ERRADO**. A ausência de marcação ou a marcação de ambos os campos não serão apenadas, ou seja, não receberão pontuação negativa. Para as devidas marcações, use a **folha de respostas**, único documento válido para a correção das suas provas.

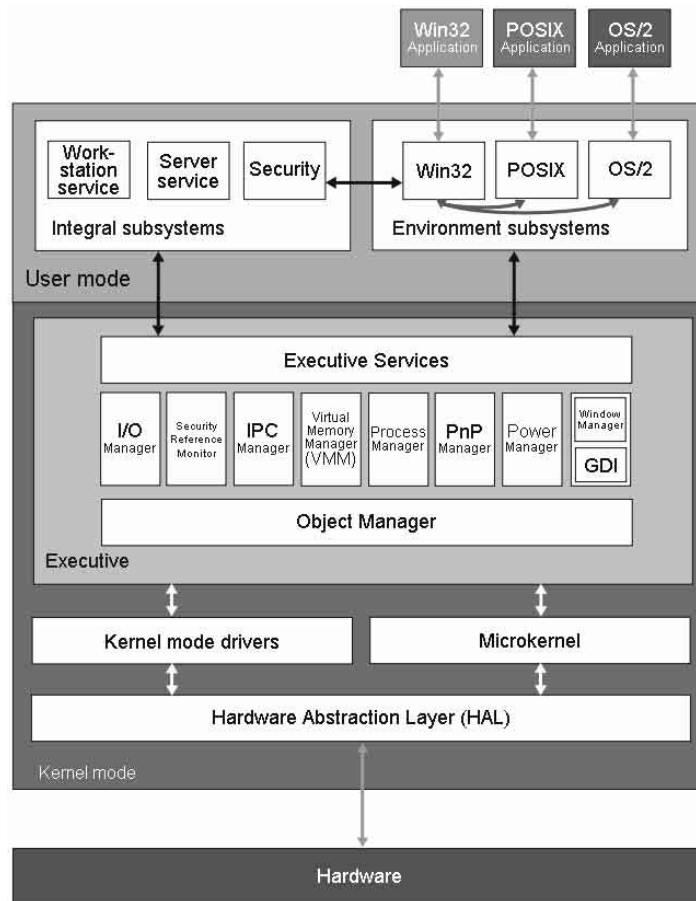
## CONHECIMENTOS ESPECÍFICOS

#	característica
1	LGA775 CPU Platform (Support for Core 2 Duo/Quad/Extreme, Pentium D, Pentium 4 CPUs)
2	800/1066/1333MHz CPU Front Side Bus Support
3	Intel P35 Northbridge/ICH9R Southbridge Chipset
4	DDR2 667/800/1066MHz Dual Channel Memory – 4 x Slots, 8GB Max Total
5	6 x SATA-II Ports (1 x External/eSATA)
6	12 x USB2.0 Ports (6 on Back Panel, 6 by header on motherboard)
7	2 x PCI-Express x16 Slots (One Supports x16, other supports x4 or x1 modes) – Crossfire Compatible
8	1 x PCI-Express x1 Slot, 3 x PCI Slots
9	2 x Firewire400 Ports (1 x Rear Panel, 1 x Motherboard header)
10	8-Channel Audio w. S/PDIF output.
11	1 x GigLAN Port
12	1 x ATA-133 Connector, 1 x FDD Connector, 1 x Serial Port Header, 3 x Aux Fan Connectors
13	12" x 9.6" ATX Form Factor



A figura acima ilustra a placa mãe de um computador, em que foram identificados elementos pelas letras de A a G. A tabela acima apresenta algumas características dessa placa mãe, indicadas pelos números de 1 a 13. Julgue os próximos itens, relativos às informações apresentadas.

- 51 Entre as características listadas na tabela, a mais relacionada com o aspecto físico geral da placa mãe ilustrada é a de número 13.
- 52 A região da placa indicada por E tenderá a permanecer em temperatura superior à da região indicada por C, quando o computador estiver em funcionamento.
- 53 Os *chipsets* da placa estão localizados nas regiões indicadas por B e F, sendo que o *chip* da região B controla barramentos de menor vazão que o da região F.
- 54 Na região A, encontram-se 6 saídas USB de alta velocidade.
- 55 A região G contém 6 *slots* PCI ou PCI PCI Express, os quais possuem velocidade inferior à dos *slots* da região D.



Internet: <upload.wikipedia.org>.

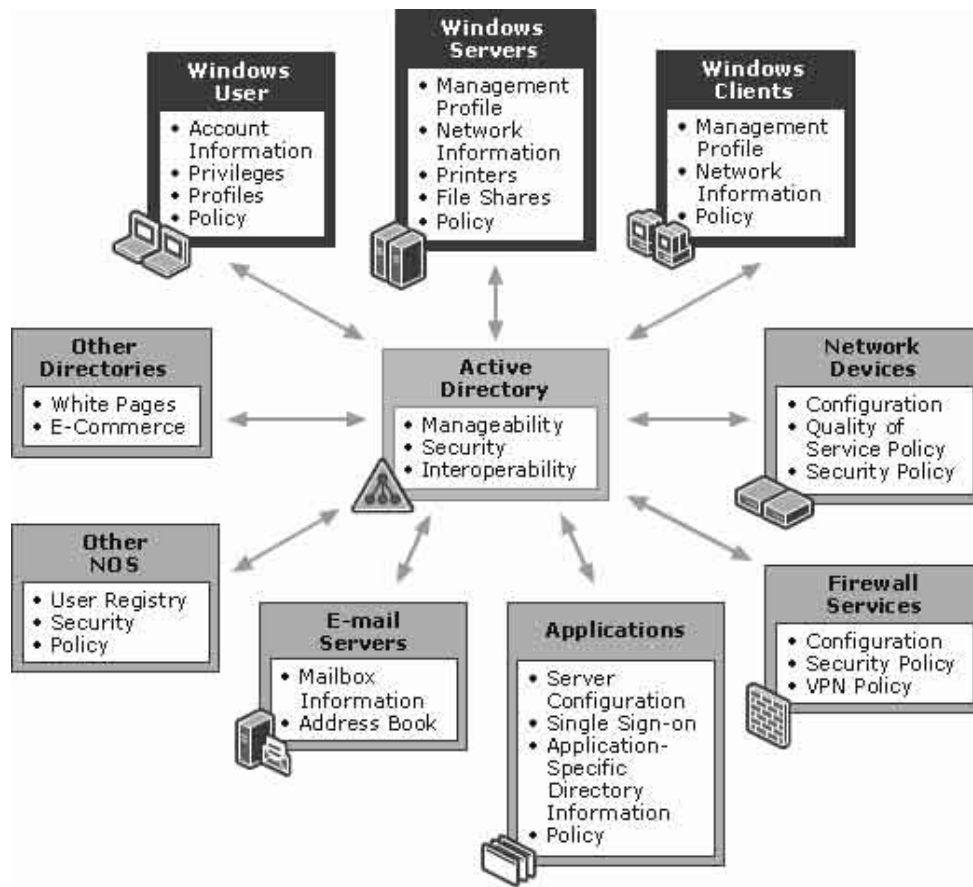
A figura acima apresenta um esquema arquitetural da família de sistemas operacionais Windows NT. Julgue os itens que se seguem acerca das informações apresentadas nesse esquema.

- 56 No módulo executivo do Windows NT realiza-se: gerenciamento da comunicação entre processos; escalonamento de processos; carga dos *drivers* de dispositivos no interior do núcleo do sistema operacional; gerenciamento das operações de entrada e de saída de dados.
- 57 Uma política FIFO de escalonamento de processos em sistemas operacionais é mais justa na atribuição de tempo a processos que uma política LIFO.

Quotas
Arquivos Esparsos ( <i>Sparse files</i> )
Pontos de Montagem de Volume ( <i>Volume mount points</i> )
<i>Hard links</i>
Gerenciamento Hierárquico de Armazenamento ( <i>Hierarchical Storage Management - HSM</i> )
Armazenamento Estruturado Nativo ( <i>Native Structured Storage - NSS</i> )
Cópia Sombra ( <i>Volume Shadow Copy</i> )
Compressão de Arquivos ( <i>File compression</i> )
Criptografia ( <i>Encrypting File System - EFS</i> )
<i>Symbolic links</i>
Sistema de Arquivos Transacional ( <i>Transactional NTFS</i> )

A tabela acima apresenta características do NTFS, em ambiente Windows. Julgue os itens a seguir, acerca dessas características.

- 58 Do ponto de vista do usuário de um sistema de arquivos, *hard links* são funcionalmente similares a *symbolic links*, sendo uma das diferenças o fato de que os primeiros fazem referências cruzadas entre arquivos e diretórios embasadas nos nomes desses arquivos.
- 59 A característica de cópia sombra, disponível no NTFS, reduz a chance de conflitos durante a realização de *backups* com utilitários do sistema operacional Windows, como o *ntbackup*.
- 60 O uso de criptografia no sistema NTFS é embasado em criptografia de chave pública e não, de chave privada; um de seus efeitos é diminuir a entropia de um arquivo, diferentemente do uso de compressão, que aumenta a entropia de um arquivo.



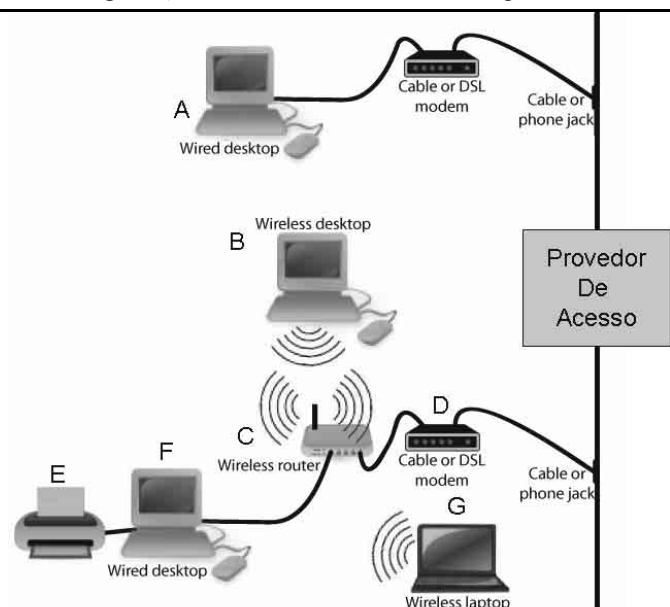
Internet: <wikipedia.org>.

A figura acima apresenta características da tecnologia Active Directory, disponível em plataforma Windows. Julgue os itens seguintes, acerca dessas características.

- 61 A tecnologia Active Directory guarda similaridades funcionais com a tecnologia LDAP, como uso de domínios hierarquizados, controle de acesso a diretórios e registro de usuários.
- 62 A tecnologia Active Directory permite a definição de políticas de segurança relacionadas ao uso de redes privadas virtuais.

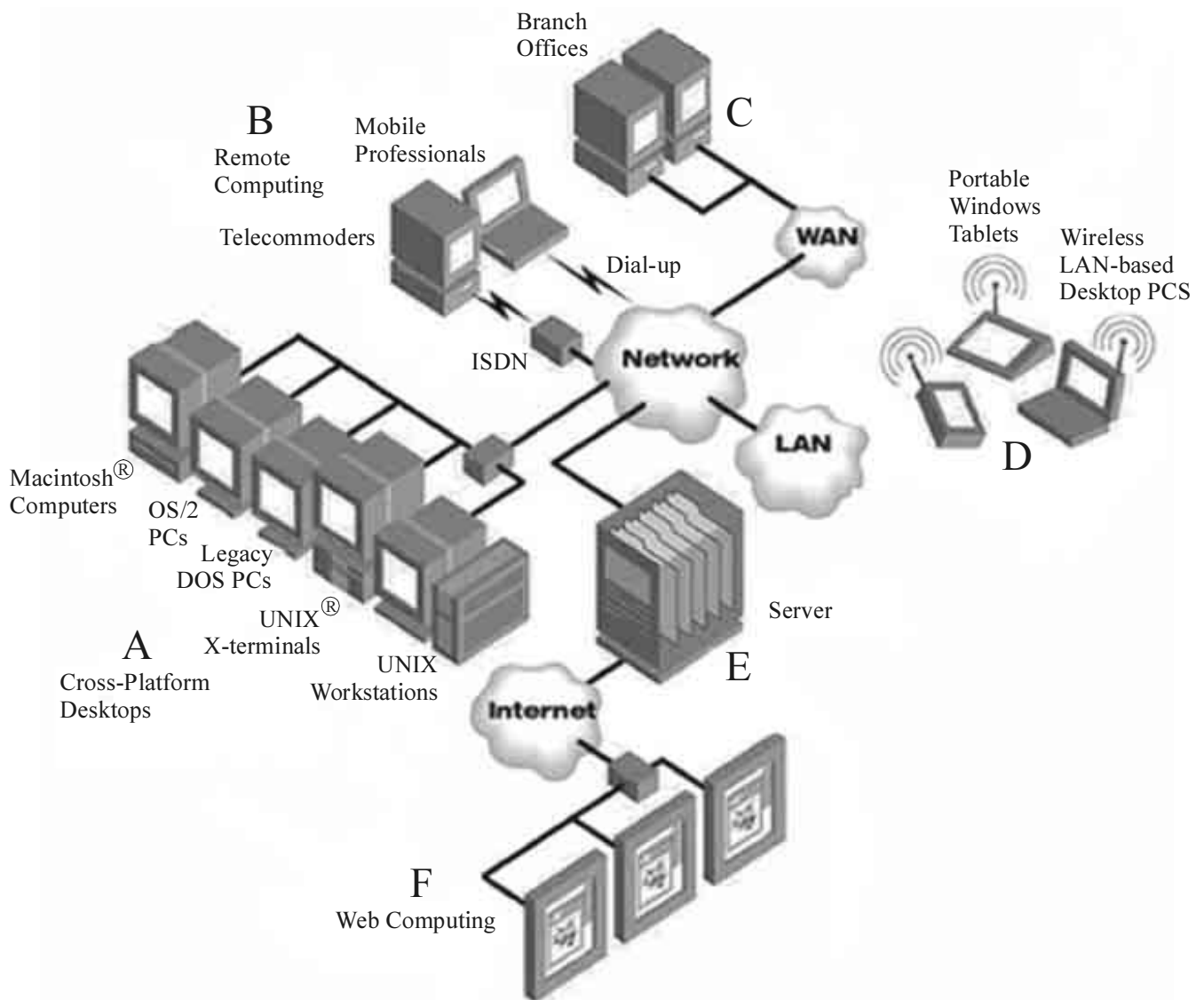
A figura ao lado apresenta elementos de redes de computadores, sendo destacados elementos indicados pelas letras de A a G. Julgue os itens subsequentes, acerca das informações apresentadas e das características de redes de computadores.

- 63 O uso da tecnologia e dos protocolos de comunicação Bluetooth é mais relevante para dispositivos do tipo G que para dispositivos do tipo A.
- 64 Os protocolos WEP e WPA podem ser empregados na comunicação entre os dispositivos F e C, mas não, entre os dispositivos B e C. Além disto, preferência deve ser dada ao uso do protocolo WPA, em detrimento do protocolo WEP, devido ao maior número de vulnerabilidades de segurança presentes neste último.
- 65 O uso do protocolo NAT no elemento indicado por C exerce mais influência sobre as informações trocadas entre B e G que entre A e F.
- 66 Considere que haja necessidade de tornar disponível, através da rede, os serviços do dispositivo E. Nesse caso, o *gateway* de comunicação que atua no nível de aplicação e que permite o uso de E por A, por exemplo, será o dispositivo F e não, o dispositivo C.



Internet: <computer-help-center.com>.

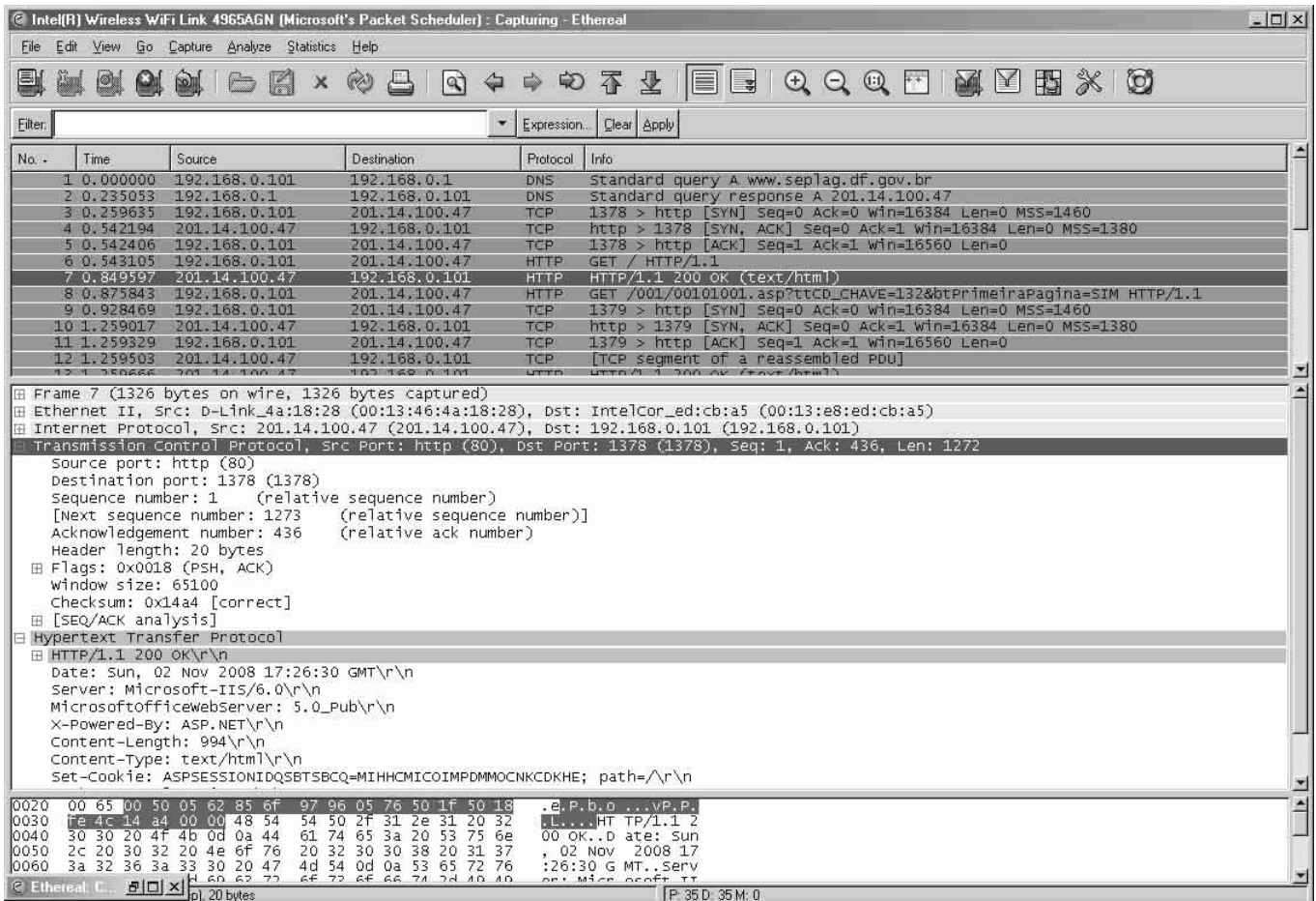
- 67 O equipamento que melhor desempenha o papel de *switch*, na configuração de rede local apresentada, é o indicado por C.



Internet: <www.mbidata.com>

A figura acima apresenta a arquitetura de uma rede corporativa, sendo destacados os elementos indicados pelas letras de A a F. Julgue os itens que se seguem, relativos às informações apresentadas, no contexto de redes de computadores.

- 68** Protocolos de roteamento de borda são usados em dispositivos intermediários que formam o canal de comunicação entre os dispositivos A e B, mas não, necessariamente, em dispositivos intermediários que formam o canal entre E e F.
- 69** A instalação de uma zona desmilitarizada (DMZ) no ambiente apresentado será mais adequadamente efetuada junto ao conjunto de dispositivos E.
- 70** Os dispositivos indicados por D devem estar se comunicando por meio da nuvem denominada WAN (*wireless access network*).

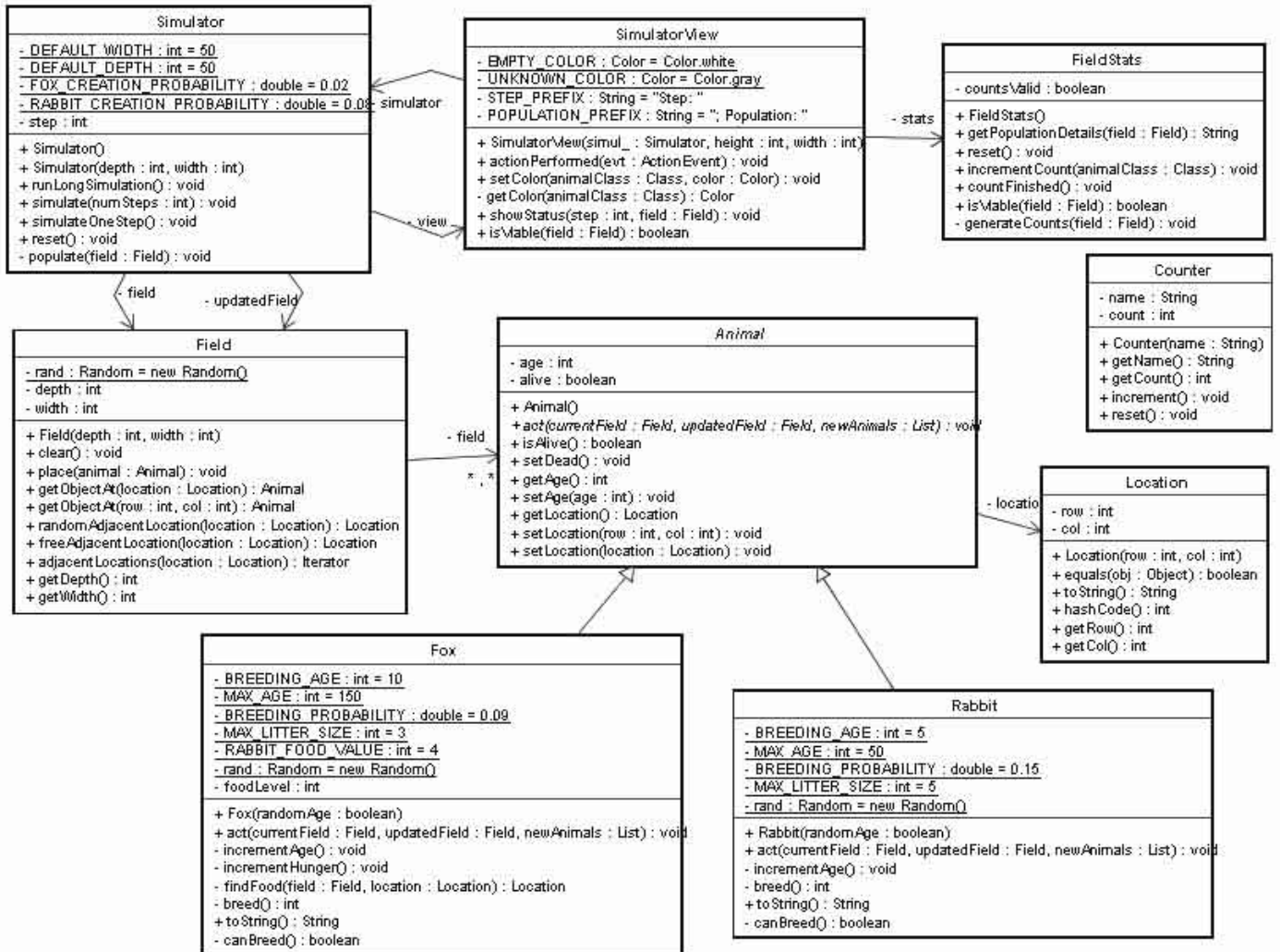


A figura acima apresenta a interface gráfica de um aplicativo de monitoramento de rede, sendo detalhadas informações trocadas entre computadores. Julgue os próximos itens, relativos às informações apresentadas e a conceitos de redes de computadores.

- 71 As informações sugerem que o computador local de onde foram coletadas estas informações utilizou um *browser web*; que o *gateway* de rede do computador local possui o endereço IP igual a 192.168.0.101; e que o endereço IP do servidor http cujo nome de domínio é `www.seplag.df.gov.br` é 201.14.100.47.
- 72 Os endereços MAC dos adaptadores de rede dos computadores cujos endereços IP são 201.14.100.47 e 192.168.0.101 são, respectivamente, 00:13:46:4a:18:28 e 00:13:e8:Ed:cb:a5.
- 73 É correto concluir, a partir das informações apresentadas, que o envio e o recebimento de pedidos http por um *browser web* que funciona em um computador *desktop* empregam o protocolo de transporte TCP e não, o protocolo de transporte UDP.
- 74 As informações apresentadas permitem concluir que, no dia 2 de novembro de 2008, aproximadamente às 17 h e 26 min, do fuso horário GMT (Greenwich *mean time*), trafegou, em um único *frame* da rede, uma página HTML, com tamanho de 994 bytes, a qual foi gerada por um servidor no qual era executado o *software* Microsoft IIS, que é capaz de interpretar páginas na tecnologia ASP.NET.
- 75 É correto afirmar, a partir das informações apresentadas, que o estabelecimento de conexões via *sockets* feitas entre a porta http *default* de uma máquina e a porta 1378 de outra depende da troca de pacotes que contêm *flags* de transmissão com valores lógicos SYN e FIN.
- 76 Sabendo-se que as classes A, B e C de endereços IP são iniciadas com as cadeias de bits “0”, “10”, “110”, respectivamente, bem como que a quantidade teórica máxima de *hosts* individualmente identificados em cada uma dessas classes de redes equivale a  $2^4$ ,  $2^{16}$  e  $2^8$  bits, respectivamente, então é correto afirmar que o servidor *web* envolvido na comunicação apresentada na figura possui endereço classe C e *hostid* (identificador individual de rede) igual a 101.
- 77 A MTU determina a quantidade máxima de *bits* que trafegam em um datagrama de uma rede local e, na rede monitorada, ela possui tamanho superior a 1000 *bytes*.

Abaixo, apresenta-se um diagrama UML obtido a partir da engenharia reversa de um código do projeto Foxes and Rabbits. O código implementa um simulador gráfico de ecossistema, no qual coelhos (da classe `Rabbit`) e raposas (da classe `Fox`) são animais (da classe `Animal`) que competem pela sobrevivência ocupando posições da classe (`Location`) em um espaço físico (da classe `Field`).

**Diagrama UML para os itens de 78 a 87**



Barros e Kolling, Objects first with Java: a practical introduction using BlueJ, 2004.

Julgue os itens seguintes, acerca das informações apresentadas e de conceitos de desenvolvimento de sistemas e de linguagens de programação.

- 78 As classes `Fox` e `Rabbit` herdam de `Animal`, portanto qualquer instância de `Fox` ou `Rabbit` possui as propriedades `age` e `location`.
- 79 As classes `Field`, `Rabbit` e `Fox` possuem, cada uma, um gerador de números pseudo-aleatórios, que é referenciado por uma variável de instância declarada em cada classe.
- 80 A classe `Location` encapsula as variáveis de instância `row` e `col`, e permite que as mesmas sejam alteradas de forma controlada por qualquer cliente dessa classe.
- 81 Todas as classes apresentadas no diagrama herdam diretamente da classe `Object`.

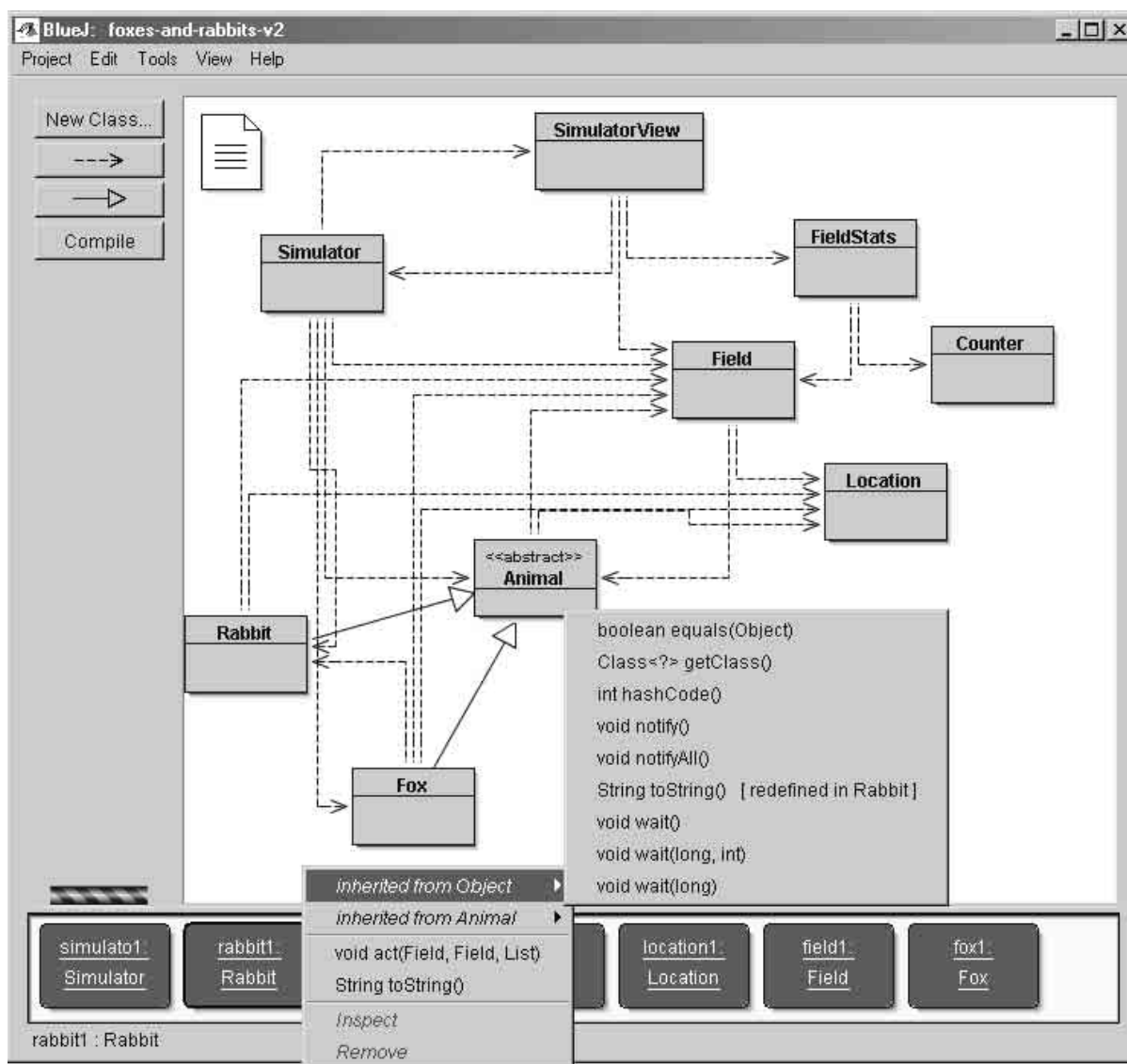


Figura I — Internet: <www.bluej.org>.

A figura I, acima, apresenta a interface de um programa de apoio ao ensino de programação em Java denominado BlueJ. São apresentadas as classes que compõem o projeto Foxes and Rabbits, o mesmo utilizado para realização da engenharia reversa para UML, referida anteriormente. No canto inferior da interface do BlueJ, há instâncias de algumas das classes apresentadas, sendo cada instância referenciada por uma variável. Para que uma instância de cada classe seja criada, é necessária a invocação de *menu pop-up* (botão direito do *mouse*) quando o *mouse* se encontra sobre a classe específica da qual se deseja criar uma instância. Os métodos construtores bem como os métodos estáticos públicos da classe são apresentados no *menu pop-up* e podem ser invocados diretamente pelo usuário, conforme ilustra a figura II, a seguir. Há também a opção de informar parâmetros para a invocação de cada método construtor — como exemplifica a figura III — ou estático. Verifica-se, por exemplo, na parte inferior da interface da figura I, que instâncias das classes Simulator, Rabbit, Location, Field e Fox foram criadas pelo usuário e que simulato1, rabbit1, location1, field1 e fox1 são variáveis que apontam para as instâncias de Simulator, Rabbit, Location, Field e Fox, respectivamente. Percebe-se, ainda nessa figura, a existência de *menus pop-up* que estão sobrepondo objetos. Esses *menus pop-up* permitem ao usuário interagir diretamente com os objetos ou instâncias criadas, realizando invocação dos métodos disponíveis nas interfaces das instâncias e inspecionando os resultados retornados pela invocação de cada método, os quais são apresentados em janelas independentes. Os *menus pop-up* da figura I permitem a invocação dos métodos que existem na interface do objeto apontado pela variável rabbit1, que é uma instância da classe Rabbit. Também é possível realizar inspeção do estado de cada objeto instanciado, conforme verificado nas figuras IV e V, a seguir, por meio do acionamento da opção *Inspect*, disponível no *menu pop-up*.





Figura II

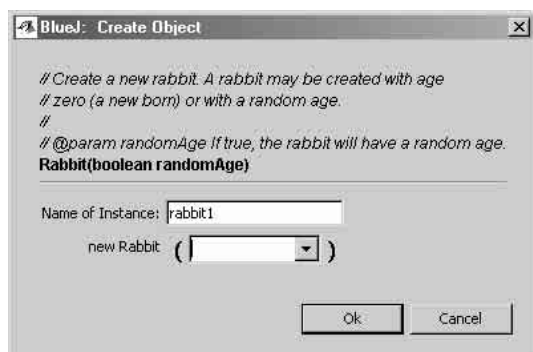


Figura III

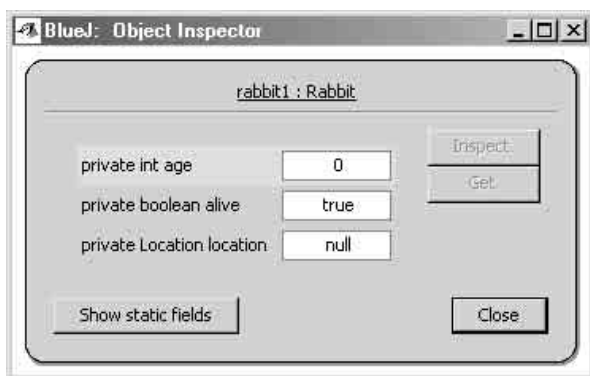


Figura IV

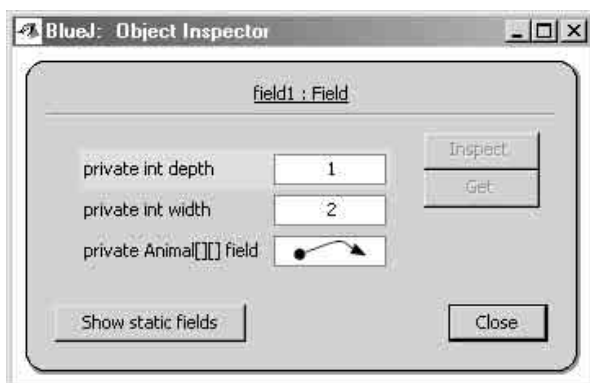
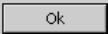


Figura V

A partir das informações apresentadas nas figuras de I a V, julgue os itens de 82 a 87, relativos a conceitos de desenvolvimento de sistemas e linguagens de programação.

82 Caso o usuário invoque o *menu pop-up* sobre a instância de nome `simulato1`, este *menu pop-up* apresentará ao usuário a opção de invocação de métodos com nome `reset`, `runLongSimulation`, `simulate`, `simulateOneStep` e `populate`.

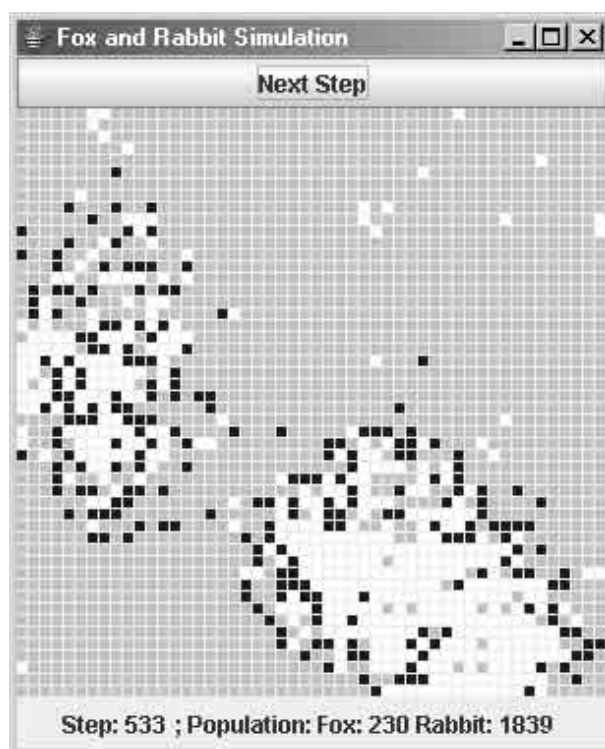
83 Caso o usuário invoque o *menu pop-up* sobre a instância de nome `fox1`, será permitido que ele invoque métodos de nome `reset`, `getAge`, `getLocation`, `isAlive`, `setAge`, `equals`, `hashCode`, `toString` e `wait`.

84 Para que a invocação do construtor apresentada na Figura III seja completada com sucesso, ao se pressionar o botão  nessa janela, o usuário deverá escrever no campo específico valores do tipo `String`.

85 A fim de se preservar a semântica adequada da linguagem Java, a tentativa de instanciação de um objeto da classe `Animal` não deverá ser aceita pelo programa, devido ao fato de a mesma ser declarada abstrata.

86 A inspeção do estado de um objeto do tipo `Location`, como, por exemplo, `location1` na figura I, apresentará ao usuário uma janela contendo valores para variáveis de nome `name` e `count`.

87 A inspeção do estado de um objeto do tipo `Fox`, como, por exemplo, `fox1` na figura I, apresentará ao usuário uma janela contendo valores para variáveis de nome `row`, `foodLevel`, `age`, `alive`, `location`.



A figura acima apresenta uma janela produzida pelo programa `Foxes and Rabbits`, na qual cada retângulo representa um local desabitado ou ocupado por um coelho ou uma raposa. Quando o botão `Next Step` é acionado, uma nova simulação do ecossistema é realizada. Essa janela é implementada pela classe `SimulatorView`, que utiliza a interface `java.awt.event.ActionListener` para tratar eventos produzidos pelo usuário quando do acionamento do botão `Next Step`. O código parcial da classe `SimulatorView` é apresentado a seguir, tendo sido suprimidas algumas linhas.

```

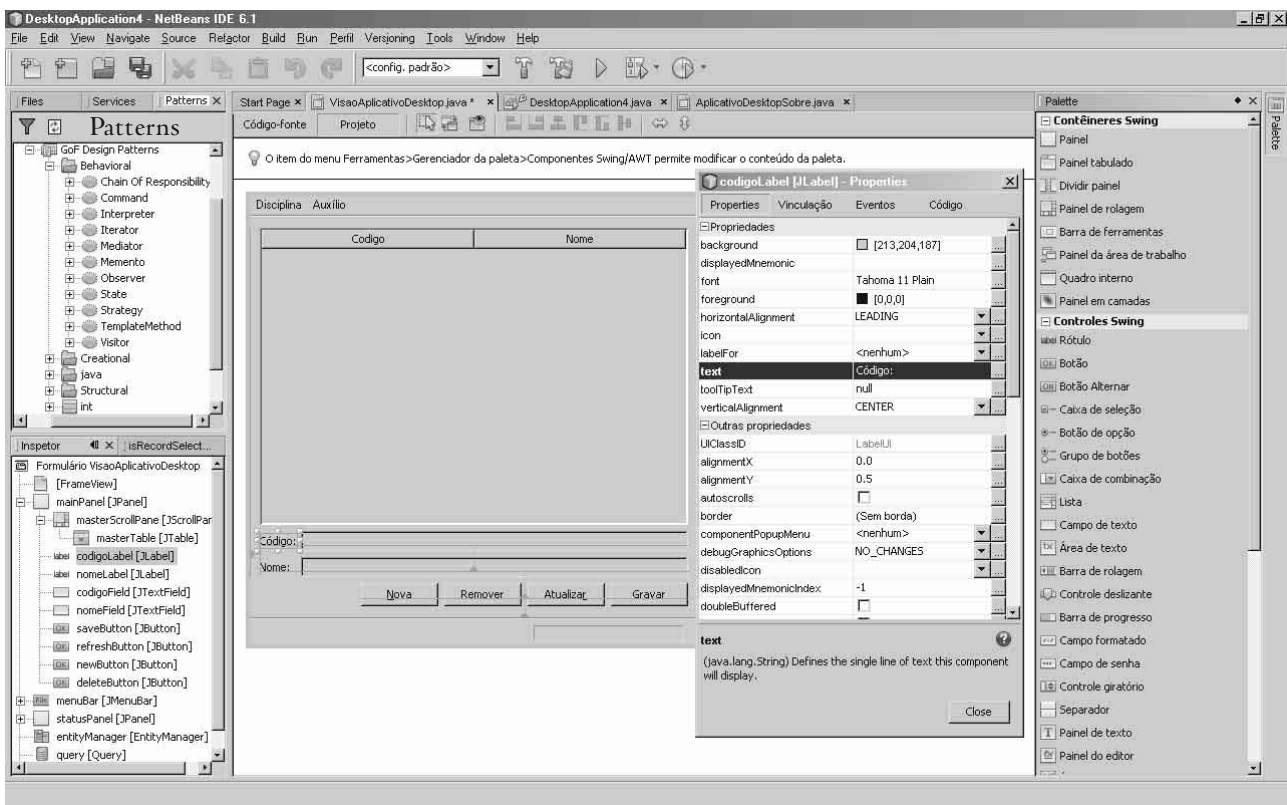
...
public class SimulatorView extends JFrame implements ActionListener {
    private Simulator simulator;
    public void actionPerformed(ActionEvent evt) {
        simulator.simulateOneStep();
    }
...
}

```

Julgue os próximos itens, acerca das informações apresentadas sobre o projeto Foxes and Rabbits e dos conceitos de desenvolvimento de sistemas.

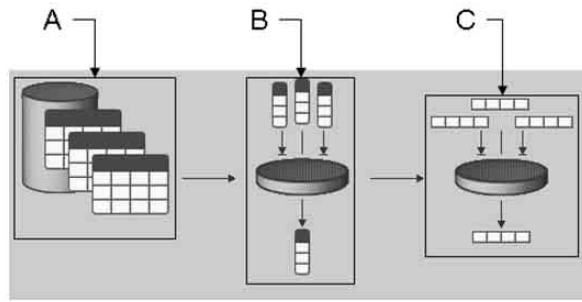
**88** Conforme o padrão MVC (*model-view-controller*), as classes que desempenham os papéis de *model*, *view* e *controller*, na implementação da interface gráfica do projeto Foxes and Rabbits, são Simulator, SimulatorView e SimulatorView, respectivamente.

**89** ActionListener é uma interface Java que declara a assinatura de método público: void actionPerformed(java.awt.event.ActionEvent evt).



Na figura acima, que apresenta um aspecto da interface produzida pela IDE (ambiente interativo de desenvolvimento), ou ferramenta RAD (desenvolvimento rápido de aplicações), chamada NetBeans 6.1, são apresentadas várias janelas ou abas de informação, entre as quais se destacam as de nome: Patterns, Inspector, VisaoAplicativoDesktop, codigoLabel [JLabel] - Properties e Palette. Considerando essa figura, julgue os itens de 90 a 94, relativos às informações apresentadas e a conceitos de desenvolvimento de sistemas e linguagens de programação.

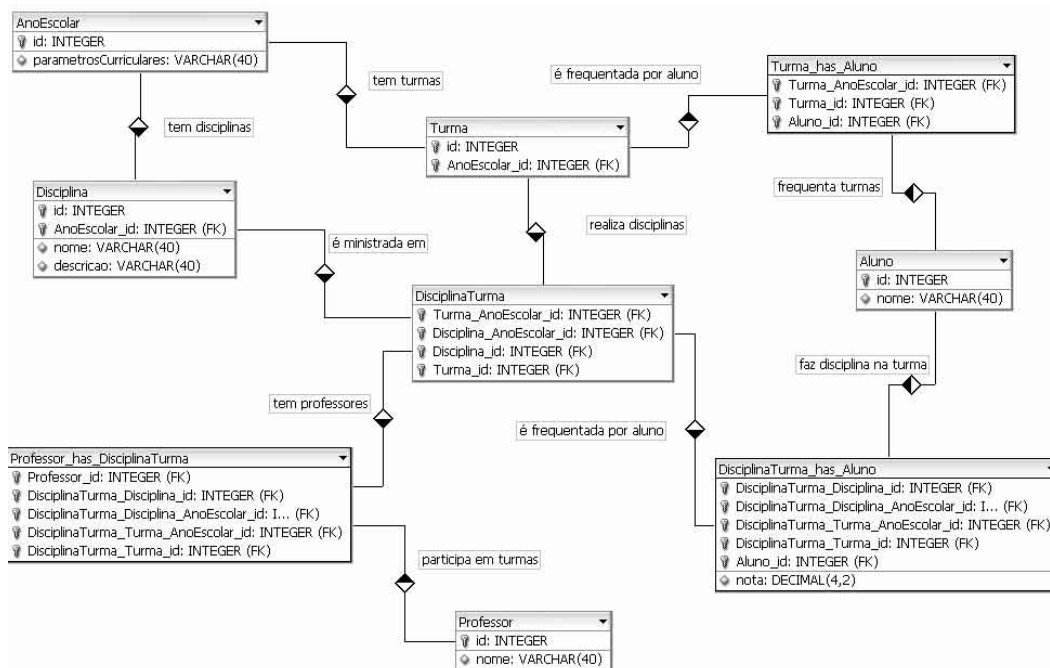
- 90** NetBeans é um ambiente de programação visual, conforme as informações de quatro das janelas ou abas destacadas.
- 91** O *framework* arquitetural MVC incorpora, usualmente, um dos padrões de desenho GoF do tipo comportamental apresentados na interface da figura acima.
- 92** As classes `java.awt.Container` e `java.awt.Component`, declaradas na biblioteca padrão da linguagem Java, implementam, juntas, um padrão de projeto GoF do tipo comportamental, denominado Composite.
- 93** A interface gráfica sob edição declara variáveis de tipos contêineres e controles. São variáveis de tipos contêiner: `mainPanel` e `statusPanel`. São variáveis de tipo controles: `newButton`, `refreshButton`, `deleteButton`, `codigoField` e `nomeField`.
- 94** Uma inspeção do código da classe `VisaoAplicativoDesktop` revelará que existe qualquer atribuição de valor à propriedade `text` do componente apontado pela variável `codigoLabel` declarada dentro da classe `VisaoAplicativoDesktop`.



A figura acima apresenta três abstrações conceituais de uma consulta SQL, empregadas no gerador de consultas do SGBD DB2, e identificadas por A, B e C. Considerando que uma consulta possua as cláusulas SELECT, FROM, ORDER BY, WHERE, HAVING e GROUP BY, julgue os itens seguintes, acerca dessas abstrações de consultas em bancos de dados.

- 95 A abstração A está diretamente relacionada com os argumentos da cláusula FROM.
- 96 A abstração B está diretamente relacionada com os argumentos da cláusula GROUP BY.
- 97 A abstração C está diretamente relacionada com os argumentos da cláusula WHERE.

**Texto para os itens de 98 a 107**



O diagrama da figura acima apresenta um modelo lógico de um banco de dados de registro escolar. As principais entidades do modelo são: AnoEscolar (ex: 1.º, 2.º, 3.º anos etc.), Disciplina (ex: Matemática do 1.º ano, Matemática do 2.º ano etc.), Turma (ex: Turma 1 do 1.º ano escolar, Turma 2 do 1.º ano escolar etc.), Aluno (ex: Lucas tem identificação número 1, Mariana tem identificação número 2 etc.) e Professor (ex: João é identificado pelo número 1, Maria é identificada pelo número 2 etc.). As relações de chave estrangeira estabelecidas entre as tabelas são marcadas com (FK) e definidas por meio de prefixos baseados nos nomes das tabelas referenciadas.

Considerando as informações do texto, julgue os itens de 98 a 102, acerca dos conceitos de bancos de dados.

- 98 São estabelecidas relações de cardinalidade 1:n, nos níveis conceitual e lógico, entre AnoLetivo e Turma e entre AnoLetivo e Disciplina.
- 99 São estabelecidas relações de cardinalidade m:n, no nível conceitual, entre Aluno e DisciplinaTurma e entre Professor e DisciplinaTurma.
- 100 O modelo não permite que uma Turma esteja sem vínculo a um AnoEscolar; e que uma turma seja criada sem nenhum aluno matriculado.
- 101 O modelo não permite que um registro de professor seja inserido sem que haja uma turma associada a ele; e que um registro de aluno seja inserido sem que haja uma correspondente disciplina a cursar.
- 102 Acerca das dependências funcionais do modelo, é correto afirmar que o nome de um professor é funcionalmente dependente do seu id; e que o identificador de uma turma não é funcionalmente nem transitivamente dependente do identificador de um ano letivo.

Considere, para julgamento dos próximos itens, que o banco de dados cujo modelo lógico é apresentado na figura encontra-se implementado com registros inseridos adequadamente.

**103** O comando SQL abaixo retornará os nomes de todos os alunos que freqüentam a turma 1.

```
SELECT A.NOME
FROM ESCOLA.TURMA_HAS_ALUNO as T, ESCOLA.ALUNO as A
WHERE A.id = T.Aluno_id and T.TURMA_id = 1;
```

**104** O comando SQL abaixo retornará a lista das notas em disciplinas de todos os alunos, de todas as disciplinas, de todas as turmas da escola.

```
SELECT A.NOME, D.NOME, DTA.nota
FROM ESCOLA.ALUNO as A, ESCOLA.Disciplina as D, ESCOLA.DisciplinaTurma as DT,
ESCOLA.DisciplinaTurma_has_Aluno as DTA;
```

Julgue os itens seguintes, com relação ao uso de um SGDB do tipo IBM DB2, considerando que o usuário esteja devidamente conectado ao banco de dados que contém um esquema denominado ESCOLA, para o qual o usuário possui plenas permissões para executar comandos administrativos.

**105** O comando SQL a seguir declara satisfatoriamente a tabela AnoEscolar, considerado o conjunto das restrições de integridade de entidade apresentadas no modelo lógico.

```
CREATE TABLE ESCOLA.AnoEscolar (
  id INTEGER NOT NULL,
  parametrosCurriculares INTEGER NOT NULL,
  PRIMARY KEY(id));
```

**106** O comando SQL abaixo declara satisfatoriamente a tabela DisciplinaTurma\_has\_Aluno, considerado o conjunto das restrições de integridade referencial apresentadas no modelo do diagrama.

```
CREATE TABLE ESCOLA.DisciplinaTurma_has_Aluno (
  DisciplinaTurma_Disciplina_id INTEGER NOT NULL ,
  DisciplinaTurma_Disciplina_AnoEscolar_id INTEGER NOT NULL ,
  DisciplinaTurma_Turma_AnoEscolar_id INTEGER NOT NULL ,
  DisciplinaTurma_Turma_id INTEGER NOT NULL ,
  Aluno_id INTEGER NOT NULL ,
  nota DECIMAL(4,2) NULL ,
  PRIMARY KEY(DisciplinaTurma_Disciplina_id, DisciplinaTurma_Disciplina_AnoEscolar_id,
  DisciplinaTurma_Turma_AnoEscolar_id, DisciplinaTurma_Turma_id, Aluno_id),
  FOREIGN KEY(DisciplinaTurma_Turma_AnoEscolar_id, DisciplinaTurma_Disciplina_AnoEscolar_id,
  DisciplinaTurma_Disciplina_id, DisciplinaTurma_Turma_id)
  REFERENCES ESCOLA.DisciplinaTurma (Turma_AnoEscolar_id, Disciplina_AnoEscolar_id, Disciplina_id,
  Turma_id),
  FOREIGN KEY(Aluno_id)
  REFERENCES ESCOLA.Aluno(id));
```

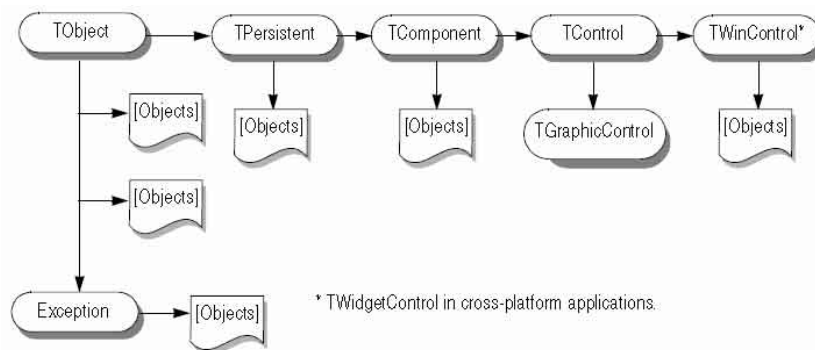
**107** Se o banco de dados escola encontra-se sem nenhum registro em nenhuma de suas tabelas, então, o comando SQL a seguir será bem sucedido em inserir um registro no banco de dados.

```
INSERT INTO ESCOLA.Professor_has_DisciplinaTurma
( Professor_idProfessor, DisciplinaTurma_Disciplina_idDisciplina,
DisciplinaTurma_Disciplina_AnoEscolar_idAnoEscolar, DisciplinaTurma_Turma_idTurma,
DisciplinaTurma_Turma_AnoEscolar_idAnoEscolar)
VALUES(2,1 ,1 ,1 ,1 );
```

<pre> 1 import java.io.*; 2 public class DictionaryTest { 3     public static void main(String args[]) throws IOException { 4         Dictionary dict = new Dictionary(); // Create a empty dictionary. 5         dict.put("1", "Biologia"); 6         dict.put("2", "Desenho Técnico"); 7         dict.put("2", "Educação Física"); 8         dict.put("4", "Eletrônica"); 9         dict.put("9", "Informática"); 10         System.out.println(dict.remove("1")); 11         System.out.println(dict.remove("2")); 12         if ((dict.get("1") == null)&amp;(dict.get("2") == null)) { 13             String disciplina2 = (String) dict.get("9"); 14             System.out.println(disciplina2); 15         } 16         System.out.println(dict.isEmpty()); 17     } 18 } </pre>	<pre> 1 public class Dictionary { 2     protected DictLinkEntry head; 3     public Dictionary() { 4         head = new DictLinkEntry(null, null); 5         head.setNext(head); 6         head.setPrev(head); 7     } 8     protected DictLinkEntry find(Object key) { 9         DictLinkEntry node = head.getNext(); 10         while ((node != head) &amp;&amp; (key.equals(node.getKey()))) { 11             node = node.getNext(); 12         } 13         if (node == head) { 14             return null; 15         } else { 16             return node; 17         } 18     } 19     protected void linkIntoList(DictLinkEntry node) { 20         DictLinkEntry headNext = head.getNext(); 21         node.setNext(headNext); 22         node.setPrev(head); 23         head.setNext(node); 24         headNext.setPrev(node); 25     } 26     protected void unlinkFromList(DictLinkEntry node) { 27         DictLinkEntry prev = node.getPrev(); 28         DictLinkEntry next = node.getNext(); 29         prev.setNext(next); 30         next.setPrev(prev); 31     } 32     public Object put(Object key, Object element) { 33         DictLinkEntry where = find(key); 34         if (where == null) { 35             DictLinkEntry node = new DictLinkEntry(key, element); 36             linkIntoList(node); 37             return null; 38         } else { 39             Object previousElement = where.getElement(); 40             where.setElement(element); 41             return previousElement; 42         } 43     } 44     public Object remove(Object key) { 45         DictLinkEntry where = find(key); 46         if (where == null) { 47             return null; 48         } else { 49             unlinkFromList(where); 50             return where.getElement(); 51         } 52     } 53     public Object get(Object key) { 54         DictLinkEntry where = find(key); 55         if (where == null) { 56             return null; 57         } else { 58             unlinkFromList(where); 59             linkIntoList(where); 60             return where.getElement(); 61         } 62     } 63     public boolean isEmpty() { 64         return head.getNext() == head; 65     } 66 } </pre>
<pre> 1 public class DictLinkEntry { 2     protected Object key; // The entry's key. 3     protected Object element; // The entry's data. 4     protected DictLinkEntry prev; // The link to the previous entry. 5     protected DictLinkEntry next; // The link to the next entry. 6     public DictLinkEntry(Object key, Object element) { 7         this.key = key; 8         this.element = element; 9         prev = null; 10         next = null; 11     } 12     public Object getKey() { 13         return key; 14     } 15     public Object getElement() { 16         return element; 17     } 18     public void setElement(Object element) { 19         this.element = element; 20     } 21     public DictLinkEntry getNext() { 22         return next; 23     } 24     public void setNext(DictLinkEntry newNext) { 25         next = newNext; 26     } 27     public DictLinkEntry getPrev() { 28         return prev; 29     } 30     public void setPrev(DictLinkEntry newPrev) { 31         prev = newPrev; 32     } 33 } </pre>	

Acima, apresenta-se a implementação funcionalmente completa de três classes declaradas na linguagem Java, que definem uma estrutura de dados com funcionalidade de um dicionário. O dicionário é implementado com as classes `Dictionary` e `DictLinkEntry`. A `DictionaryTest` contém um programa (método `main`) que realiza um teste de funcionamento do dicionário, por meio da instanciação de um dicionário e da realização de operações de inserção (método `put()`), remoção (método `remove()`) e consulta (métodos `get()` e `isEmpty()`). Julgue os itens abaixo, acerca das informações apresentadas e dos conceitos de programação em Java.

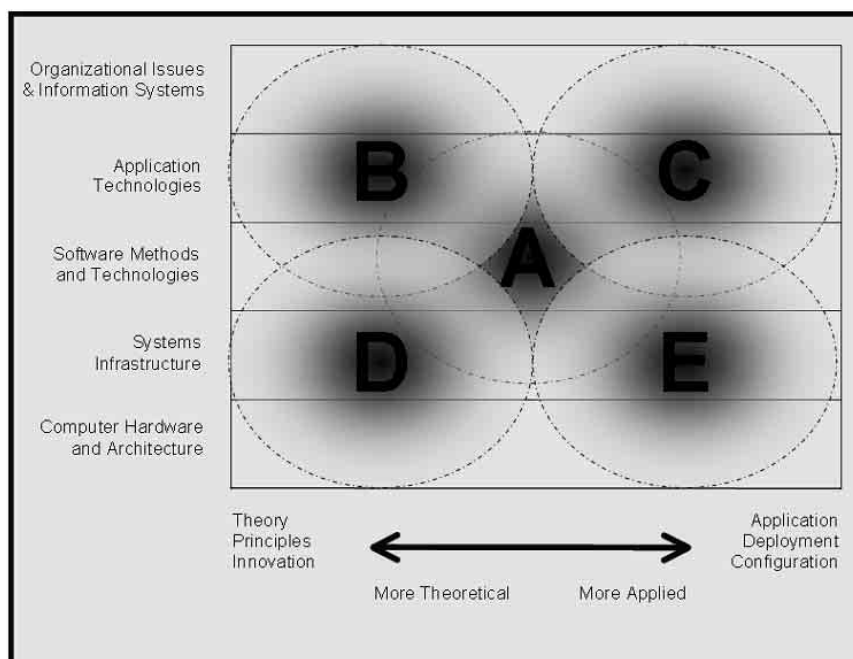
- 108 Os métodos explicitamente declarados no código das três classes são quantitativamente assim organizados: um estático; dois construtores; e 14 de instância.
- 109 Nenhuma das variáveis de instância declarada em `Dictionary` e `DictLinkEntry` são encapsuladas, mesmo que seja no nível de *package*.
- 110 A estrutura de dados implementada pela classe é uma lista ligada e circular, com encadeamento simples.
- 111 Durante a execução do programa, apenas uma instância da classe `Dictionary` será criada.
- 112 As linhas 13 e 14 da classe `DictionaryTest` não serão executadas.
- 113 Imediatamente antes da execução da linha de código 16 da classe `DictionaryTest`, a linha de código 7 da classe `DictLinkEntry` terá sido executada exatamente 5 vezes.
- 114 A saída de dados gerada pelo programa apresentará as strings `Biologia`, `Educação Física`, `Informática` e `false`, nesta seqüência.



Developer's Guide for Borland® Delphi™ for Windows, 2002, p. 45.

A figura acima apresenta um diagrama de relacionamento entre algumas das principais abstrações que compõem o ambiente Delphi. Considerando essa figura, julgue os itens a seguir, acerca dos conceitos de programação.

- 115 Os retângulos de cantos arredondados representam classes, ao passo que as setas representam adequadamente as relações de herança, em conformidade com a notação empregada na linguagem UML.
- 116 Todos os objetos gráficos que aparecem nas interfaces do Delphi são da classe TControl ou de suas subclasses, como TGraphicControl, TWinControl, TWidgetControl etc.
- 117 Programas desenvolvidos na plataforma Delphi, da mesma forma que programas desenvolvidos na plataforma Java, são compilados para um formato de `bytecode` interpretável, para posterior uso em ambientes independentes de plataforma, como Windows e Linux.



Computing Curricula 2004 Overview Report: A Guide to Undergraduate Degree Programs in Computing, Association for Computing 2004 (com adaptações).

A figura acima apresenta o espaço de problemas para ensino de computação e informática, no qual devem se situar os currículos de formação de graduação, como Ciência da Computação, Engenharia de Computação, Sistemas de Informação, Tecnologia da Informação e Engenharia de Software. Sobre esse espaço foram delimitadas regiões hipotéticas de desenvolvimento curricular, identificadas pelas letras de A a E. Considerando a necessidade de desenvolvimento de currículos para ensino de computação nos níveis fundamental e médio, julgue os próximos itens.

- 118 A educação de crianças, no ensino básico, deve concentrar-se na abordagem dos aspectos das regiões B e D, em detrimento da exploração das regiões na união entre C e E.
- 119 A formação de técnicos de nível médio para atuar no desenvolvimento de *software* junto a empresas deve concentrar-se na abordagem dos aspectos da região A e de sua intersecção com as regiões da direita do gráfico.
- 120 A formação de técnicos de nível médio para atuar no gerenciamento de sistemas de rede deve concentrar-se na abordagem dos aspectos da região E e de sua intersecção com as demais regiões do gráfico.