

HOSPITAL DE CLÍNICAS DE PORTO ALEGRE

EDITAL N.º 03/2015 DE PROCESSOS SELETIVOS

GABARITO APÓS RECURSOS

PROCESSO SELETIVO 22

ANALISTA DE TI I (Arquitetura de Sistemas)

01.	B	11.	E	21.	D	31.	A
02.	D	12.	C	22.	B	32.	A
03.	D	13.	E	23.	ANULADA	33.	A
04.	A	14.	E	24.	E	34.	D
05.	A	15.	A	25.	B	35.	E
06.	C	16.	C	26.	D	36.	D
07.	E	17.	A	27.	E	37.	E
08.	C	18.	B	28.	C	38.	D
09.	B	19.	B	29.	A	39.	E
10.	D	20.	C	30.	D	40.	B

EDITAL Nº 03/2015
DE PROCESSOS SELETIVOS (PS)

MISSÃO

Ser um referencial público em saúde, prestando assistência de excelência, gerando conhecimento, formando e agregando pessoas de alta qualificação.

PS 22 - ANALISTA DE TI I
(Arquitetura de Sistemas)

MATÉRIA	QUESTÕES	PONTUAÇÃO
Conhecimentos Específicos	01 a 40	0,25 cada

**FAURGS**
Fundação de Apoio da Universidade Federal do Rio Grande do Sul

DIREITOS AUTORAIS RESERVADOS. PROIBIDA A REPRODUÇÃO, AINDA QUE PARCIAL, SEM A PRÉVIA AUTORIZAÇÃO DA FAURGS E DO HCPA.

Nome do Candidato: _____

Inscrição nº: _____



- 1 Verifique se este CADERNO DE QUESTÕES corresponde ao Processo Seletivo para o qual você está inscrito. Caso não corresponda, solicite ao Fiscal da sala que o substitua.
- 2 Esta PROVA consta de **40** (quarenta) questões objetivas.
- 3 Caso o CADERNO DE QUESTÕES esteja incompleto ou apresente qualquer defeito, solicite ao Fiscal da sala que o substitua.
- 4 Para cada questão objetiva, existe apenas **uma** (1) alternativa correta, a qual deverá ser assinalada na FOLHA DE RESPOSTAS.
- 5 Os candidatos que comparecerem para realizar a prova **não deverão portar** armas, malas, livros, máquinas calculadoras, fones de ouvido, gravadores, *paggers*, *notebooks*, **telefones celulares**, *pen drives* ou quaisquer aparelhos eletrônicos similares, nem utilizar véus, bonés, chapéus, gorros, mantas, lenços, aparelhos auriculares, prótese auditiva, óculos escuros, ou qualquer outro adereço que lhes cubra a cabeça, o pescoço, os olhos, os ouvidos ou parte do rosto. **Os relógios de pulso serão permitidos, desde que permaneçam sobre a mesa, à vista dos fiscais, até a conclusão da prova.** (conforme subitem 7.10 do Edital de Abertura)
- 6 **É de inteira responsabilidade do candidato comparecer ao local de prova munido de caneta esferográfica preferencialmente de tinta azul, de escrita grossa, para a adequada realização de sua Prova Escrita. Não será permitido o uso de lápis, marca textos, lapiseira/grafite e/ou borracha durante a realização da prova.** (conforme subitem 7.15.2 do Edital de Abertura)
- 7 Não serão permitidos: nenhuma espécie de consulta em livros, códigos, revistas, folhetos ou anotações, nem o uso de instrumentos de cálculo ou outros instrumentos eletrônicos, exceto nos casos estabelecidos no item 13 do Edital. (conforme subitem 7.15.3 do Edital de Abertura)
- 8 Preencha com cuidado a FOLHA DE RESPOSTAS, evitando rasuras. Eventuais marcas feitas nessa FOLHA a partir do número **41** serão desconsideradas.
- 9 Ao terminar a prova, entregue a FOLHA DE RESPOSTAS ao Fiscal da sala.
- 10 A duração da prova é de **três horas e trinta minutos (3h30min)**, já incluído o tempo destinado ao preenchimento da FOLHA DE RESPOSTAS. Ao final desse prazo, a FOLHA DE RESPOSTAS será **imediatamente** recolhida.
- 11 **O candidato somente poderá se retirar da sala de prova uma hora (1h) após o seu início. Se quiser levar o Caderno de Questões da Prova Escrita Objetiva, o candidato somente poderá se retirar da sala de prova uma hora e meia (1h30min) após o início. O candidato não poderá anotar/copiar o gabarito de suas respostas de prova.**
- 12 **Após concluir a prova e se retirar da sala de prova, o candidato somente poderá se utilizar de sanitários nas dependências do local de prova, se for autorizado pela Coordenação do Prédio e estiver acompanhado de um fiscal.** (conforme subitem 7.15.6 do Edital de Abertura)
- 13 Ao concluir a Prova Escrita, o candidato deverá devolver ao fiscal da sala a Folha de Respostas (Folha Óptica) e/ou o Caderno de Respostas, quando se tratar de prova dissertativa. Se assim não proceder, será excluído do Concurso. (Conforme subitem 7.15.8 do Edital de Abertura)
- 14 A desobediência a qualquer uma das recomendações constantes nas presentes instruções poderá implicar a anulação da prova do candidato.

01. Qual é a principal característica de uma linguagem orientada a objetos que a diferencia de outros tipos de linguagens?

- (A) Possuir definição de tipos de dados recursivos.
- (B) Possuir suporte a abstração, subtipos, herança e carga dinâmica de métodos.
- (C) O código gerado pelo compilador é sempre executado no sistema nativo.
- (D) Todos os dados manipulados, incluindo inteiros e booleanos, são objetos.
- (E) Suportar de forma direta a herança múltipla de atributos e métodos.

02. Em linguagens orientadas a objetos, polimorfismo significa

- (A) a possibilidade de uma classe herdar métodos de várias classes diferentes.
- (B) a possibilidade de uma classe ter como parâmetro uma ou mais classes diferentes.
- (C) a definição de uma classe que fornece operações genéricas sobre suas instâncias.
- (D) a possibilidade de um objeto A ser usado em qualquer contexto que espera um objeto B, se ambos têm a mesma funcionalidade.
- (E) a possibilidade de uma classe implementar uma classe virtual.

03. Em Java, os direitos de acesso aos membros de uma classe podem ser classificados em três níveis: *private*, *public* e *protected*. Sobre esse tema, analise as afirmações abaixo.

- I - Membros *public* podem ser acessados por qualquer classe, e membros *private* são restritos à própria classe.
- II - Membros *protected* são restritos à própria classe e às classes derivadas.
- III - Membros *private* devem ser implementados por uma classe que os deriva.

Quais estão corretas?

- (A) Apenas I.
- (B) Apenas II.
- (C) Apenas III.
- (D) Apenas I e II.
- (E) Apenas II e III.

04. Acerca da linguagem Java, é correto afirmar que

- (A) uma subclasse em Java estende as funcionalidades de uma outra classe pela inclusão de novos atributos e/ou novos métodos.
- (B) o método *main* é o chamado método construtor.
- (C) os métodos de uma subclasse precisam ser chamados referenciando, também, o nome da superclasse.
- (D) para criar novas classes usa-se o comando "new".
- (E) não é possível estender uma classe acrescentando métodos públicos, devido ao esquema de encapsulamento de código.

05. Em Java, quando um objeto de uma classe A é criado, o que é executado primeiro?

- (A) Todo o corpo do método construtor da classe A.
- (B) O método construtor de objetos das classes que são estendidas pela classe A.
- (C) O método construtor de uma classe que estende a classe A.
- (D) A inicialização dos atributos da classe A.
- (E) O código de inicialização de todas as variáveis de instância que são públicas.

06. No que se refere a métodos e exceções em Java, assinale com **V** (verdadeiro) ou **F** (falso) as afirmações abaixo.

- () Um método de classe, ao contrário de um método de instância, é declarado em Java através da palavra-chave *static*.
- () Exceções permitem fluxos inesperados (alternativos) de execução, que não necessariamente correspondem a erros de programação que poderiam ter sido evitados com uma programação mais robusta.
- () A sobre-escrita de um método é a capacidade de declararmos vários métodos com o mesmo nome em uma mesma classe, desde que tenham o mesmo número de argumentos, os quais variam em tipo.
- () Métodos de classe podem acessar variáveis de classe e variáveis de instância.

A sequência correta de preenchimento dos parênteses, de cima para baixo, é

- (A) V – F – F – V.
- (B) F – V – V – F.
- (C) V – V – F – F.
- (D) F – F – V – V.
- (E) V – V – V – F.

Instrução: Considere o código Java, abaixo, para responder às questões de nº 07 a 09.

```
public class Produto
{
    public String nomeProduto;
    private int codProduto;
    private int preco;
    public Produto (String nome){ this(nome, 0, 0);}
    public Produto(String nome, int codigo, int preco)
    {
        nomeProduto = nome;
        codProduto = codigo;
        this.preco = preco;
    }
    public void imprimeDadosProduto ()
    {
        System.out.println("Nome.....:"+nomeProduto);
        System.out.println("Codigo....:"+codProduto);
        System.out.println("Preco.....:"+preco);
    }
}
```

07. Uma linha de código

```
Produto tapete = new Produto ("tapete");
```

corresponde à

- (A) declaração de uma variável com identificador "Produto", com um valor "tapete" para o campo *nomeProduto* e demais campos indefinidos.
- (B) declaração de uma variável com identificador "Produto", com um valor "tapete" para o campo *nomeProduto* e demais campos zerados.
- (C) declaração e instanciação de um objeto com identificador "Produto", com um valor "tapete" para o campo *nomeProduto* e demais campos indefinidos.
- (D) declaração e criação de um objeto com identificador "tapete", com um valor "tapete" para o campo *nomeProduto* e demais campos indefinidos.
- (E) declaração e criação de um objeto com identificador "tapete", com um valor "tapete" para o campo *nomeProduto* e demais campos zerados.

08. Na linha de código

```
this.preco = preco;
```

o uso de "this" significa uma referência

- (A) ao campo *preco* da superclasse.
- (B) ao campo *preco* da subclasse.
- (C) ao campo *preco* da instância sendo criada.
- (D) à própria variável parâmetro.
- (E) ao valor *default* que deve ser usado quando o parâmetro estiver zerado.

09. Considere o trecho de programa abaixo e as afirmações a seguir, relacionadas à linguagem Java.

```
public class Perecivel extends Produto
{
    private int validade;
    public Perecivel (String nome, int codigo, int preco, int validade)
    {
        super (nome, codigo, preco);
        this.validade = validade;
    }
}
```

- I - A classe Perecivel é uma superclasse da classe Produto.
- II - Um objeto da classe Perecivel herda todos os membros públicos e protegidos da classe Produto.
- III- Métodos da classe Perecivel podem acessar diretamente todos os atributos da classe Produto.

Quais estão corretas?

- (A) Apenas I.
- (B) Apenas II.
- (C) Apenas III.
- (D) Apenas I e III.
- (E) Apenas II e III.

10. Assinale a alternativa que apresenta o padrão de projeto que separa a construção de um objeto complexo de sua representação, de modo que o mesmo processo de construção possa criar diferentes representações.

- (A) *Factory.*
- (B) *Command.*
- (C) *Adapter.*
- (D) *Builder.*
- (E) *Interpreter.*

11. Assinale a alternativa que contém um padrão de projeto estrutural.

- (A) *Iterator.*
- (B) *Mediator.*
- (C) *Interpreter.*
- (D) *Command.*
- (E) *Adapter.*

12. Qual é o padrão de projeto utilizado para que uma interface (abstração) possa variar independente de suas implementações?

- (A) *Builder.*
- (B) *Composite.*
- (C) *Bridge.*
- (D) *Iterator.*
- (E) *Adapter.*

13. Qual é o padrão de projeto que permite a um objeto alterar o seu comportamento em função de modificações no seu estado interno?

- (A) *Strategy.*
- (B) *Composite.*
- (C) *Template.*
- (D) *Visitor.*
- (E) *State.*

14. Considere as seguintes afirmações sobre servidores de aplicação.

- I - Servidores de aplicação são *softwares* que fornecem uma infraestrutura de serviços para a execução de aplicações distribuídas.
- II - Os servidores de aplicação são executados em servidores e acessados pelos clientes através de uma conexão de rede.
- III- Os servidores de aplicação também são conhecidos como *softwares* de *middleware*.

Quais estão corretas?

- (A) Apenas I.
- (B) Apenas II.
- (C) Apenas III.
- (D) Apenas II e III.
- (E) I, II e III.

15. Assinale a alternativa que contém uma implementação de servidor de aplicação implementado na plataforma Java.

- (A) *Oracle Application Server*
- (B) PHP
- (C) SQL
- (D) PL/SQL
- (E) *Hibernate*

16. Considere o aumento de demanda – novas aplicações, maior número de usuários conectados –, quando mais máquinas podem ser incorporadas multiplicando assim o poder de resposta.

Assinale a alternativa que apresenta a característica adicional dos servidores de aplicação que oferece escalabilidade de serviços e recursos.

- (A) Gerenciamento de componentes.
- (B) Tolerância a falhas.
- (C) Balanceamento de carga.
- (D) Gerenciamento de transações.
- (E) Gerenciamento de console.

17. Gerenciamento de sessão, notificação e distribuição da lógica de negócios são componentes e serviços manipulados através de ferramentas.

Assinale a alternativa que apresenta a característica adicional dos servidores de aplicação que serve para gerenciar elementos do servidor.

- (A) Gerenciamento de componentes.
- (B) Tolerância a falhas.
- (C) Balanceamento de carga.
- (D) Gerenciamento de transações.
- (E) Gerenciamento de console.

18. Assinale a alternativa que apresenta o estado de um objeto no *Hibernate*, em que a instância é armazenada de forma persistente e se garante que uma instrução INSERT não é executada fora dos limites de uma transação.

- (A) Transitório.
- (B) Persistente.
- (C) Excluído.
- (D) Visível.
- (E) Abstrato.

19. Assinale a alternativa que apresenta o método do *Hibernate* que fornece uma maneira de recuperar uma instância persistente, pegando um objeto de classe e carregando o seu estado em uma instância mais recente de modo persistente.

- (A) persistente()
- (B) load()
- (C) transiente()
- (D) detached()
- (E) insert()

20. No *Hibernate*, uma consulta é executada ao invocar o método _____. O resultado da consulta é carregado completamente em uma coleção _____. Instâncias de entidades recuperadas por uma consulta ficam em um estado _____.

Assinale a alternativa que preenche correta e respectivamente as lacunas do texto acima.

- (A) array() – no disco – transitório
- (B) query() – na memória – persistente
- (C) list() – na memória – persistente
- (D) consult() – no disco – transitório
- (E) insert() – na memória – persistente

21. Qual é o método do *Hibernate* que oferece um atalho de forma que uma consulta retorne apenas um único objeto?

- (A) value()
- (B) query()
- (C) distinct()
- (D) uniqueResult ()
- (E) persistente()

22. Qual é o método que permite escrever consultas em SQL nativo de forma que o *Hibernate* faça o mapeamento do resultado para os objetos?

- (A) nativeSQL()
- (B) createSQLQuery()
- (C) coreSQL()
- (D) defaultSQL()
- (E) insertSQL()

23. Analise o trecho de código abaixo, escrito em *Hibernate*.

```
package eg;
public class DomesticCat extends Cat {
    private String name;
    public String getName() {
        return name;
    }
    protected void setName(String name) {
        this.name=name;
    }
}
```

Com base nesse código, é correto afirmar que

- (A) *private* define a classe *Cat* como visível.
- (B) *DomesticCat* herda da subclasse *Cat*.
- (C) *extends* define a classe *Cat* como persistente.
- (D) *Cat* herda da subclasse *DomesticCat*.
- (E) *protected* define a classe *DomesticCat* como protegida.

24. Analise o trecho de código abaixo, escrito em *Hibernate*.

```
@Entity
public class Flight implements Serializable {
    @ManyToOne( cascade = {CascadeType.PERSIST,
        CascadeType.MERGE} )
    @JoinColumn(name="COMP_ID")
    public Company getCompany() {
        return company;
    }
    ...
}
```

Com base nesse código, é correto afirmar que

- (A) *public* define a classe *Company* como protegida.
- (B) *Cascade* exclui permanentemente todos os objetos da classe *Flight*.
- (C) *getCompany()* exclui um objeto da classe *Flight*.
- (D) *@JoinColumn* define uma junção externa entre duas colunas.
- (E) *@ManyToOne* define um relacionamento muitos para um entre as entidades.

25. Analise o trecho de código abaixo, escrito em *Hibernate*.

```
@Entity
public class Person {
    @Id Integer getId() { ... }
    ...
}
```

Qual é a anotação que permite marcar uma propriedade como identificador?

- (A) @Entity
- (B) @Id
- (C) getId()
- (D) Integer
- (E) public

26. Qual é o componente do *Primefaces* que permite tratar exceções?

- (A) *ExceptionHandler*.
- (B) *ExceptionManner*.
- (C) *PrimeException*.
- (D) *AjaxExceptionHandler*.
- (E) *AjaxHandler*.

27. Assinale com **V** (verdadeiro) ou **F** (falso) as seguintes afirmações, relativas aos nomes de eventos no *Primefaces*.

- () *default* – inicia visível quando a página é carregada.
- () *start* – quando uma resposta ajax é recebida sem erro.
- () *error* – quando a resposta ajax é recebida com um erro HTTP.
- () *sucess* – quando toda execução é finalizada.

A sequência correta de preenchimento dos parênteses, de cima para baixo, é

- (A) V – F – F – V.
- (B) F – V – V – F.
- (C) F – F – V – V.
- (D) V – V – F – F.
- (E) V – F – V – F.

28. Qual é o componente do *Primefaces* que fornece sugestões quando uma entrada de dados está sendo digitada?

- (A) *InsertTips*.
- (B) *CompleteAll*.
- (C) *AutoComplete*.
- (D) *ToolTips*.
- (E) *CompleteTips*.

29. Qual é o componente do *Primefaces* que é usado para reduzir o tempo de carregamento da página, depois da renderização inicial?

- (A) *Cache*.
- (B) *Load*.
- (C) *Slow*.
- (D) *Stop*.
- (E) *Clock*.

30. Assinale com **V** (verdadeiro) ou **F** (falso) as seguintes afirmações, relativas ao controle de transações em PL/SQL.

- () COMMIT torna permanente no banco de dados todas as alterações realizadas durante a execução da transação.
- () SET TRANSACTION define as propriedades da transação, como acesso de leitura/gravação e nível de isolamento.
- () SAVEPOINT finaliza a transação atual e desfaz todas as mudanças feitas desde o início da transação.
- () ROLLBACK marca o ponto atual no processamento de uma transação.

A sequência correta de preenchimento dos parênteses, de cima para baixo, é

- (A) V – F – F – V.
- (B) F – V – V – F.
- (C) F – F – V – V.
- (D) V – V – F – F.
- (E) V – F – V – F.

31. Qual é o operador de comparação em PL/SQL que retorna *true* (verdadeiro) se a subconsulta retorna, pelo menos, uma linha da tabela?

- (A) EXIST
- (B) BETWEEN
- (C) ALL
- (D) ANY
- (E) NOT NULL

32. Um problema complexo pode ser simplificado em um conjunto de problemas simples que têm soluções fáceis de implementar.

Assinale a alternativa que apresenta a característica de PL/SQL que permite quebrar uma aplicação em módulos gerenciáveis, através de sucessivos refinamentos.

- (A) Modularidade.
- (B) Abstração.
- (C) Transações.
- (D) Estruturas de dados.
- (E) Tratamento de exceção.

33. Assinale a alternativa correta sobre os pacotes mais importantes do CAMEL.

- (A) O *model* contém as classes que constituíram o domain-specific language (DSL). A classe principal é a *RouteBuilder*.
- (B) O *impl* implementa os padrões de integração.
- (C) O *processor* implementa o modelo de dados do CAMEL.
- (D) O *designer* é um gerenciador de projetos.
- (E) O *builder* é a interface do provedor de serviços.

34. Qual é o serviço *web* que usa XML para fazer as requisições?

- (A) REST.
- (B) *Hibernate*.
- (C) *Apache*.
- (D) SOAP.
- (E) SQL.

35. Qual é o serviço *web* que pode usar quatro operações (GET, POST, PUT, and DELETE) para executar tarefas?

- (A) SOAP.
- (B) *Hibernate*.
- (C) .NET.
- (D) *Apache*.
- (E) REST.

36. Qual serviço *web* **NÃO** é necessário utilizar com *HyperText Transfer Protocol* (HTTP), porque há uma especificação para utilizá-lo com o *Simple Mail Transfer Protocol* (SMTP)?

- (A) REST.
- (B) *Hibernate*.
- (C) *Apache*.
- (D) SOAP.
- (E) SQL.

37. Dentro do contexto de controle de versões e configuração, considere as seguintes afirmações sobre *Subversion* (SVN).

- I - No SVN, a cópia de trabalho é a sua própria área de trabalho privada: o SVN nunca incorpora as alterações de outras pessoas, nem torna suas próprias alterações disponíveis para os outros, a menos que seja explicitamente configurado assim. Pode até haver múltiplas cópias de trabalho do mesmo projeto.
- II - Programas clientes do *Subversion* usam URLs para identificar arquivos e diretórios versionados em repositórios *Subversion*.
- III- Cada vez que o repositório aceita um *commit*, isto cria um novo estado – chamado revisão – da árvore do sistema de arquivos.

Quais estão corretas?

- (A) Apenas I.
- (B) Apenas I e II.
- (C) Apenas I e III.
- (D) Apenas II e III.
- (E) I, II e III.

38. No contexto de controle de versões e configuração, em particular do sistema *Subversion* (SVN), _____ é uma linha de desenvolvimento que existe independente de outra linha, embora ainda compartilhe uma história comum, se for olhado para trás na linha do tempo.

Assinale a alternativa que preenche corretamente a lacuna do parágrafo acima.

- (A) *merge*
- (B) *data store*
- (C) repositório (*repository*)
- (D) *branch*
- (E) alteração (*change*)

39. Considere as seguintes afirmações sobre Teste de *Performance* (teste de desempenho).

- I - Os testes de desempenho são muitas vezes acoplados ao teste de esforço e usualmente requerem instrumentação de *hardware* e *software*.
- II - O teste de desempenho é projetado para testar o desempenho em tempo de execução do *software* dentro do contexto de um sistema integrado. O teste de desempenho é feito em todas as etapas no processo de teste. Até em nível de unidade, o desempenho de um módulo individual pode ser avaliado durante o teste. No entanto, o verdadeiro desempenho de um sistema só pode ser avaliado depois que todos os elementos do sistema estiverem totalmente integrados.
- III- Para sistemas em tempo real e embarcados (*embedded systems*), um *software* que execute a função necessária, mas não esteja em conformidade com os requisitos de desempenho, é inaceitável.

Quais estão corretas?

- (A) Apenas I.
- (B) Apenas I e II.
- (C) Apenas I e III.
- (D) Apenas II e III.
- (E) I, II e III.

40. Considere as seguintes afirmações sobre Qualidade de *Software*.

- I - No contexto da gestão da qualidade, "defeito" implica um problema de qualidade que é descoberto depois de o *software* ter sido liberado para os usuários finais (ou para uma outra atividade estrutural dentro da gestão da qualidade). Já "erro" indica um problema de qualidade que é descoberto por engenheiros de *software* (ou outros) antes de o *software* ser liberado aos usuários finais (ou para uma outra atividade estrutural dentro da gestão da qualidade).
- II - "Eficiência" é um fator de qualidade que expressa a quantidade de recursos computacionais e códigos exigidos por um programa para desempenhar sua função.
- III- "Portabilidade" é um fator de qualidade que expressa o esforço necessário para modificar um programa em operação.

Quais estão corretas?

- (A) Apenas I.
- (B) Apenas I e II.
- (C) Apenas I e III.
- (D) Apenas II e III.
- (E) I, II e III.