



EDITAL Nº 01/2020
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 06 - ANALISTA DE TI I
(Desenvolvimento)

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



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

Nome do Candidato: _____

Inscrição nº: _____



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

INSTRUÇÕES

- 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, próteses auditivas, óculos escuros, ou qualquer outro adereço que lhes cubra a cabeça, o pescoço, os olhos, os ouvidos ou parte do rosto, sob pena de serem excluídos do certame. **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, régua, lapiseiras/grafites e/ou borrachas durante a realização da prova.** (conforme subitem 7.15.2 do Edital de Abertura)
- 7 Não será permitida 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 em que forem pré-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 (3h)**, 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, 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, o candidato somente poderá se utilizar de sanitários nas dependências do local de prova se for autorizado pela Coordenação do Prédio e se 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). Se assim não proceder, será excluído do Processo Seletivo. (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 destas operações de escrita **NÃO** é suportada pelo mecanismo de bulkWrite do banco de dados MongoDB?

- (A) insertOne.
- (B) updateOne.
- (C) updateMany.
- (D) replaceMany.
- (E) deleteMany.

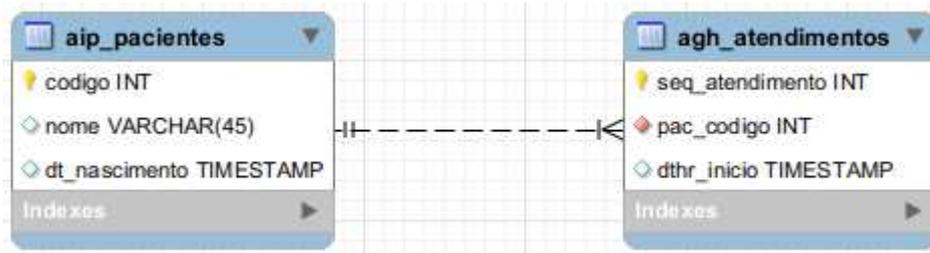
02. Considere o código abaixo, executado em um SGDB Oracle 11G:

```
declare
  on_hand INTEGER := null;
begin
  if on_hand = 0 then
    on_hand := 1;
  else
    on_hand := 2;
  end if;
  DBMS_OUTPUT.PUT_LINE(on_hand);
end;
```

Qual a saída esperada na saída DBMS do SGDB?

- (A) 0
- (B) 1
- (C) 2
- (D) O código não compila.
- (E) Ocorre erro em tempo de execução.

03. Considere as tabelas aip_pacientes e agh_atendimentos abaixo:



No SGBD Oracle, a consulta que retorna o nome do aip_pacientes e a quantidade de agh_atendimentos para pacientes que tiveram mais de 10 agh_atendimentos no hospital é:

- (A) SELECT nome, COUNT(*) AS QUANTIDADE FROM aip_pacientes, agh_atendimentos WHERE codigo = pac_codigo AND COUNT(*) > 10
- (B) SELECT nome FROM aip_pacientes, agh_atendimentos WHERE codigo = pac_codigo AND COUNT(*) > 10 GROUP BY codigo
- (C) SELECT nome, count(*) AS QUANTIDADE FROM aip_pacientes, agh_atendimentos WHERE codigo = pac_codigo GROUP BY nome HAVING COUNT(*) >10
- (D) SELECT nome, COUNT(*) AS QUANTIDADE FROM aip_pacientes, agh_atendimentos WHERE codigo = pac_codigo AND QUANTIDADE > 10 GROUP BY codigo
- (E) SELECT nome, COUNT(*) AS QUANTIDADE FROM aip_pacientes, agh_atendimentos WHERE codigo = pac_codigo GROUP BY codigo HAVING QUANTIDADE >10

04. Assinale a alternativa que apresenta o comando que **NÃO** executa em uma base PostgreSQL.

- (A) create table public.prova (id int not null)
- (B) create or replace view public.v_prova as (select id from public.prova)
- (C) with w_table as (select id from public.prova) select * from w_table
- (D) select now() from dual
- (E) select coalesce(null,1,2)

05. Considere o parágrafo abaixo.

Com o PL / SQL, você pode usar instruções SQL para _____ dados do Oracle e instruções de fluxo de controle para _____ os dados. Você também pode declarar _____, definir _____ e interceptar erros de tempo de execução.

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

- (A) manipular – processar – constantes e variáveis – procedimentos e funções
- (B) manipular – processar – procedimentos e funções – constantes e variáveis
- (C) processar – manipular – procedimentos e funções – constantes e variáveis
- (D) processar – manipular – constantes e variáveis – procedimentos e funções
- (E) ler – manipular – procedimentos e funções – constantes e variáveis

06. Sobre as ferramentas Helm e Kubernetes, assinale a alternativa correta.

- (A) Kubernetes é uma plataforma de containerização de aplicações e sistemas.
- (B) Helm é uma ferramenta que auxilia o gerenciamento das aplicações em execução na *engine* de orquestração Kubernetes.
- (C) Na *engine* Kubernetes somente é permitido o uso da plataforma de *containers* Docker.
- (D) Um *Chart* do Helm é um único arquivo que define os valores *default* de uma instalação na *engine* Kubernetes.
- (E) O comando “helm install” apenas aceita *Charts* de arquivos locais.

07. Considere as afirmações abaixo a respeito da ferramenta de orquestração de *containers* Kubernetes e assinale-as com **V** (verdadeiro) ou **F** (falso).

- () *Pod* é a unidade básica de execução de uma aplicação no Kubernetes.
- () É possível controlar um *cluster* Kubernetes pela ferramenta *kube-control*.
- () *ReplicaPod* é o componente da arquitetura Kubernetes que possui a função de manter executando um conjunto de réplicas de Pods simultaneamente.
- () Na arquitetura de um *cluster* Kubernetes, *nodes* são as unidades de processamento controladas pelo *master*.
- () Kubernetes é uma ferramenta tradicional de PaaS (*Platform as a Service*).

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

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

08. Considere as afirmações abaixo sobre princípios fundamentais de DevOps segundo *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations. IT Revolution* (KIM et al., 2016).

- I - A primeira maneira trata do fluxo de trabalho da esquerda para a direita.
- II - A segunda maneira trata do *feedback* recíproco rápido e constante.
- III- A terceira maneira foca na criação de uma cultura de aprendizagem contínua e na experimentação.

Quais estão corretas?

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

09. Considere as afirmações abaixo sobre Scrum.

- I - O Scrum é um *framework* de gerenciamento de projetos leve, simples de entender e fácil de dominar.
- II - O Time Scrum é composto pelo Product Owner, Time de Desenvolvimento e Scrum Master.
- III- O Scrum Master ajuda todos a entenderem a teoria, a prática, as regras e os valores do Scrum.
- IV- Os artefatos do Scrum são uma lista sobre tudo que se sabe ser necessário para o produto.

Quais estão corretas?

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

10. Considere as afirmativas abaixo sobre programação orientada a objetos.

- I - Sobrecarga permite que se tenham diversos métodos de mesmo nome com parâmetros de entrada diferentes em uma mesma classe.
- II - Métodos construtores são utilizados para instanciar uma determinada classe. Toda classe deve ter um, mas este não necessariamente precisa ser declarado pelo desenvolvedor.
- III- Polimorfismo permite que os mesmos atributos e objetos possam ser utilizados por diferentes objetos de formas distintas e com implementações lógicas diferentes.

Quais estão corretas?

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

11. Qual a palavra-chave utilizada para implementar o conceito de herança na linguagem de programação Java?

- (A) extend.
- (B) implement.
- (C) inherit.
- (D) extends.
- (E) implements.

12. Considere a seguinte classe Java sendo executada em um interpretador Java devidamente instalado e configurado:

```
class Somador {
    private Integer resultado;

    Somador() {
        this.resultado = 0;
    }

    public Integer somar(Integer a, Integer b) {
        resultado = a + b + 1;
        return resultado;
    }

    public int somar (int a, int b) {
        return a + b;
    }

    public int somar (int a) {
        return a + 1;
    }

    public int somar() {
        return resultado + 1;
    }

    public static void main (String args []) {
        Somador s = new Somador();

        System.out.print(" Teste 1: " + s.somar(new
        Integer(1), new Integer(2)));
        System.out.print(" Teste 2: " + s.somar(1,
        2));
        System.out.print(" Teste 3: " + s.somar(new
        Integer(2)));
        System.out.print(" Teste 4: " + s.somar());
    }
}
```

Qual a saída esperada no console do sistema ao executar o código acima?

- (A) Teste 1: 3 Teste 2: 3 Teste 3: 3 Teste 4: 1
- (B) Teste 1: 3 Teste 2: 4 Teste 3: 4 Teste 4: 1
- (C) Teste 1: 4 Teste 2: 3 Teste 3: 3 Teste 4: 5
- (D) O código não compila.
- (E) Ocorre erro em tempo de execução.

13. Sobre os diferentes tipos de testes que podem ser aplicados em um *software* para identificar suas falhas, assinale com **V** (verdadeiro) ou **F** (falso) as afirmações abaixo.

- () Técnicas de teste de caixa-branca avaliam tanto os aspectos internos quanto os externos, de entrada e saída.
- () Técnicas de teste de caixa-preta se concentram nas entradas e saídas do objeto de teste sem referência à sua estrutura interna.
- () Os testes de componente validam unidades de um *software* que podem ser testados separadamente.
- () Os testes de *stress* avaliam os limites de uso do *software*, o quanto ele suporta em volume de informações, tráfego, etc., sem que apresente erros.

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

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

14. Considere os parágrafos abaixo sobre as três dimensões de qualidade.

_____ : o sistema é resistente a falhas durante a execução, isto é, não entra em *loop*, não interrompe a execução por falta de recursos, etc.

_____ : o sistema se comporta conforme o esperado e definido em seus requisitos.

_____ : o sistema tem um tempo de resposta adequado e aceitável, mesmo quando submetido a um volume de processamento próximo de situações reais ou de pico.

Assinale a alternativa que completa, correta e respectivamente, as lacunas dos parágrafos acima.

- (A) Confiança – Funcionalidade – Performance
- (B) Confiança – Funcionalidade – Robustez
- (C) Robustez – Funcionalidade – Performance
- (D) Confiança – Comportamental – Robustez
- (E) Robustez – Comportamental – Performance

15. Numere a segunda coluna de acordo com a primeira, relacionando os algoritmos de ordenação com as suas respectivas características.

- (1) MERGESORT
- (2) QUICKSORT
- (3) SELECTION SORT
- (4) SHELLSORT

- () É o método de dividir e conquistar para classificação. Ele funciona particionando uma matriz em duas submatrizes e, em seguida, classificando-as independentemente. A posição da partição depende do conteúdo da matriz.
- () Divide-se em duas metades, classifica as duas metades de forma recursiva e mescla os resultados.
- () É uma extensão simples da classificação por inserção que ganha velocidade ao permitir trocas de entradas de matriz distantes para produzir matrizes parcialmente classificadas, que podem ser classificadas com eficiência, eventualmente por classificação por inserção.
- () Encontra o menor item da matriz e troca-o pela primeira entrada, em seguida, encontra o próximo item menor e troca-o pela segunda entrada. Continua dessa maneira até que toda a matriz seja classificada.

A sequência numérica correta de preenchimento dos parênteses da segunda coluna, de cima para baixo, é

- (A) 1 – 2 – 3 – 4.
- (B) 1 – 2 – 4 – 3.
- (C) 2 – 1 – 3 – 4.
- (D) 2 – 1 – 4 – 3.
- (E) 4 – 3 – 2 – 1.

16. Sobre Collections Java, considere as afirmações abaixo.

- I - *Set collections* não podem conter elementos duplicados.
- II - *Map* são formados por chave valor, e sua chave não pode ser duplicada.
- III- *SortedMap* mantém seus mapeamentos em ordem crescente de valor.
- IV - *List* pode conter valores duplicados.

Quais estão corretas?

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

17. A injeção de dependência do projeto XYZ foi implementada utilizando CDI. Este projeto possui 2 *beans* com o mesmo nome, porém com implementações diferentes, que devem ser escolhidas dinamicamente durante o *deploy* da aplicação. Qual técnica deve ser utilizada para o correto *deploy* e a correta execução de cada implementação?

- (A) Utilizar a anotação `@Alternatives` em uma das implementações e declarar a classe com a tag `<alternatives>` dentro do arquivo `beans.xml`.
- (B) Utilizar a anotação `@Implementation` em uma das implementações e declarar a classe com a tag `<implementation>` dentro do arquivo `beans.xml`.
- (C) Utilizar a anotação `@Alternative` em uma das implementações e declarar a classe com a tag `<alternative>` dentro do arquivo `beans.xml`.
- (D) Utilizar a anotação `@Implementations` em uma das implementações e declarar a classe com a tag `<implementations>` dentro do arquivo `beans.xml`.
- (E) Isto não é permitido pela especificação do CDI.

18. Dentro de uma solução estruturada numa arquitetura de microsserviços e implementada utilizando Spring Cloud, qual o principal papel do Spring Cloud Netflix Zuul?

- (A) Fornecer a configuração dos demais componentes da arquitetura.
- (B) Permitir que o *front-end* chame os serviços do *back-end*.
- (C) Depurar a comunicação entre microsserviços.
- (D) Permitir comunicação assíncrona entre os microsserviços.
- (E) Permitir que o *back-end* execute consultas na camada de persistência.

19. Considere o parágrafo abaixo.

A arquitetura predominante na plataforma Java EE é composta de três camadas, porque os componentes desta arquitetura são distribuídos em: _____, _____ e máquinas de _____.

Assinale a alternativa que completa, correta e respectivamente, as lacunas do parágrafo acima.

- (A) funções – servidor Apache – balanceamento de carga
- (B) clientes – servidor de Aplicação – banco de dados
- (C) métodos – servidor Nginx – *firewall*
- (D) objetos – servidor Apache – banco de dados
- (E) clientes – servidor Nginx – monitoramento

20. Qual padrão de projetos desacopla uma abstração de sua implementação para que abstração e implementação possam variar de forma independente?

- (A) Bridge.
- (B) Strategy.
- (C) Abstract Factory.
- (D) Flyweight.
- (E) Facade.

21. O protocolo SOAP possui uma linguagem de definição do serviço cuja sigla é

- (A) XLSX.
- (B) WASP.
- (C) REST.
- (D) WADL.
- (E) WSDL.

22. Considere o parágrafo abaixo.

No Java EE, o _____ fornece a funcionalidade para WebServices _____. Ele é adequado para cenários _____ de integração e geralmente é mais bem integrado ao HTTP do que os serviços baseados em _____, não requer mensagens XML ou definições de API de serviço.

Assinale a alternativa que completa, correta e respectivamente, as lacunas do parágrafo acima.

- (A) JAX-WS – REST – básicos – SOAP
- (B) JAX-WS – SOAP – complexos – REST
- (C) JAX-WS – REST – complexos – SOAP
- (D) JAX-RS – REST – básicos – SOAP
- (E) JAX-RS – SOAP – complexos – REST

23. Considere o trecho de código na linguagem Java abaixo:

```
public class Questao {
    public static void main(String[] args) {
        List<Integer> entradas = new ArrayList<Integer>();
        for (int i=0; i < 10; i++) {
            entradas.add(i);
        }
        Questao q = new Questao();
        System.out.println(q.processaEntradas(entradas));
    }

    public HashSet<Integer> processaEntradas(List<Integer> entradas){
        HashSet<Integer> a = new HashSet();

        for (Integer in : entradas) {
            if(in%2 == 0){
                continue;
            }else {
                a.add(this.processaValor(in));
            }
        }
        return a;
    }

    public Integer processaValor(Integer valorProcessado){
        boolean encontrou = false;
        for (int i=2; i < valorProcessado-1; i++) {
            if(valorProcessado%i == 0){
                return i;
            }
        }
        if(!encontrou)
            return valorProcessado;
        return -1;
    }
}
```

Assinale a alternativa que contém o resultado correto da execução desse código.

- (A) [1, 2, 3, 5, 7, 3]
- (B) [1, 2, 3, 5, 7]
- (C) [1, 3, 5, 7]
- (D) [1, 3, 5, 7, -1]
- (E) [1, 3, 5, 7, 3]

24. Dado o trecho de código Java:

```
String[] textos = new String[7];
Arrays.fill(textos, "Paciente");
System.out.println(textos[3]);
```

Qual a saída esperada da execução do código acima?

- (A) i
- (B) c
- (C) null
- (D) Paciente
- (E) Ocorre um erro *ArrayIndexOutOfBoundsException* na execução.

25. Considere o objeto java Cliente abaixo:

```
class Cliente {
    @Id
    private Integer id;
    private String Nome;

    public void setId(Integer id) {this.id = id;}
    public Integer getId() { return this.id;}
    public void setNome(String nome) { this.nome = nome;}
    public String getNome() { return this.nome; }
}
```

Considere que o objeto "em" contenha uma instância do EntityManager devidamente conectado com banco, rodando em Servidor de Aplicação EJB com transação gerenciada por um *container*. É executado o trecho de código abaixo:

```
Cliente c = new Cliente();
c.setId(1);
c.setNome("Cliente 1");
c = em.merge(c);
c.setNome("Cliente com nome alterado");
```

Após encerrada a transação do código acima, qual o resultado esperado ao recuperar o valor da propriedade "nome" do objeto Cliente com id=1 em outra sessão do EntityManager?

- (A) null
- (B) Cliente 1
- (C) Cliente com nome alterado
- (D) Ocorre erro em tempo de execução ao encerrar a transação da gravação dos dados, sem ação de finalização de transação definida.
- (E) Ocorre erro em tempo de execução ao tentar recuperar o Cliente id=1, sem ação de finalização de transação definida.

26. Considere o SQL abaixo:

```
SELECT
  nome
FROM
  AIP_PACIENTES
WHERE
  nome LIKE 'EDSON%'
  AND cor IS NULL
  AND codigo BETWEEN 10 AND 200000
```

Considere a seguir a existência de uma instância `DetachedCriteria "criteria"` e a implementação Hibernate do JPA para a tabela `AIP_PACIENTES`. Qual alternativa trará o mesmo retorno da consulta SQL acima?

- (A) `criteria.add(like("nome", "EDSON%"));`
`criteria.add(isNull("cor"));`
`criteria.add(between("codigo", 10, 200000));`
`criteria.setProjection("nome");`
- (B) `criteria.add(like("nome", "EDSON", MatchMode.END));`
`criteria.add(isNull("cor", true));`
`criteria.add(between("codigo", 10, 200000));`
`criteria.setProjection(property("nome"));`
- (C) `criteria.add(Restrictions.like("nome", "EDSON", MatchMode.END));`
`criteria.add(Restrictions.isNull("cor"));`
`criteria.add(Restrictions.between("codigo", 10, 200000));`
`criteria.setProjection(Projections.property("nome"));`
- (D) `criteria.add(Restrictions.ilike("nome", "EDSON", MatchMode.END));`
`criteria.add(Restrictions.isNull("cor", true));`
`criteria.add(Restrictions.between("codigo", 10, 200000));`
`criteria.setProjection(Projections.property("nome"));`
- (E) `criteria.add(Restrictions.like("nome", "EDSON%"));`
`criteria.add(Restrictions.isNull("cor"));`
`criteria.add(Restrictions.between("codigo", 10, 200000));`
`criteria.setProjection(Projections.property("nome"));`

27. Considere o código em Java abaixo:

```
interface Pes {
    Integer getProva();
}
class Pessoa extends PessoaPai {
    @Override
    public Integer getProva() {
        return 4;
    }
}
class PessoaAvo extends PessoaNeto {
    private Integer prova = 4;
}
class PessoaBis extends PessoaNeto {
    private Integer prova = 4;
    @Override
    public Integer getProva() {
        return ++prova + super.getProva();
    }
}

class PessoaFilho extends Pessoa {}
class PessoaNeto extends Pessoa {
    private Integer exercicio = 2;
    @Override
    public Integer getProva() {
        return exercicio-- + 3;
    }
}

class PessoaPai implements Pes {
    @Override
    public Integer getProva() {
        return 6;
    }
}

public class Prova {
    public static void main(String[] args) {
        Pes[] ps = new Pes[5];
        ps[0] = new PessoaBis();
        ps[1] = new Pessoa();
        ps[2] = new PessoaFilho();
        ps[3] = new PessoaNeto();
        ps[4] = new PessoaAvo();
        for (Pes pac : ps) {
            System.out.println(pac.getProva());
        }
    }
}
```

Assinale a alternativa que contém o resultado gerado, de cima para baixo, no console, após a execução desse código.

- (A) 10
4
4
5
5
- (B) 9
4
4
4
4
- (C) 9
4
4
5
5
- (D) 10
4
4
4
4
- (E) 9
5
5
5
5

28. Considere o código em Java abaixo:

```
import java.util.regex.Pattern;
import java.util.regex.Matcher;
public class MMS {
    private static final String REGEX = "\\bhcp\\b";
    private static final String INPUT = "hcp hcp hcp
hcpa hcpaa";
    public static void main(String[] args) {
        Pattern p = Pattern.compile(REGEX);
        Matcher m = p.matcher(INPUT);
        int count = 0;
        while (m.find()) {
            count++;
            System.out.println("Match number " +
count + " - " + m.start() + " - " + m.end());
        }
    }
}
```

Assinale a alternativa que contém o resultado gerado, de cima para baixo, no console, após a execução do código acima.

- (A) Match number 1 - 0 - 3
Match number 2 - 4 - 7
Match number 3 - 8 - 11
Match number 4 - 12 - 15
Match number 5 - 17 - 20
- (B) Match number 1 - 0 - 3
Match number 2 - 4 - 7
Match number 3 - 8 - 11
- (C) Match number 1 - 0 - 3
Match number 2 - 4 - 7
Match number 3 - 8 - 11
Match number 4 - 12 - 15
- (D) Match number 1 - 1 - 4
Match number 2 - 5 - 8
Match number 3 - 9 - 12
Match number 4 - 13 - 16
Match number 5 - 18 - 21
- (E) Match number 1 - 1 - 4
Match number 2 - 5 - 8
Match number 3 - 9 - 12

29. Considere o ciclo de vida de um aplicativo JavaServer Faces. Depois que a árvore de componentes é restaurada durante uma solicitação de *postback*, cada componente da árvore extrai seu novo valor dos parâmetros de solicitação, usando seu método `decode()` (`processDecodes()`). O valor é, então, armazenado localmente em cada componente. Essas ações estão relacionadas à qual fase do ciclo de vida de um aplicativo JavaServer Faces?

- (A) *Restore View Phase.*
- (B) *Apply Request Values Phase.*
- (C) *Process Validations Phase.*
- (D) *Update Model Values Phase.*
- (E) *Invoke Application Phase.*

30. A classe `org.primefaces.PrimeFaces` fornece métodos utilitários para processamento de requisições AJAX dentro do *back-end*. Qual método deve ser utilizado em uma *controller* Java para disparar um comando *Javascript* ao término de uma requisição AJAX?

- (A) `PrimeFaces.current().executeJs(String script);`
- (B) `PrimeFaces.current().execute(String script);`
- (C) `PrimeFaces.current().execJs(String script);`
- (D) `PrimeFaces.current().callScript(String script);`
- (E) `PrimeFaces.current().executeScript(String script);`

31. Ao receber os requisitos do Analista de Sistemas, foi solicitada a implementação da exibição de uma coleção de dados com carregamento sob demanda usando rolagem. Qual componente do PrimeFaces deve ser utilizado para realizar essa tarefa?

- (A) `<p:dataScroll value="#{carBean.cars}" var="car" mode="inline" chunkSize="10">#{car.manufacturer}</p:dataScroll>`
- (B) `<p:dataScroller value="#{carBean.cars}" var="car" mode="inline" chunkSize="10">#{car.manufacturer}</p:dataScroller>`
- (C) `<p:dataTable value="#{carBean.cars}" var="car" mode="scroller" chunkSize="10">#{car.manufacturer}</p:dataTable>`
- (D) `<p:dataTable value="#{carBean.cars}" var="car" mode="inline" scroller="true" chunkSize="10"> #{car.manufacturer}</p:dataTable>`
- (E) `<p:dataTableScroller value="#{carBean.cars}" var="car" mode="inline" chunkSize="10">#{car.manufacturer}</p:dataTableScroller>`

32. Assinale a alternativa que contém o comando jQuery para esconder todas as células pares de linhas ímpares de uma tabela HTML.

- (A) `$("#tr:odd td:even").hide();`
- (B) `$("#tr:even td:odd").hide();`
- (C) `$("#td:even tr:odd").css("display", "none");`
- (D) `$("#td:even tr:odd").hide();`
- (E) `$("#tr:odd td:even").css("display", "block");`

33. Considere o fragmento abaixo de uma página web criada com HTML5 e CSS3:

```
<html>
<head>
<style>
  [title~=car] {
    border: 5px solid #ff0000;
  }
</style>
</head>
<body>





</body>
</html>
```

Pode-se afirmar que a configuração CSS3 de borda

- (A) não será aplicada.
- (B) será aplicada a todos os elementos cujo atributo *title* seja exatamente a palavra "car".
- (C) será aplicada a todos os elementos cujo atributo *title* inicie pela palavra "car".
- (D) será aplicada a todos os elementos cujo atributo *title* contenha a palavra "car".
- (E) será aplicada a todos os elementos cujo atributo *title* termine pela palavra "car".

34. Na sintaxe do HTML5, existem tipos diferentes de elementos. Assinale a alternativa que lista todos esses tipos de elementos.

- (A) *Tags elements, void elements, the template element, raw text elements, escapable raw text elements, foreign elements e normal elements.*
- (B) *Tags elements, void elements, the template element, raw text elements, foreign elements e normal elements.*
- (C) *Tags elements, void elements, the template element, raw text elements e foreign elements.*
- (D) *Void elements, the template element, raw text elements, escapable raw text elements, foreign elements e normal elements.*
- (E) *Void elements, the template element, raw text elements, foreign elements e normal elements.*

35. Assinale a alternativa que contém a correta definição de React.

- (A) É uma biblioteca JavaScript para construção de interfaces para o usuário.
- (B) É um *framework* para desenvolvimento web.
- (C) É uma linguagem de programação orientada a objetos.
- (D) É uma ferramenta que auxilia na criação e na manutenção de *containers docker*.
- (E) É uma ferramenta para o desenvolvimento orientado a TDD.

36. Segundo as definições de *Test Driven Development* (TDD), assinale com **V** (verdadeiro) ou **F** (falso) as afirmações abaixo.

- () A aplicação de TDD requer que o desenvolvedor crie testes que validem os requisitos em código antes de escrever o código da aplicação.
- () A utilização de TDD encoraja o desenvolvimento de código complexo e inspira confiança.
- () Através da aplicação de TDD, os desenvolvedores podem aplicar o conceito para melhorar e depurar código legado desenvolvido a partir de outras técnicas.
- () Ao utilizar TDD, deve-se escrever primeiramente um teste que falha.

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

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

37. Considere o código-fonte abaixo, escrito em linguagem Python:

```
tup = (1, 2, 3)
list = [2, 3, 4]
dic = {"1": 1, "2": 2, "3": 3}
```

A sintaxe **INCORRETA** para a alteração de tuplas, listas ou dicionários é

- (A) dic["tres"] = list
- (B) list [0] = [i for i in tup]
- (C) tup[0] = list[0]
- (D) list = tup [::]
- (E) dic[1] = tup[1]

38. Considere os comandos abaixo na linguagem Python:

```
x = "awesome"
def myfunc():
    global x
    x = "fantastic"
myfunc()
print("Python is " + x)
```

Qual o resultado da execução do código acima?

- (A) Python is
- (B) Python is awesome
- (C) Python is fantastic
- (D) Python is awesome fantastic
- (E) Código não executa.

39. Considere as afirmativas abaixo sobre *Git*.

- I - *Git clone* clona um repositório em um novo diretório.
- II - *Git mv* move ou renomeia um arquivo, diretório ou uma ligação simbólica.
- III- *Git push* atualiza referências remotas junto aos objetos associados.
- IV- *Git pull* busca e integra outro repositório ao repositório local.
- V - *Git branch* lista, cria ou deleta repositórios.

Quais estão corretas?

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

40. Sobre *Pipeline* Jenkins, assinale as afirmações abaixo com **V** (verdadeiro) ou **F** (falso).

- () O script do *Pipeline* somente pode ser definido dentro da ferramenta Jenkins e nunca é versionado como código-fonte.
- () *Pipelines* podem ser organizados em *stages*; e estes, por sua vez, em *steps*.
- () A diretiva "agent" permite definir um comportamento multiagente no *Pipeline*.
- () *Pipelines* podem ser *Declaratives* ou *Scripted*.

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

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

HOSPITAL DE CLÍNICAS DE PORTO ALEGRE

EDITAL Nº 01/2020 DE PROCESSOS SELETIVOS

GABARITO APÓS RECURSOS

PROCESSO SELETIVO 06

ANALISTA DE TI I (Desenvolvimento)

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