**TRABALHO N2**

|  |
| --- |
| Curso: Análise e Desenvolvimento de Sistemas  NOTA  Componente curricular: Programação de Objetos Distribuidos  Acadêmico:  Data:  Ano / semestre: 2017-1 |

|  |
| --- |
| Orientações durante a avaliação  Celular desligado - Silêncio absoluto - Interpretação das questões faz parte da prova - Caneta azul ou preta - Não comer nem beber - Sem consulta  Leia atentamente cada uma das questões e responda marcando a alternativa correta no gabarito presente ao final da prova. Somente o gabarito será analisado como sendo a resposta final a cada uma das questões e ele deverá conter o seu nome. Leia as instruções do gabarito antes de preenchê-lo. |

1. Indique Verdadeiro (V) ou Falso nas opções abaixo.

1. ( ) RPC – Remote Procedure Call - serve para desenvolver aplicações distribuídas cliente-servidor com processos locais e remotos.
2. ( ) RMI - Remote Method Invocation – serve para desenvolver aplicações distribuídas cliente-servidor com threads executadas assincronamente.
3. ( ) RPC utiliza sockets para a comunicação entre um cliente-servidor.
4. ( ) RMI utiliza sockets entre os módulos de comunicação cliente-servidor com objetos.
5. ( ) Sockects TCP stream utiliza o protocolo UDP com datagramas.

( ) Multicast Socket utiliza um endereço IP único para um grupo de membros participantes no multicast.

1. ( ) Importância de transparência em RMI - Todas as etapas necessárias para empacotamento e troca de mensagens, mais a tarefa de localizar e contactar um objeto remoto, são ocultadas do programador que faz uma chamada remota.
2. ( ) Em um sistema distribuído com transparência de acesso, o cliente não precisa saber da localização do servidor. Em RMI, o objeto que está fazendo a invocação não pode identificar se o objeto é local ou não.
3. ( ) Nos programas distribuídos baseados em eventos, os objetos que geram eventos e os objetos que recebem notificações desses eventos, não precisam conhecer a localização um dos outros.
4. Complete as sentenças abaixo
5. Interfaces .................... e referências de objetos remotos caracterizam parcialmente o modelo de objetos distribuídos.
6. Um objeto *...................................*, usado no contexto de Java RMI serve para criar objetos localmente em objetos-clientes.
7. Uma implementação de *................................*em Java RMI, serve para fornecer respostas em tempo real, para objetos clientes que invocam respostas imediatas de alguma programação implementada em algum *objeto-servant* instanciado pelo objeto-servidor.
8. Uma aplicação que trabalha com Hibernate e EJB possui uma classe POJO − ................................................................. utilizada no mapeamento objeto-relacional com uma tabela do banco de .................................... Nessa classe, há um atributo calculado chamado valorTotalPedido que, para ser utilizado apenas em tempo de execução e descartado após finalizar o seu serviço temporário, deverá ser anotado com ...................................................
9. Responda as questões abaixo
10. O que é um servlet e qual sua finalidade?
11. O que é um framework e qual sua finalidade?
12. Defina EJB
13. Explique a utilização do JSTL
14. Para que serve uma API?
15. O que é uma classe POJO?
16. Quais são os 3 (três) tipos fundamentais de beans do componente EJB?explique cada um
17. Na implementação de projeto corporativo de comércio eletrônico construído na plataforma J2EE, a lógica do negócio poderá ser encapsulada em EJBs (Enterprise JavaBeans)? Porque?
18. O que é thread?
19. Explique JEE
20. O glassfish é?
21. Quais as diferenças do Glassfish para o TomCat?
22. É importante aplicar um padrão MVC em um projeto JEE? Por que?
23. Para que serve o RMI? Como funciona?
24. Para que serve o CORBA? Como funciona?
25. Recentemente foi aprovado o desenvolvimento do seguinte projeto no Tribunal Regional do Trabalho da 15a Região:

O projeto é uma aplicação web simples, que implementa as operações básicas de um cadastro e segue um modelo de implementação orientada a objetos na plataforma Java EE. O MySQL foi o banco de dados escolhido para persistir as informações do cadastro. O projeto utilizará os seguintes frameworks Java:

O I é uma tecnologia para desenvolvimento de soluções web especificada pela plataforma Java EE. O objetivo desse framework é estipular um padrão para soluções web corporativas em Java. O Primefaces é uma extensão deste framework e disponibiliza uma série de componentes visuais predefinidos. Trata-se de uma tecnologia open source cujo objetivo é aumentar a produtividade durante o desenvolvimento, com foco em qualidade e eficiência.

O II é um modelo de componentes, especificado pela plataforma Java EE, elaborado para resolver problemas e desafios complexos de softwares corporativos. São componentes que atuam na camada servidor, classificados como componentes de negócio. Podem ser utilizados em diferentes situações como desenvolvimento distribuído, integração/conectividade com legado, processamento assíncrono baseado Fila / Mensagens, controle transacional e outros. Este componente é responsável pelas regras de negócio, ou seja, a persistência e o controle transacional. A JPAJava Persistence API é o padrão adotado pelo Java para soluções ORM (Objeto Mapeamento Relacional).

A JPA define as regras ORM via anotações, mas por ser uma especificação ou padrão ela depende de um provider. Nessa aplicação poderá ser utilizado o III como o provider JPA.

Os frameworks I, II e III acima que serão utilizados no projeto são, respectivamente:

A Tomcat; JBoss; APEX.

B JSF; EJB; Hibernate.

C Hibernate; JSF; EJB.

D EJB; Hibernate; JSF.

E JBoss; Tomcat; Hibernate

.