

# OBJETOS DISTRIBUÍDOS E INVOCAÇÃO REMOTA

## Trabalho Individual

1. O que é um *Middleware* ? Desenhe uma arquitetura de camadas de software onde uma camada *middleware* é incluída.
  2. Quais características um *middleware* apresenta quanto à (a) transparência de localização, (b) protocolos de comunicação, (c) hardware do computador, (d) sistemas operacionais, (e) uso de linguagens de programação.
  3. O que caracteriza o modelo de objetos distribuídos ?
  4. Questões de projeto para RMI são bastante importantes na implementação de certas aplicações. Procure entender e descrever os três casos citados no Cap 5.
  5. Qual a importância de transparência em RMI ?
  6. Implementação de RMI. Descreva resumidamente o que significa cada uma dos componentes da Figura 5.7.
  7. Qual a diferença principal entre RPC (*Remote Procedure Call*) e RMI (*Remote Method Invocation*) ? Figuras 5.7 e 5.8.
  8. Eventos e Notificações podem ser usados em uma ampla variedade de diferentes aplicações distribuídas. Descreva resumidamente, o que é um sistema baseado em eventos distribuído. Quais são suas principais características. Que componentes do sistema são influenciados por essas características.
  9. Dê um exemplo de um tipo de evento e de um evento e seus atributos.
  10. Quais são os participantes em um sistema distribuído de notificação de eventos ? Descreva resumidamente o que significam.
  11. Uma interface de *Eleição* provê duas operações:
    - vote*: com dois parâmetros através dos quais o eleitor fornece o nome de um candidato (uma string) e o número do eleitor (um inteiro usado para garantir que cada cliente vota somente 1 vez). Os números dos eleitores são alocados, espaçadamente, dentro de um domínio de inteiros para torná-los difícil de adivinhar.
    - result*: com dois parâmetros através dos quais o servidor fornece ao apurador o nome de um candidato e o número de votos obtidos para aquele candidato.
- (a) Codifique esta interface em Java para usar em Java RMI.
- (b) Discutir a semântica de invocação que pode ser utilizada quando o protocolo *request-reply* é implementado sobre uma conexão TCP/IP, que garante que os dados são entregues na ordem enviada, sem perda ou duplicação. Leve em consideração condições causando uma conexão falhar.