



1. **[5 valores]** A evolução de um modelo de *Data Center* para *Cloud Computing* permite uma gestão mais eficiente do sobredimensionamento e a melhoria dos índices de ocupação de recursos (computacionais, armazenamento, comunicação e outros). O sobredimensionamento está associado à necessidade da infra-estrutura computacional responder a picos de utilização, ou a um crescimento das necessidades da organização.
 - a) Caracterize os aspectos que fundamentam a afirmação acima associando à sua descrição os conceitos de elasticidade (Elasticity), CAPEX e OPEX.
 - b) Apresente um exemplo de utilização para cada um dos modelos de disponibilização de serviços, *Infrastructure-as-a-Service* (IaaS), *Platform-as-a-Service* (PaaS) e *Software-as-a-Service* (SaaS), justificando o seu enquadramento.
2. **[5 valores]** O quadro de modularidade OSGi define um modelo de componentes e serviços para Java. As componentes podem ser dinamicamente instaladas/desinstaladas, activadas/desactivadas e actualizadas, entre outras funcionalidades.
 - a) Apresente os aspectos que considera relevantes na diferenciação entre o quadro de desenvolvimento e execução Java e o quadro de desenvolvimento e execução OSGi. Apresente os troços de código que considerar importantes para clarificar os aspectos que os distinguem e os aproximam.
 - b) Num quadro de desenvolvimento de sistemas distribuídos onde existam componentes *on-premises* e na *cloud computing*, discuta sobre o quadro de execução tendo como base Java e OSGi. Relacione a abordagem com os modelos de disponibilização de serviços (IaaS), (PaaS) e (SaaS).
3. **[5 valores]** A comunicação (*Message Oriented Middleware* - MOM) e partilha de mensagens entre serviços (modelo Linda) são aspectos chave no desenvolvimento de sistemas distribuídos.
 - a) Estabeleça um paralelo entre a utilização de um sistema de mensagens e a utilização de partilha de informação num quadro Linda, tendo como exemplo deste paradigma o serviço JINI JavaSpaces. Poderá usar uma figura para clarificar a sua resposta.
 - b) Apresente a arquitectura e os troços de código que considerar essenciais para o desenvolvimento de um exemplo onde duas aplicações Java trocam entre elas *tuplos* usando o serviço JavaSpaces (contextualize através de um exemplo simples).
4. **[5 valores]** Considere um cenário onde se pretende um serviço de reserva de lugares em Cinema com salas e, para cada sala, lugares numerados (serviço cinema). O serviço a desenvolver no quadro JINI disponibiliza as operações *estadoLugar*(Cinema, Sala, lugar) e *reservaLugar*(Cinema, Sala, lugar).
 - a) Assumindo a necessidade de uma aplicação cliente que permita reservar um determinado lugar, apresente uma estratégia de solução, devidamente fundamentada com base nos conceitos da especificação aberta X/Open referente à normalização de modelos de gestão de transacções na garantia das propriedades ACID (atomicidade, consistência, isolamento e durabilidade), fundamental em sistemas distribuídos.
 - b) Apresente os troços de código que considerar fundamental para explicitar as principais funcionalidades e serviços a que acede (e como) de uma aplicação cliente de reserva de lugares.