

# Regras para escolher o tipo de Bean

- Uso de Entity Beans
  - Quando o estado do Business Object deve ser armazenado de forma persistente e o comportamento do objeto tem a ver basicamente com o acesso ao estado
  - Para prover acesso concorrente por múltiplos clientes, quando o estado não é específico a um cliente
  - Para representar uma única linha lógica de um BD
  - Para prover robustez. Se dados de um cliente devem permanecer quando uma sessão de cliente acabou ou depois de um crash de servidor, use um Entity Bean
- Uso de Stateful Session Bean
  - Para manter estado específico a um cliente (estado conversacional). Não há compartilhamento com outros clientes
  - Para representar objetos não persistentes (objetos com curto tempo de vida e que não precisam permanecer depois de uma sessão ou após um crash de servidor)
  - Para representar o workflow entre Business Objects (para administrar as interações entre Business Objects)
- Uso de Stateless Session Bean
  - Para modelar serviços reutilizáveis (prevendo um serviço genérico para os clientes, sem manter dados específicos de um cliente)
  - Para prover alto desempenho (porém, lembre que o cliente terá que manter o estado, se necessário)
  - Para operar sobre múltiplas linhas lógicas de um BD (para representar uma visão compartilhada de dados fixos)
  - Para prover uma visão procedural dos dados (quando se foge do modelo OO e todos os dados necessários são fornecidos na entrada e todos os resultados retornados no fim do procedimento)
- Business Logic deve estar nos Session Beans
  - Para escalabilidade, prefira Stateless Session Beans
  - O uso de Stateful Beans deve ser minimizado
- Para acesso a BD, prefira Entity Beans para:
  - Ter uma visão OO dos dados
  - Não ter que tratar de aspectos transacionais, de persistência, de segurança, etc.
- Faça com que o cliente acesse apenas Session Beans
  - Esconda os Entity Beans atrás de Session Beans
  - Por quê?
    - Para possibilitar armazenar estado transacional o que não deve ser feito em Entity Beans
    - Se houver aspectos transacionais não triviais a considerar, eles podem ser tratados no Session Bean (escrito por um especialista) e o Application Developer não vai ter que saber nada sobre transações
    - Melhora a integridade dos Bancos de Dados, já que o Application Developer não vai manipular o BD diretamente. Ele só usaria Session Beans
- Há um outro tipo de bean (Message-Driven) que não veremos
  - Usado para processar mensagens assíncronas