

Entreprise JavaBeans - EJB do JavaEE

**O desaparecimento do EJB**

[[](https://www.linkedin.com/in/chmulato/)](https://www.linkedin.com/in/chmulato/)

**[Christian Mulato](https://www.linkedin.com/in/chmulato/)**

Desenvolvedor Java Sênior na Develcode

24 de março de 2024

O **Enterprise JavaBeans** (**EJB**), que costumava ser uma parte central do desenvolvimento de aplicações distribuídas em Java, passou por mudanças significativas ao longo do tempo. Vamos explorar algumas razões pelas quais o EJB perdeu parte de sua relevância:

**1. Complexidade e Overhead:**

- Nas versões anteriores do EJB, havia uma quantidade significativa de complexidade e overhead associados à configuração e gerenciamento de componentes EJB.

- Os desenvolvedores precisavam lidar com arquivos de configuração XML, anotações específicas e detalhes de transações, o que tornava o desenvolvimento mais trabalhoso.

**2. Desempenho:**

- O EJB também apresentava problemas de desempenho. A invocação de métodos remotos era feita principalmente usando CORBA (e posteriormente outros protocolos semelhantes), o que nem sempre era eficiente.

- Muitas aplicações não necessitavam de toda a funcionalidade distribuída oferecida pelo EJB, o que tornava seu uso excessivo para cenários mais simples.

**3. Evolução das Tecnologias:**

- Com o tempo, outras tecnologias surgiram e evoluíram, como o Spring Framework e a Java Persistence API (JPA).

- O Spring Framework oferecia uma alternativa mais leve e flexível para o desenvolvimento de componentes empresariais, enquanto a JPA simplificava o acesso a bancos de dados relacionais.

**4. Padrões de Design Alternativos:**

- Padrões como POJOs (Plain Old Java Objects) e Injeção de Dependência (DI) ganharam popularidade.

- Esses padrões permitiam que os desenvolvedores criassem componentes empresariais sem a necessidade de um container EJB pesado.

**5. EJB 3.x e Simplificação:**

- A especificação EJB 3.x trouxe melhorias significativas. Ela introduziu anotações, simplificou a configuração e permitiu que os desenvolvedores criassem componentes EJB mais facilmente.

- A nova versão permitia que os desenvolvedores se concentrassem na lógica de negócios, sem se preocupar tanto com detalhes de infraestrutura.

**6. Alternativas Modernas:**

- Com a ascensão de arquiteturas baseadas em ***microsserviços***, muitas empresas optaram por tecnologias mais leves e específicas para suas necessidades.

- O EJB, por sua natureza mais robusta, não se encaixava bem nesses cenários.

Em resumo, embora o EJB ainda seja usado em alguns contextos, sua relevância diminuiu à medida que outras alternativas mais leves e flexíveis surgiram no mundo do desenvolvimento de software empresarial.

Ainda existem oportunidades de trabalho relacionadas ao suporte de aplicações em EJB. Embora o EJB tenha perdido parte de sua relevância como uma escolha central para o desenvolvimento de aplicações distribuídas, muitas empresas ainda mantêm sistemas legados que utilizam essa tecnologia. Como resultado, há demanda contínua por profissionais que possam oferecer suporte, manutenção e otimização dessas aplicações.