

Desenvolvimento JavaEE seguindo uma especificação - mares seguros.

**JakartaEE - Desenvolvimento Seguro**

[[](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

5 de maio de 2024

Vamos ver aqui cinco curiosidades sobre a especificação Jakarta EE que podem enriquecer o seu conhecimento:

**1. Evolução da Plataforma Java EE:** Jakarta EE é uma evolução da plataforma Java EE, que foi transferida para a Eclipse Foundation pela Oracle. Essa mudança tem como objetivo promover a colaboração aberta e o desenvolvimento comunitário.

**2. Compatibilidade com Java SE:** Jakarta EE é projetado para ser compatível com a plataforma Java SE, permitindo que os desenvolvedores criem aplicativos empresariais robustos e escaláveis usando as APIs e tecnologias fornecidas.

**3. Foco em Micros-serviços e Nuvem:** A especificação Jakarta EE está se adaptando para atender às demandas do desenvolvimento de micro-serviços e aplicações em nuvem, oferecendo suporte para arquiteturas modernas e distribuídas.

**4. Abordagem Modular:** Jakarta EE adota uma abordagem modular, permitindo que os desenvolvedores escolham as APIs específicas de que precisam para suas aplicações, resultando em um ambiente mais leve e flexível.

**5. Comunidade Ativa:** A especificação Jakarta EE é impulsionada por uma comunidade ativa de desenvolvedores, empresas e entusiastas que contribuem para o seu desenvolvimento e evolução, garantindo um futuro promissor para a plataforma.

Jakarta EE oferece um conjunto de especificações que são úteis para o desenvolvimento de micro-serviços. Alguns dos principais componentes e especificações do Jakarta EE que são frequentemente utilizados no contexto de micro-serviços incluem:

**1. JAX-RS (Java API for RESTful Web Services):** Esta especificação é usada para criar APIs RESTful em Java, sendo uma escolha comum para o desenvolvimento de endpoints de serviços em micro-serviços.

**2. CDI (Contexts and Dependency Injection):** CDI fornece um modelo poderoso para injetar dependências e gerenciar o ciclo de vida de objetos, facilitando a construção de micro-serviços com componentes bem definidos e desacoplados.

**3. JSON-P (Java API for JSON Processing):** Para manipulação de dados JSON, o JSON-P oferece uma maneira conveniente de processar e gerar dados no formato JSON, comumente usado na comunicação entre micro-serviços.

**4. JPA (Java Persistence API):** Quando o micro-serviço precisa interagir com um banco de dados relacional, o JPA fornece uma maneira padronizada de mapear objetos Java para entidades do banco de dados, facilitando a persistência dos dados.

Quanto às bibliotecas mais recomendadas dentro do Jakarta EE para o desenvolvimento de micros-serviços, isso pode variar dependendo das necessidades específicas do projeto. No entanto, além das especificações mencionadas acima, algumas bibliotecas populares que são frequentemente utilizadas em conjunto com Jakarta EE para desenvolvimento de micros-serviços incluem:

**- MicroProfile:** Uma iniciativa que visa otimizar Jakarta EE para ambientes de micros-serviços, fornecendo um conjunto de especificações e bibliotecas voltadas para a criação e gerenciamento de micros-serviços.

**- Quarkus:** Embora não seja estritamente uma biblioteca Jakarta EE, o Quarkus é uma estrutura moderna projetada para suportar nativamente as especificações do Jakarta EE, oferecendo um ambiente otimizado para a execução eficiente de micros-serviços.

Essas são apenas algumas das opções disponíveis, e a escolha das bibliotecas mais adequadas dependerá dos requisitos específicos do projeto e das preferências da equipe de desenvolvimento.

O projeto Jakarta EE oferece várias vantagens tecnológicas e financeiras para o desenvolvimento de aplicações. Do ponto de vista tecnológico, o uso do Jakarta EE proporciona um conjunto robusto de APIs e especificações que são projetadas para facilitar o desenvolvimento de aplicações empresariais escaláveis e seguras. Isso pode resultar em redução de tempo e custos durante o desenvolvimento, já que muitas funcionalidades comuns já estão implementadas nas APIs do Jakarta EE.

Além disso, Jakarta EE oferece portabilidade, o que significa que as aplicações desenvolvidas com essa tecnologia podem ser executadas em uma variedade de ambientes de aplicação compatíveis, o que pode reduzir os custos de infraestrutura.

**Financeiramente,** o uso do Jakarta EE pode resultar em economia de custos a longo prazo, pois a plataforma é de código aberto e não requer o pagamento de licenças para ser utilizada. Isso pode ser especialmente vantajoso para empresas que buscam reduzir os custos operacionais e de manutenção das aplicações.

No geral, o Jakarta EE oferece benefícios tanto tecnológicos quanto financeiros para o desenvolvimento de aplicações empresariais, tornando-se uma escolha atraente para muitas organizações.