



# Como apps **Java** são desenvolvidos atualmente?

Por Bruna Dalmagro

Curso Java back-end

Módulo II

24/07/2023

# Etapas e desafios



Desenvolver um app de sucesso para dispositivos móveis requer considerar as tendências do mercado, oferecer suporte e atualizações frequentes, além de garantir uma experiência intuitiva para os usuários.

- É essencial ter em mente a diversidade de dispositivos no mercado e o público-alvo de cada um.
- O desenvolvimento contínuo e a priorização das necessidades dos clientes são fundamentais para otimizar a experiência do usuário. A detecção e correção de bugs são desafios comuns, sendo importante usar ferramentas de relatórios de erros.
- Além disso, é necessário escolher a tecnologia apropriada para o desenvolvimento do app, considerando suas vantagens e desvantagens.

# Tecnologias utilizadas

Vamos falar um pouco sobre as tecnologias utilizadas no mercado atualmente e quais eram usadas antigamente para esses processos, e também seus prós e contras.

Vamos começar pela mais antiga:

## JSP

O JSP, acrônimo para Java Server Pages, é uma linguagem de script de código aberto criada pela SUN que tem como principal objetivo a geração de conteúdo dinâmico para páginas da Internet. Por ser uma linguagem gratuita o JSP requer um servidor Java como Tomcat, GlassFish, JBoss para ser executado.

Ele permite combinar linguagem Java com HTML que por sua vez tem conteúdo estático, o JSP trouxe a revolução de conteúdos dinâmicos e interativos.

## Prós

- Aperfeiçoamento de recursos utilizando Java em Servlets;
- Integração com o ecossistema Java: JSP é parte da plataforma Java, permitindo uma fácil integração com outras tecnologias e frameworks da linguagem Java.
- Suporte a diversos bancos de dados, suporte a sistemas de relatórios como o Jasper Report entre outros;
- Maturidade e estabilidade: JSP é uma tecnologia madura e amplamente utilizada há muitos anos, o que significa que possui uma base sólida e estável, além de uma comunidade bem estabelecida.
- Familiaridade: Para desenvolvedores com experiência em Java, JSP pode ser uma escolha natural, pois utiliza sintaxe semelhante.
- Facilidade na manutenção: Como os códigos Java estão separados do HTML, a manutenção do código é facilitada, tornando a colaboração entre desenvolvedores mais eficiente.
- Suporte a bibliotecas e APIs Java: Como é baseado em Java, JSP pode fazer uso de todas as bibliotecas e APIs disponíveis na linguagem, permitindo uma vasta gama de funcionalidades e recursos.
- Segurança: O uso de Java como linguagem base para JSP traz consigo as práticas de segurança da plataforma Java, tornando-o uma escolha segura para aplicações corporativas.

## Contras

- Complexidade: Para tarefas simples, JSP pode acabar se tornando mais complexo do que o necessário, devido à estrutura server-side.
- Mistura de lógica e apresentação: Em projetos mal organizados, é possível misturar lógica de negócios com o código de apresentação no JSP, tornando o código menos legível e mais difícil de manter.
- Desempenho: Embora seja uma linguagem eficiente, o desempenho do JSP pode ser afetado por uma má implementação ou por lidar com muitas requisições simultâneas.
- Interação com o cliente: Por ser uma tecnologia server-side, cada requisição ao servidor precisa ser processada, o que pode resultar em maior latência e interações mais lentas com o cliente.
- Renderização no servidor: A renderização dos componentes ocorre no servidor antes de serem enviados para o cliente, o que pode levar a um maior consumo de recursos do servidor em comparação com abordagens frontend modernas, como o Angular.
- Curva de aprendizado: Para desenvolvedores com pouca experiência em Java ou familiaridade com a plataforma, a curva de aprendizado do JSP pode ser íngreme, exigindo um tempo adicional para dominar a tecnologia.

## Angular

Este é um framework Javascript de código aberto, diferentemente do JSP que é uma tecnologia da plataforma Java baseado em side-server script, também baseado em construção de páginas web dinâmicas baseado em html.

### A sintaxe Angular é baseada em TypeScript

TypeScript, que é uma linguagem de programação de alto nível desenvolvida pela Microsoft, uma superset da linguagem JavaScript, inclui todos os recursos Javascript e possui recursos adicionais como tipagem estática e recursos orientados a objetos. É uma solução de Single Page Application (SPA), o que significa que a navegação ocorre em uma única página, tornando a experiência do usuário mais rápida e suave. Possibilita a criação de interfaces de usuário complexas e ricas, com atualizações dinâmicas sem recarregar a página inteira, especialmente útil para aplicações modernas e interativas, como redes sociais, e-commerces e ferramentas de produtividade.

### Prós

- SPA (Single Page Application): Angular possibilita a criação de SPAs, o que oferece uma experiência de usuário mais fluida e rápida, sem a necessidade de recarregar a página inteira em cada interação.

- **Framework robusto:** Angular é um framework completo e poderoso, com recursos embutidos, como injeção de dependências, roteamento, gerenciamento de estados e manipulação de eventos.
- **Reutilização de código:** A abordagem baseada em componentes permite a reutilização eficiente de código, tornando o desenvolvimento mais produtivo.
- **Atualizações dinâmicas:** O uso de data binding em Angular permite atualizações dinâmicas nos dados da aplicação, refletindo instantaneamente na interface do usuário.
- **Performance:** O Angular possui recursos internos de otimização, como detecção de mudanças (change detection) inteligente, o que pode melhorar o desempenho e a eficiência da aplicação.
- **Gerenciamento de estados:** Angular oferece uma solução integrada para gerenciamento de estados através do serviço NgRx, que é uma implementação do padrão Redux no Angular. Isso ajuda a gerenciar o fluxo de dados de maneira mais previsível e consistente.
- **Suporte da comunidade:** Angular é mantido pelo Google e possui uma grande comunidade ativa, o que significa que há uma ampla gama de recursos, documentação e suporte disponíveis.

## Contras

- **Curva de aprendizado:** Angular pode ter uma curva de aprendizado íngreme para iniciantes, devido à sua complexidade e grande quantidade de recursos.
- **Tamanho da aplicação:** Como um framework completo, o tamanho da aplicação Angular pode ser maior do que em abordagens mais leves, o que pode afetar o tempo de carregamento da página.
- **Dependência do cliente:** Como Angular é executado no navegador do cliente, a desativação do JavaScript pode impedir que a aplicação funcione corretamente.
- **Atualizações frequentes:** O Angular tem um ciclo de lançamento regular, o que significa que as atualizações e mudanças frequentes podem exigir esforço adicional para manter a aplicação atualizada.
- **Dependência do framework:** Uma vez que uma aplicação é desenvolvida com Angular, ela fica fortemente ligada ao framework, o que pode dificultar a migração para outras tecnologias no futuro.
- **Abordagem mais estruturada:** Embora a organização estruturada do código seja um ponto positivo, pode ser uma desvantagem para projetos menores ou simples, onde uma abordagem mais leve e flexível pode ser preferível.

## Tanto uma quanto outra

Ambas as tecnologias possuem suas vantagens e desvantagens, oferecendo abordagens diferentes para o desenvolvimento de aplicações web.

*A escolha entre JSP e Angular dependerá dos requisitos específicos do projeto, da experiência da equipe de desenvolvimento e das preferências do desenvolvedor em relação à abordagem, estrutura e ecossistema de desenvolvimento. Ambas as tecnologias têm sido amplamente adotadas com sucesso em diferentes contextos, e a decisão deve ser tomada considerando-se todos os fatores relevantes para garantir o desenvolvimento de um aplicativo de sucesso.*



