**Framework: o que é e pra que serve essa ferramenta?**



**Jeniffer Bittencourt**

Atualizado em 16/11/2021

**COMPARTILHE**

Autores: [Cássio Murilo](https://cursos.alura.com.br/user/murilo-bass) e [Jeniffer Bittencourt](https://cursos.alura.com.br/user/jeniffer-bittencourt)

**O que é um framework?**

No início dos seus estudos, com certeza deve ter vindo à sua mente a pergunta “*Será que não existe uma forma mais fácil de fazer isso no meu projeto?*” enquanto estava escrevendo linhas e linhas de código para obter algum comportamento, ou mesmo repetindo códigos nas estruturas e até mesmo adicionando dependências para desenvolver alguma funcionalidade. É neste momento que nos deparamos com algo chamado **framework**.

O framework nada mais é do que uma ferramenta que vai te ajudar a ter como único objetivo focar em desenvolver o projeto, não em detalhes de configurações.

Vamos pensar que um **frame** representa a estrutura de uma casa, na qual só existem as paredes de tijolos levantadas ou mesmo um carro somente com a lataria, como vemos nas montadoras. Essas estruturas estão aguardando o desenvolvimento (**work**), que no exemplo da casa, corresponderia à escolha da massa corrida que será colocada nas paredes, as cores das tintas, os tipos de pisos. No caso do carro, se serão colocados bancos de tecido ou couro, qual será a cor do painel, vidros escuros ou claros e demais componentes.

Sendo assim, o framework trouxe a prática de evitar que tenhamos que fazer tarefas repetitivas, automatizando parte do trabalho. Pensando numa situação de desenvolvimento, se precisarmos criar um formulário de cadastro de usuário, ele sempre vai requerer algum tipo de validação como email e senha. O framework já terá essa validação pronta para ser utilizada.

Pode ter surgido uma voz em sua mente dizendo: "Mas posso optar por não fazer validação, ou por uma validação específica". E é verdade! Além das que o framework já disponibiliza, nada te impede de personalizar sua própria validação.

Voltando aos exemplos da casa e do carro citados acima, podemos pintar a parede da casa na cor que acharmos melhor em vez de usar a cor branca, utilizada como padrão. No caso do carro, posso colocar um painel vermelho em vez do preto que vem de fábrica, como na maioria dos carros.

**Quais as vantagens de utilizar frameworks?**

É interessante abordarmos esses fatores positivos, pois muitas vezes já as utilizamos mas ainda não sabemos exatamente onde está a vantagem dessa ferramenta. Vamos a elas:

* Sem sombra de dúvidas, quando usamos frameworks conseguimos agilizar nosso trabalho, já que **nossos esforços se voltam para o desenvolvimento, em vez de nos preocuparmos tanto com detalhes de configurações e padrões de projeto**.
* As comunidades de pessoas programadoras são as responsáveis diretas pelo desenvolvimento desses frameworks. Por essa razão eles se tornam mais seguros e com um certo padrão de código. **Devido a ser um trabalho colaborativo, cada vez mais soluções são implementadas à ferramenta**.
* Com frameworks, temos por padrão um código mais limpo, garantindo maior clareza de entendimento em tudo que é implementado pela ferramenta, o que facilita nosso trabalho e de outras pessoas que lidam com o projeto.

**Existem contras?**

Algumas pessoas desenvolvedoras apontam aspectos que podem ser ruins. São eles:

* Problemas de configurações, o que demanda tempo para a manutenção. [Tomando como exemplo o Spring](https://www.alura.com.br/artigos/spring-conheca-esse-framework-java), havia um problema comum usando MVC e devido a ele tínhamos que baixar e declarar as dependências, o módulo de injeção de dependência, as dependências dos módulos, usar o Spring Validator para validação de formulários, configurar a camada de visualização com JSP, Velocity ou Thymeleaf. Ou seja, teríamos que passar por todas essas etapas para termos o Spring MVC configurado.

**Quando trabalhamos com dependências, que são funcionalidades já prontas, precisamos lembrar de algumas questões:**

* Não podemos ignorar que o objeto gerador das dependências não está no código do projeto, mas abstraído;
* Pode ser uma dificuldade caso você não consiga fornecer as dependências diretamente no código do próprio teste;
* O framework resolve primeiro todas as dependências antes de subir a aplicação, o que consome um certo tempo;
* Conforme as dependências são inseridas no projeto, muitas configurações são baixadas sem necessidade;

**Quais frameworks utilizados nas linguagens**

Pensando em ganho de tempo e qualidade de código, segue abaixo as linguagens de programação com alguns de seus respectivos frameworks mais conhecidos:

**Javascript**:

* [Angular](https://cursos.alura.com.br/formacao-angular)
* [Vue.js](https://cursos.alura.com.br/formacao-vuejs)
* [React Js](https://www.alura.com.br/formacao-react-ts) (Apesar de existir uma polêmica dentro da comunidade por alguns dizerem que é uma biblioteca e outros afirmarem que é um framework).

**CSS**:

* [Bootstrap (de forma disparada o mais usado em CSS)](https://www.alura.com.br/artigos/bootstrap)

**PHP**:

* [Laravel](https://cursos.alura.com.br/course/laravel-produtividade-desenvolvimento-web)
* [Symfony](https://cursos.alura.com.br/course/php-symfony-api-restful), dentre outros

**Java**:

* [Spring Framework](https://cursos.alura.com.br/formacao-spring-framework)

**Python**:

* [Django](https://cursos.alura.com.br/formacao-django)

**Agile**:

* [Scrum](https://cursos.alura.com.br/course/agile-scrum): até mesmo na metodologia ágil temos o Scrum que é considerado um Framework, pois facilita o trabalho de um time na execução de tarefas durante períodos determinados.

**Como mergulhar dentro dos frameworks?**

Antes de qualquer mergulho devemos colocar nossos "pés na água". É obrigatório saber que toda utilização de framework exige um conhecimento prévio em qualquer linguagem, mesmo que seja apenas saber para quê ela é utilizada.

Muitas vezes quem está no início do aprendizado comete o erro de já partir para os estudos em frameworks sem estar com uma base inicial consolidada. Sei que pensou em algo como: "Mas preciso ser expert na linguagem X ou Y para estudar um framework? Vai demorar muito!". A resposta é ”Não!” Porém, ter uma base na qual se sinta confortável vai fazer com que seu crescimento no desenvolvimento em frameworks seja fluido e você não precise retornar ao básico do aprendizado a todo instante para esclarecer pontos simples. Sempre será necessário relembrar, mas que seu crescimento seja constante e sem travas nesta trajetória!

Para "molhar os pés" antes do mergulho em frameworks, veja essas formações:

* [Formação Front-End](https://cursos.alura.com.br/formacao-front-end)
* [Formação Java e Orientação a Objetos](https://cursos.alura.com.br/formacao-java)
* [Formação PHP](https://cursos.alura.com.br/formacao-desenvolvedor-php)
* [Formação Python](https://cursos.alura.com.br/formacao-Python-linguagem)

Gostou desse artigo e quer se aprofundar mais nesse universo?

Confira esta lista de podcasts:

* [Angular vs React](https://hipsters.tech/angular-vs-react-hipsters-142/)
* [Vue.js na Wirecard](https://hipsters.tech/vue-js-na-wirecard-hipsters-on-the-road-09/)
* [Do Front End ao React](https://hipsters.tech/do-front-end-ao-react-hipsters-ponto-tech-258/)
* [PHP não é coisas de sobrinho](https://hipsters.tech/php-nao-e-coisa-de-sobrinho-hipsters-36/)
* [Spring Framework](https://hipsters.tech/spring-framework-hipsters-87/)