



Faculdade de Tecnologia de Sorocaba  
Tecnologia em Análise e Desenvolvimento de Sistemas

## FRAMEWORKS

Prof.<sup>a</sup> Denilce Veloso  
Disciplina: Programação para WEB

Marcos Eduardo de Camargo

0030482123018

Sorocaba  
Agosto/2023

## Sumário

<b>1. Introdução.....</b>	<b>3</b>
1.1. O que é um Framework?.....	3
<b>2. Tipos de Frameworks.....</b>	<b>4</b>
2.1. Framework de Suporte.....	4
2.2. Framework de aplicação.....	4
2.3. Framework de domínio.....	4
<b>3. Quais maneiras de se utilizar um Framework?.....</b>	<b>4</b>
3.1. Inheritance-focused.....	4
3.2. Composition-focused.....	4
3.3. Hybrid.....	4
3.4. Injeção de dependência.....	5
<b>4. Frameworks Back end.....</b>	<b>5</b>
4.1. Spring.....	5
4.2. Django.....	5
4.3. Express.js.....	5
4.4. Asp.Net.....	6
4.5. Laravel.....	6
<b>5. Frameworks Front End.....</b>	<b>6</b>
5.1. AngularJS.....	6
5.2. Vue.js.....	6
5.3. Svelte.....	7
5.4. Bootstrap.....	7
5.5. Ember.js.....	7
<b>6. Conclusão.....</b>	<b>8</b>
<b>7. Referências.....</b>	<b>9</b>

## **1. Introdução**

No mundo da programação e desenvolvimento de software, os frameworks desempenham um papel fundamental ao proporcionar uma estrutura sólida e organizada para a criação de aplicações e sistemas. Um framework é essencialmente um conjunto de ferramentas, bibliotecas, convenções e diretrizes que agilizam o processo de desenvolvimento, permitindo que os desenvolvedores se concentrem mais nas funcionalidades específicas de suas aplicações e menos nas tarefas repetitivas e infraestruturais.

### **1.1. O que é um Framework?**

Framework é um conjunto de bibliotecas, que abordam funcionalidades, e estruturas, para o desenvolvimento de aplicações, a fim de fornecer soluções para um mesmo domínio de problema, permitindo a reutilização do seu código. Assim, através das diversas linguagens de programação, os Frameworks são criados e identificados pelas metodologias, propósitos, e implementações, aos quais os diversos tipos de aplicações, na maioria orientadas a objetos, poderão reutilizar suas estruturas e códigos.

Um Framework é formado por um conjunto classes implementadas em uma determinada linguagem de programação, que fornece recursos comuns já prontos, validados e testados, os quais podem ser usados para auxiliar o desenvolvimento de software, viabilizando maior eficiência na resolução dos problemas, otimização de recursos, e detecção de erros. Ele fornece um modelo de dados, usados para resolver um problema específico, abstraindo parte do código das soluções que estamos tratando.

Na programação orientada a objetos, especificamente, um Framework é um conjunto de classes e interfaces, com objetivo da reutilização de arquitetura de software, é composto por objetos, pelos mapeamentos das suas responsabilidades, que através de interfaces, com um fluxo de controle definido, possibilitará a interação com aplicações, ou seja é ele quem irá guiar à solução de um domínio específico.

## **2. Tipos de Frameworks**

### **2.1. Framework de Suporte**

Disponibiliza serviços para sistemas operacionais, e como nosso objetivo principal é a informação de dados técnicos para desenvolvimento de softwares, não será detalhado neste artigo.

### **2.2. Framework de aplicação**

Também chamado de Framework Horizontal, encapsula conhecimentos aplicáveis a vários tipos de aplicações, ou seja, eles podem ser usados em diferentes domínios, sem qualquer dependência dos mesmos. Ele resolve apenas uma parte da solução do problema, como por exemplo framework para construção de interface GUI ou interface gráfica.

### **2.3. Framework de domínio**

Também chamado de Framework Vertical, ou Framework Especialista, encapsula conhecimentos aplicáveis a aplicações com um domínio particular em comum, obtidos através de um determinado contexto específico. Ele resolve boa parte da aplicação, e temos como exemplo um Framework para aplicações de controle financeiro.

## **3. Quais maneiras de se utilizar um Framework?**

### **3.1. Inheritance-focused**

White-box ou architecture-driven, possibilitam o reuso por herança e associação dinâmica, entendendo o funcionamento do framework, ou seja, estender ou modificar a funcionalidade pela definição de sub-classes com override de métodos.

### **3.2. Composition-focused**

Black-box ou data-driven, usa a funcionalidade já presente no framework, abstraindo o código interno do Framework, que permanece inalterado, obrigando a utilização das interfaces fornecidas. O comportamento da aplicação é definido pelas instâncias e composições do Framework, tornando-se Component-Oriented (COFW).

### **3.3. Hybrid**

Gray-box, utilizam do tipo inheritance-focused com algumas funcionalidades prontas (composition-focused), possibilitando reuso por herança, associação dinâmica e definição de interfaces, possuindo boa flexibilidade e extensibilidade escondendo informações desnecessárias.

### **3.4. Injeção de dependência**

É um padrão de desenvolvimento utilizado quando é necessário manter baixo o nível de acoplamento entre diferentes módulos de um sistema, onde as dependências não são codificadas, mas configuradas através de uma estrutura de software (container), responsável por "injetar", em cada componente, as dependências declaradas. Ela se relaciona com o padrão Inversão de controle, presente nos Frameworks, fazendo a conexão entre os diferentes elementos.

## **4. Frameworks Back end**

### **4.1. Spring**

É um framework open source para a plataforma Java criado por Rod Johnson, baseado nos padrões de projeto inversão de controle (IoC) e injeção de dependência, onde o container se encarrega de instanciar as classes, e define as dependências através de um arquivo de configuração em formato XML, com módulos voltados para desenvolvimento Web, persistência, acesso remoto e programação orientada a aspectos.

### **4.2. Django**

É um framework para desenvolvimento web, prático e rápido, escrito em Python, utilizando o padrão model-template-view (MTV), inicialmente a ideia era construir um sistema para gerenciar um site jornalístico na cidade de Lawrence, no Kansas, então tornou-se um projeto de código aberto, o que favoreceu sua evolução. Ele é baseado no princípio DRY (Don't Repeat Yourself), o que possibilita aproveitamento do código, evitando repetição. É utilizado no desenvolvimento web, com uma linguagem de templates poderosa, extensível e amigável, fornecendo suporte à modelagem e persistência no banco de dados através de um ORM (Object Relational Mapping), que é feito com o uso de classes.

### **4.3. Express.js**

É um framework para desenvolvimento web, escrito em Node.js (Java Script), sendo um software livre de código aberto, foi criado por TJ Holowaychuk, e atualmente pertence a IBM, sob os cuidados da Fundação Node.js. Ele foi construído através do V8 JavaScript engine do Chrome, e serve para a criação de APIs, fornecendo recursos para criação de aplicativos web e mobile, com diversos métodos utilitários HTTP, middlewares, template engines, afim de viabilizar robustez, performance, com facilidade e praticidade no desenvolvimento.

#### **4.4. Asp.Net**

É um framework para desenvolvimento de aplicativos Web, pode ser escrito em C#, F# e Visual Basic.NET, é integrada na .NET Framework, herdando todas as suas características, portanto necessitam do Framework.NET e do servidor IIS para executar, pelo menos na plataforma Windows. Ela possibilita a reutilização do código de outros projetos escritos para a plataforma.NET, mesmo em linguagem diferente, como uma página ASP.NET escrita em VB.NET chamando componentes escritos em C# ou Web Services escritos em C++, suporta criação de APIs, e fornece três estruturas para criação de aplicativos Web: Web Forms, ASP.NET MVC e Páginas da Web do ASP.NET (Razor Pages).

#### **4.5. Laravel**

É um framework PHP, software livre, open-source, criado por Taylor B. Otwell, para o desenvolvimento de sistemas web que utilizam o padrão MVC (model, view, controller). Criado como uma alternativa mais avançada para o framework CodeIgniter, para fornecer recursos como suporte integrado para autenticação e autorização de usuário, possui uma sintaxe simples e concisa, um sistema modular com gerenciador de dependências dedicado, várias formas de acesso a banco de dados relacionais e vários utilitários indispensáveis no auxílio ao desenvolvimento e manutenção de sistemas.

### **5. Frameworks Front End**

#### **5.1. AngularJS**

É um framework JavaScript, de código aberto, desenvolvido em 2009 por Miško Hevery e Adam Abrons, para o serviço de armazenamento JSON online, é mantido pelo Google, e auxilia na execução de single-page applications. Seu objetivo é aumentar o número de aplicativos acessados por um navegador web, construído sob o padrão model-view-view-model (MVVM), para facilitar o desenvolvimento e o teste dos aplicativos. Através dele os atributos em HTML são lidos, então se executa a diretiva como uma tag, assim fazendo a interação entre a apresentação e seu modelo, essa comunicação pode ser setada manualmente, ou via JSON estático ou dinâmico.

#### **5.2. Vue.js**

É um framework JavaScript de código-aberto, criado pelo chinês Evan You, em 2013, pretendendo selecionar partes do Angular que o interessava e criar algo muito leve, focando no desenvolvimento de interfaces de usuário e single page applications. Ele fornece uma arquitetura que pode ser implementada de forma incremental, focando na renderização declarativa e na composição de componentes, estendendo elementos HTML básicos, utilizando objetos JavaScript simples e re-renderização otimizada, inclusive possuindo funcionalidades avançadas para aplicações complexas, como gestão de estados e automação de compilação.

### **5.3. Svelte**

É um web framework, para executar a compilação do front-end, de código aberto, criado por Rich Harris, em 2016, que compila templates HTML, escreve código que atualiza o DOM quando o estado do seu aplicativo é alterado, possibilitando redução do tamanho dos arquivos transferidos, melhorando o desempenho do lado do cliente, o código do aplicativo também é processado, inserindo chamadas recalculando dados automaticamente para re-renderizar elementos quando modificados.

### **5.4. Bootstrap**

É um framework web, de código aberto, desenvolvido por Mark Otto e Jacó Thornton, em 2011, originalmente desenvolvido para o Twitter, fornece recursos para desenvolvimento de componentes front-end usando HTML, CSS e JavaScript. Ele é baseado em modelos de design, visando melhorar a experiência do usuário em um site amigável e responsivo, visando o ajuste dinâmico do layout das páginas, de acordo com dispositivo utilizado, através de um kit de ferramentas front-end, com variáveis e mixins Sass, sistema de grade responsivo, componentes pré-construídos e plugins JavaScript.

### **5.5. Ember.js**

É um framework web JavaScript, de código aberto, desenvolvido por Yehuda Katz, em 2011, baseado na arquitetura Model–view–viewmodel (MVVM), também possibilita criação de aplicações para dispositivos móveis e para desktop. Ele possui o Ember CLI, que gera código automaticamente, para criação de entidades e organização dos arquivos, com um ambiente de desenvolvimento integrado com reconstruções rápidas, recarregamento automático e recursos para execução de testes.

## **6. Conclusão**

Em um cenário tecnológico cada vez mais dinâmico e exigente, os frameworks de desenvolvimento emergem como aliados indispensáveis para os desenvolvedores, proporcionando estruturas robustas e eficientes para a construção de aplicações e sistemas. Ao longo deste trabalho, exploramos a essência dos frameworks e sua influência na otimização do processo de desenvolvimento. Estes conjuntos de ferramentas não apenas aceleram a criação de software, mas também promovem a adoção de padrões de codificação coesos, a utilização de boas práticas e a reutilização de componentes, fatores cruciais para a construção de soluções confiáveis e de alto desempenho.

Os frameworks estão presentes em diversas áreas, como desenvolvimento web, front-end, mobile e até mesmo jogos. Cada um deles traz consigo uma gama de recursos predefinidos que abrangem desde a arquitetura da aplicação até questões de segurança, facilitando a vida dos desenvolvedores ao eliminar a necessidade de reinventar a roda em cada projeto. Além disso, ao aderir a um framework, os desenvolvedores podem beneficiar-se da vasta comunidade de suporte e de atualizações regulares, garantindo que suas aplicações estejam em conformidade com os mais recentes padrões e tendências da indústria.



## **7. Referências**

NORONHA, Cristiano. Disponível em <<https://balta.io/blog/o-que-e-um-framework>> Acesso em 16 Ago. 2023

OPENAI, Disponível em <<https://chat.openai.com/>> Acesso em 16 Ago. 2023