**Introdução**

Um dos temas mais importantes, e discutidos, na área do desenvolvimento de softwares, é a otimização de tempo e de recursos, como alcançá-los favorecendo a maior eficiência, garantindo qualidade, segurança, no menor espaço de tempo. Isso nos remete a um dos maiores e melhores recursos na área da programação, o Framework.

**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.

**Duas partes fundamentais de um Framework**

Hot-Spots – são as partes específicas de sistemas individuais, projetados para serem genéricos, os quais podem ser adaptados às necessidades da aplicação.

Frozen-Spots - definem a arquitetura geral de um sistema de software, seus componentes básicos e os relacionamentos entre eles, imutáveis em todas as instanciações do Framework.

**A Diferença entre um Framework e Biblioteca**

A fim de fornecer uma melhor distinção sobre um Framework, devemos esclarecer alguns fundamentos de uma Biblioteca, ou Library, com o propósito de evitar qualquer tipo de confusão entre eles, e que também nos fornecerá um melhor entendimento sobre um Framework.

Uma Biblioteca é uma coleção de implementações, escritas sob os termos de uma certa linguagem, fornecendo recursos, independentes, para determinado comportamento ser executado em uma aplicação. Assim, o programa invoca o comportamento fornecido por meio de um mecanismo da linguagem, possuindo assim o controle sobre os recursos fornecidos pela biblioteca.

Já um Framework é um conjunto de classes e interfaces que impõe um modelo colaborativo e um padrão de interação ao qual o nosso programa deverá se adaptar, as dependências entre as classes são embutidas, com a comunicação entre os objetos já definida. Então o Framework possui a responsabilidade para definir o comportamento da aplicação e chamar cada método, ou seja, é o Framework que chama o código da nossa aplicação, e é ele quem tem o controle sobre a nossa aplicação, definindo o seu comportamento e o fluxo de controle da aplicação, esse conceito é chamado de Inversão de Controle.

**Tipos de Frameworks.**

* Framework de Suporte: Disponibiliza serviços para sistemas operacionais.
* 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 deles. Ele resolve apenas uma parte da solução do problema, como por exemplo framework para construção de interface GUI ou interface gráfica.
* 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.

**5 Frameworks Frontend.**

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.

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.

Svelte: É um framework web, 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.

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.

Ember.js: É um framework web JavaScript, de código aberto, desenvolvido por Yehuda Katz, em 2011, baseado na arquitetura Model–view–view-model (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.

**Quais maneiras de se utilizar um Framework?**

* Inheritance-focused: white-box ou architecture-driven, possibilitam o reuso por herança e associação dinâmica, entendendo o funcionamento do framework, ou seja, estende ou modifica a funcionalidade pela definição de sub-classes com override de métodos.
* 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 instanciações e composições do Framework, tornando-se Component-Oriented (COFW).
* 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.
* 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.

**Conclusão**

Sendo um framework uma estrutura que serve de base para a construção de aplicações web de finalidade específica, que cujo seu desenvolvimento pode ser muito custoso, lento e problemático. Com um framework é possível construir sites, aplicativos e softwares a partir de um esqueleto pré-definido, apenas chamando as funções já definidas no framework, agilizando o desenvolvimento da aplicação e gerando maior confiabilidade no projeto.