

Trabalho Desenvolvimento Web

Docente: Prof Moacir Degasperi JR

Discente:

Rodrigo Luis Tavano Bosso – PC3005623

Frame Works Foundation & Angular



0-Resumo:

O seguinte trabalho tem como objetivo explicar, demonstrar exemplos e funcionamento do que são frameworks. Serão apresentados os Frameworks Foundation e Angular. Essa produção trabalhará tanto algumas características mais genéricas dos mesmos; retratando, para isso, parte da história ,objetivo e características; quanto aspectos específicos desses, retratando nesse ponto parte dos funcionamentos para o gerenciamento, demonstrando um pouco sobre os frameworks de HTML e Javascript famosos do mercado.

Para que se possa realizar tal discussão, estaremos fundamentados em uma série de pesquisas em links e livros, assim como os sites oficiais dos mesmos que explicam o funcionamento geral dos Frameworks. Consoante a tais fatores partiremos para o texto propriamente dito.

Palavras-chave: Frameworks, Foundation, Angular, HTML, Javascript.



Sumário:

0-Resumo:	2
Sumário:	3
1- Introdução:	7
1.1- O que é um Framework?	7
1.2- Características dos Frameworks:	8
2- Diferença com relação a Bibliotecas:	9
3- Vantagens de Frameworks:	10
3.1- Redução de tempo:	10
3.2 - Segurança:	10
3.3- Padrões de legibilidade:	10
4- Desvantagens de Frameworks:	11
4.1- Problemas de configuração:	11
4.2- Dependência:	11
4.3- Códigos desnecessários:	11
5 – Framework HTML- Foundation:	12
5.1- Apresentação Foundation:	12
5.2- Características e ganhos do Fondation 6 :	13
5.2.1- Código enxuto:	13
5.2.2- Acessibilidade:	13
5.2.3- Códigos criativos compartilhados:	13
5.2.4- Padrões de navegação flexíveis:	13
5.2.5- Utilitários modulares de JavaScript:	13
5.2.6- Estilos de base CSS:	13
5.2.7- UI de movimento:	14
5.2.8- Início rápido do projeto:	14
5.2.9- Grade Flexbox opcional:	14
5.2.10- Grade de massa personalizável:	14
5.2.11- Pilha de desenvolvimento do ZURB:	14
6- Framework Javascript- Angular:	15
6.1- Apresentação Angular:	15
6.2- Features- Angular	16
6.2.1- Aplicativos progressivos da Web:	16
6.2.2- NativeScript:	16
6.2.2- Geração de código:	16
6.2.4- Universal:	16
6.2.5- Divisão de código:	16
6.2.6- Modelos:	16



6.2.7- CLI angular:	16
6.2.8- IDEs:	16
6.2.9- Testes:	17
6.2.10- Animação:	17
6.2.11- Acessibilidade:.....	17
6.3- IDEs do Angular:.....	17
6.3.1- MetaMagic :.....	17
6.3.2- IDE angular por Webclipse:.....	17
6.3.2- IntelliJ IDEA:.....	17
6.3.2- Código do Visual Studio:.....	17
6.3.2- WebStorm:	18
6.4- Ferramentas - Angular:	18
6.4.1- CLI angular:	18
6.4.2- ESLint angular:.....	18
6.4.3- Parque Angular:.....	18
6.4.4- Universal angular:	18
6.4.5- Pré-renderização angular:	18
6.4.6- Compodoc:	18
6.4.7- Servidor de Lite:	19
6.4.8- Ng-mocks:	19
6.4.9- Nx - Sistema de construção inteligente, rápido e extensível:.....	19
6.4.10- Espião observador:.....	19
6.4.11- Protrator:.....	19
6.4.12- Scully:.....	19
6.4.13- Padaria da interface do usuário:	19
6.5- Bibliotecas de dados do Angular:.....	20
6.5.1- Akita:	20
6.5.2- Fogo angular:.....	20
6.5.3- Apolo:.....	20
6.5.4- ButterCMS:.....	20
6.5.5- Formamente:	20
6.5.6- Meteoro:	20
6.5.7- ngrx:.....	20
6.5.8- ngx-api-utils:	20
6.5.9- NGXS:.....	21
6.5.10- rx-angular:.....	21
6.5.11- Validadores de formulários reativos da RxWeb:.....	21
6.6- Componentes e interface do Angular:.....	21
6.6.1- ag-Grid:.....	21



6.6.2- Ayle UI:.....	21
6.6.3- Amexio - Extensões angulares:	22
6.6.4- Projeto angular e material 2.0 para Bootstrap 5:.....	22
6.6.5- Projeto angular e material para Bootstrap 4:	22
6.6.6- Componente angular do gráfico de Gantt:.....	22
6.6.7- Material angular:	22
6.6.8- Kit de ferramentas angular da interface do usuário:.....	22
6.6.9- Angular-Slickgrid:	22
6.6.10- Design de formigas móvel de angular (ng-zorro-antd-móvel) :.....	23
6.6.11- Projeto de formiga de angular (ng-zorro-antd) :.....	23
6.6.12- Serra de vaivém (chinês) :	23
6.6.13- Material Blox:	23
6.6.14- Componentes de carbono angulares:.....	23
6.6.15- Sistema de Design de Clareza:	23
6.6.16- CoreUI para Angular:	23
6.6.17- DevExtreme:	23
6.6.18- JS essencial 2:.....	24
6.6.19- FancyGrid:.....	24
6.6.20- Ignite a interface do usuário para Angular:	24
6.6.21- jqWidgets:.....	24
6.6.22- Kendo UI:	24
6.6.23- Componentes da comunidade material:.....	24
6.6.24- adaptador material-dayjs:.....	24
6.6.25- Mosaic - Angular UI Components:.....	24
6.6.26- Nebular:	25
6.6.27- ng-bootstrap:	25
6.6.28- ng-relâmpago:.....	25
6.6.29- Serviço de Cookies NGX:	25
6.6.30- Carregador Esqueleto NGX:.....	25
6.6.31- ngx-bootstrap:.....	25
6.6.32- ngx-smart-modal:	25
6.6.33- NGX-YOUI:.....	25
6.6.34- Onsen UI:.....	26
6.6.35- PO UI:	26
6.6.36- Faces Primeiros:	26
6.6.37- Sencha para Angular:.....	26
6.6.38- UI de qualidade simples:	26
6.6.39- Componentes inteligentes da Web:.....	26
6.6.40- UI de Taiga:	26



6.6.41- Verdadeiramente UI:	27
6.6.42- Vaadin:	27
6.6.43- Wijmo:	27
6.7- Desenvolvimento entre plataformas com Angular:.....	27
6.7.1- Elétron:	27
6.7.2- Iônico:.....	27
6.7.3- MiniPrograma:	27
6.7.4- NativeScript:	27
7- Conclusão:.....	28
8- Referências Bibliográficas:	29
FIM	30



1- Introdução:

1.1- O que é um Framework?

Framework é uma definição criada no mercado de software que abrange outros contextos, em suma, Frameworks são uma série de ações e estratégias que visam solucionar um problema bem específico. Assim, quando se deparam com esse cenário, os profissionais recorrem a um conjunto pronto de abordagens e otimizam os seus resultados. Quando nós voltamos a criação de aplicações, Frameworks trata-se de uma série de bibliotecas e classes ou códigos prontos, que oferecem alguma funcionalidade específica.

Simplificando, são funções ou outros pedaços de códigos que podem ser inseridos nos códigos compatíveis com as versões de cada framework. Esses códigos apresentam uma função específica e só funcionam dentro do contexto inteiro, por isso ajudam quando o programador precisa economizar o tempo do conserto de alguma função/tread/objeto defeituosa/o.

A automação de tarefas repetitivas é facilitada com o uso dos frameworks, uma vez que as funções já estão relativamente prontas, para softwares de diferentes naturezas. Assim, o programador pratica o reuso de código.

Um exemplo famoso de funcionalidade que é implantada com frameworks: formulário de login. As ações que envolvem essa função raramente mudam a depender do contexto da aplicação — um formulário de login quase sempre requer as mesmas coisas, que é um usuário/e-mail e uma senha, por exemplo.



1.2- Características dos Frameworks:

A principal função de um framework é facilitar o processo de desenvolvimento de um software ou aplicação. Ao oferecerem uma estrutura básica sobre a qual o sistema pode ser programado, os frameworks representam uma vantagem em termos de tempo e segurança.

Como são criados por equipes de desenvolvedores experientes e por contarem com comunidades ativas, os frameworks muitos dos quais são de código aberto tendem a passar por testes e processos de otimização com certa regularidade.

Assim reduzindo drasticamente a probabilidade de falas ou bugs, também contando com uma equipe de apoio que responde os reports realizados caso um novo bug for encontrado.

Por isso, alguns dos principais motivos pelos quais frameworks são amplamente utilizados no mundo da programação hoje em dia, são:

- Otimizar o tempo de desenvolvimento
- Definir e padronizar as melhores práticas de programação
- Oferecer maior segurança
- Evitar códigos duplicados
- Diminuir a ocorrência de bugs
- Gerar maior consistência no processo de desenvolvimento e nas aplicações criadas
- Reduzir as chances de erro no código
- Possibilitar que os desenvolvedores se dediquem aos elementos específicos do projeto
- Simplificar e encurtar a curva de aprendizado do time de desenvolvimento
- Poupar custos



2- Diferença com relação a Bibliotecas:

“ Um framework consiste basicamente em elementos de código que guiam o desenvolvimento de uma aplicação, otimizando esse processo. Por causa disso, ele às vezes pode ser confundido com uma biblioteca. Mas estes são dois conceitos distintos utilizados para alcançar objetivos diferentes.

Um framework representa a estrutura dentro da qual você desenvolverá um software. Assim, seu código deve desde o princípio seguir os padrões estabelecidos pelo framework. Enquanto uma biblioteca representa recursos que poderão ser utilizados por você no decorrer do desenvolvimento, fornecendo elementos para completar uma etapa do desenvolvimento ou otimizá-lo.

Ilustrando através de um exemplo: uma empresa utiliza funções curl, que é uma biblioteca PHP. Quando utiliza-se uma das funções curl, o código PHP solicita essa função particular dentro da biblioteca em questão. É o código PHP que “chama” função curl dentro da biblioteca.

Agora , caso utiliza-se um framework PHP como, por exemplo, o Laravel. A relação de controle será invertida, pois o framework é quem “chama” o código escrito nele.

É importante lembrar que, dentro de um framework, podem estar contidas uma ou mais bibliotecas. Afinal, elas são um dos muitos recursos inteligentes presentes em um framework.”

<https://www.hostinger.com.br/tutoriais/frameworks>



3- Vantagens de Frameworks:

3.1- Redução de tempo:

Uma das principais vantagens são a agilidade e economia do tempo. Uma vez que essas ferramentas facilitam o desenvolvimento dos sistemas, permitindo que os programadores não percam tempo com funcionalidades mais básicas.

Desse modo, eles podem investir recursos nas funções próprias do sistema, que o diferencia de outros já existentes, melhorando assim, a produtividade pois o tempo e esforço dos profissionais serão melhor direcionados. Isso, no geral, contribui para resultados melhores e maior satisfação dos clientes.

3.2 - Segurança:

Os frameworks são desenvolvidos por comunidades de programadores ou grandes empresas como a Google e Microsoft. Geralmente são abertos para modificações e revisões por esses profissionais. Dessa maneira, eles são consistentes e robustos, apresentando um bom padrão de qualidade de código.

Em razão disso, ao aplicá-los, ocorre um ganho de segurança para o sistema, com menos erros. Sendo possível utilizar esses padrões prontos sem receio e concentrar mais nos códigos originais.

3.3- Padrões de legibilidade:

Outra vantagem, é que por ser desenvolvido por uma comunidade ocorre a validação da legibilidade. Esse quesito é essencial para a construção de código, sendo um dos aspectos mais cobrados em cursos e graduações da área. Afinal, um código limpo é importante para que programadores consigam realizar manutenções e entender do que se trata quando leem.

Os frameworks são geralmente criados sob esse padrão e garantem uma clareza muito grande que auxilia no dia a dia.



4- Desvantagens de Frameworks:

4.1- Problemas de configuração:

Apesar de todos os pontos positivos, os frameworks podem ser complexos para configurar em alguns casos. Por isso, é mais difícil adaptá-lo ao que já está implementado e garantir a comunicação deles com outras partes do sistema que está sendo criado. Geralmente nesses casos as instalações e configurações costumam ser mais trabalhosas e consumir muito tempo antes mesmo do desenvolvedor conseguir usar os códigos.

4.2- Dependência:

Outro problema do uso exagerado de funcionalidades prontas é a dependência. Usar muitos deles é como construir um sistema com partes distintas criadas e modificadas externamente. Com isso pode ocorrer de haver sistemas dependentes completamente de mais de um único tipo de framework.

O problema surge caso o framework sofra algum problema, o sistema no qual ele foi utilizado também é afetado.

4.3- Códigos desnecessários:

Existem frameworks mais pesados que buscam resolver problemas maiores, sendo usados por várias empresas de diversas formas. Contudo, eles podem ser de instalação complexa, como já comentado, por vezes trazem mais códigos e especificações do que o necessário. Logo, o código fica cheio de configurações que não são importantes no contexto.



5 – Framework HTML- Foundation:



Foundation
Start here, build everywhere.

5.1- Apresentação Foundation:

Fondation ou “ Fundação” é um framework de estrutura frontal responsiva . Gera a estrutura para qualquer dispositivo, por meio de acessibilidade.

Em síntese, o Foundation é uma família de estruturas responsivas de front-end que facilitam o design de sites, aplicativos e e-mails responsivos e bonitos que ficam incríveis em qualquer dispositivo.

O Foundation é semântico, legível, flexível e completamente personalizável. Os criadores estão constantemente adicionando novos recursos e trechos de código, incluindo Modelos HTML para ajudar os usuários do framework.

Atualmente na versão 6 ele já conta com diversas features para auxiliar os programadores.

Assim o “Foundation for Sites 6” foi projetado para levar seu projeto do protótipo à produção com mais eficiência, incluindo uma ampla gama de componentes modulares e flexíveis que são facilmente estilizados. Esses blocos de construção versáteis e leves facilitam a vida de diversos programadores de desenvolvimento Web.



5.2- Características e ganhos do Fondation 6 :

5.2.1- Código enxuto:

O tamanho total de arquivos de cada componente e classe agora pesa em 60 KB CSS e 84 KB JS, com muito espaço para torná-los ainda menores ao remover seletivamente componentes não utilizados.

5.2.2- Acessibilidade:

Todos os trechos de código vêm com atributos e funções do ARIA, juntamente com instruções sobre como usar adequadamente esses componentes. Isso ajuda a garantir que todos os sites criados no Foundation 6 possam ser usados em qualquer lugar, em qualquer dispositivo, por qualquer pessoa.

5.2.3- Códigos criativos compartilhados:

Carregue suas páginas da web de design responsivo para obter feedback contextual sobre qualquer ponto de interrupção. As partes interessadas e colaboradores podem anotar e revisar páginas da web codificadas em qualquer dispositivo. Leve seus projetos da Fundação do protótipo para a produção usando Código Notável.

5.2.4- Padrões de navegação flexíveis:

Sistema de menus completamente personalizável e modular. O novo componente Menu é um grande salto em frente, pois combina vários componentes.

5.2.5- Utilitários modulares de JavaScript:

Cria-se mais bibliotecas de utilitários compartilhados. As bibliotecas de utilitários são acessíveis ao público para que você possa criar seus próprios plugins incríveis.

5.2.6- Estilos de base CSS:

Os estilos de base atuam como uma estrutura de arame codificada, e não como um design final. Os estilos CSS mais simples permitem modificá-los mais facilmente.



5.2.7- UI de movimento:

A Motion UI é a nossa biblioteca de animação baseada em Sass que ajudará diversos desenvolvedores, a adicionar movimentos significativos aos seus sites. Esta biblioteca Sass inclui mais de duas dúzias de aulas de transição e animação integradas. É muito personalizável e super fácil criar o seu próprio com os mixins Sass.

5.2.8- Início rápido do projeto:

A ferramenta de linha de comando (CLI) permite configurar projetos em branco do Foundation for Sites, Apps ou E-mails com menos dependências do que antes. Pode ser instalada através do NPM, Bower, Meteor ou Composer.

5.2.9- Grade Flexbox opcional:

Para usuários que podem adotar a mais nova tecnologia, a Foundation vem com uma grade opcional baseada em Flexbox. Com opções ainda diversas de pedidos e alinhamento de fontes.

5.2.10- Grade de massa personalizável:

Os mixins de grade Sass foram feitos com inteligência para oferecer ainda mais flexibilidade para personalizar as grades com qualquer número de colunas.

5.2.11- Pilha de desenvolvimento do ZURB:

Modelo que o ZURB usa em todos os projetos. Esse modelo inicial é uma pilha cheia de um gerador de site estático personalizado para ajudar a achatar arquivos em documentos HTML únicos. Ele compila seu Sass, fornece modelos de guidão, UglifyJS, UnCSS e compactação de imagem.



6- Framework Javascript- Angular:



6.1- Apresentação Angular:

O Angular é um framework utilizado para a criação de Single-Page Applications, que é uma aplicação web consumida em uma única página. Gerenciado e administrado hoje pela gigante Google, o Angular é uma das mais poderosas ferramentas de frameworks disponíveis para desenvolvedores, hoje, sendo implementada em sites como: YouTube, Netflix, Upwork, Lego, IBM, Istock, Freelancer, Wikiwand, entre muitos outros.

O SPA funciona comparando o processamento com o de uma aplicação web comum: Quando uma página comum é carregada e a pessoa usuária faz qualquer solicitação de atualização de dados, como uma busca por produto, o navegador faz uma requisição ao servidor para processar essa busca e retornar os dados solicitados. Nesse momento, então, é feito o recarregamento de toda a página. Em uma página SPA toda essa comunicação é feita de uma forma mais dinâmica.

Com esse método a atualização é feita dinamicamente e apenas referente a solicitação em questão, sem o recarregamento total da página, reduzindo assim a quantidade de dados trafegados, o tempo de processamento e elimina a necessidade de recarregar a página inteira sempre que qualquer requisição for feita.

Dessa forma, a experiência da pessoa usuária fica mais fluída, ou seja, a comunicação entre o navegador e o servidor é praticamente transparente para ela. Isso é possível uma vez que o Angular facilita essa operação de forma simples e bem estruturada.



6.2- Features- Angular

6.2.1- Aplicativos progressivos da Web:

Possibilita o usuário a ter acesso a recursos modernos de plataforma da web para oferecer experiências semelhantes a aplicativos. Instalação de alto desempenho, offline e passo zero.

6.2.2- NativeScript:

Melhora o desempenho com aplicativos móveis nativos com estratégias de Cordova, Ionic ou NativeScript.

6.2.2- Geração de código:

O Angular transforma modelos em código altamente otimizado para as máquinas virtuais JavaScript atuais, oferecendo todos os benefícios do código escrito à mão com a produtividade de uma estrutura.

6.2.4- Universal:

Possibilita uma primeira visualização do aplicativo no Node.js®, .NET, PHP e outros servidores para renderização quase instantânea em apenas HTML e CSS. Também abre caminho para sites que otimizam para SEO.

6.2.5- Divisão de código:

Os aplicativos angulares são carregados rapidamente com o novo Component Router, que fornece divisão automática de código para que os usuários carreguem apenas o código necessário para renderizar a visualização solicitada.

6.2.6- Modelos:

Possibilita criar rapidamente visualizações de interface do usuário com sintaxe de modelo simples e poderosa.

6.2.7- CLI angular:

Ferramentas da linha de comando: construir rapidamente, adicionar componentes e testes e implantar instantaneamente.

6.2.8- IDEs:

Conclusão inteligente do código, erros instantâneos e outros comentários em editores e IDEs populares.



6.2.9- Testes:

Com o Karma para testes de unidade, possibilita saber se algo parou de funcionar, toda vez que economiza.

6.2.10- Animação:

Criar coreografias complexas e de alto desempenho e cronogramas de animação com muito pouco código através da API intuitiva da Angular.

6.2.11- Acessibilidade:

Criar aplicativos acessíveis com componentes habilitados para ARIA, guias do desenvolvedor e infraestrutura de teste integrada a11y.

6.3- IDES do Angular:

6.3.1- MetaMagic :

O Amexio Canvas é um ambiente de arrastar e soltar para criar aplicativos HTML5 / angulares de dispositivos inteligentes e Web totalmente responsivos. O código será gerado automaticamente e implantado a quente pelo Canvas para testes ao vivo. Fora da caixa 50+ Suporte ao tema Design de material. Confirme seu código no repositório público ou privado do GitHub.

6.3.2- IDE angular por Webclipse:

Construído em primeiro lugar para Angular. Configuração chave na mão para iniciantes; poderoso para especialistas.

6.3.2- IntelliJ IDEA:

Java capaz e ergonômico * IDE

6.3.2- Código do Visual Studio:

O VS Code é uma ferramenta gratuita e leve para editar e depurar aplicativos da Web.



6.3.2- WebStorm:

IDE leve, porém poderoso, perfeitamente equipado para desenvolvimento complexo do lado do cliente e do servidor com o Node.js

6.4- Ferramentas - Angular:

6.4.1- CLI angular:

A CLI angular oficial facilita a criação e o desenvolvimento de aplicativos desde o commit inicial até a implantação da produção. Ele já segue nossas melhores práticas imediatamente!

6.4.2- ESLint angular:

A ferramenta que permite ao ESLint restringir projetos angulares.

6.4.3- Parque Angular:

Ambiente de desenvolvimento de interface do usuário para construção, teste e documentação de aplicativos angulares.

6.4.4- Universal angular:

Renderização do servidor para aplicativos angulares.

6.4.5- Pré-renderização angular:

Uma ferramenta de linha de comando para pré-renderizar aplicativos angulares que usa o Angular Universal, mas também suporta plugins Scully.

6.4.6- Compodoc:

Esta ferramenta gera documentação dedicada para aplicativos angulares.



6.4.7- Servidor de Lite:

Desenvolvimento leve apenas servidor Node.js®

6.4.8- Ng-mocks:

Uma biblioteca de testes para criar serviços, componentes, diretrizes, tubos e módulos simulados a partir de dependências irritantes nos testes de unidade.

6.4.9- Nx - Sistema de construção inteligente, rápido e extensível:

O Nx é um sistema de compilação de código aberto que apresenta muitos recursos monorepo que ajudam as equipes empresariais a desenvolver o Angular em escala.

6.4.10- Espião observador:

Uma biblioteca para testar observáveis do RxJS que oferece uma alternativa aos testes de mármore.

6.4.11- Protrator:

A estrutura oficial de teste de ponta a ponta para aplicações angulares.

6.4.12- Scully:

Scully (Jamstack Toolchain for Angular) torna a construção, teste e implantação de aplicativos Jamstack extremamente simples.

6.4.13- Padaria da interface do usuário:

A UI Bakery é uma plataforma de código baixo para gerar aplicativos angulares.



6.5- Bibliotecas de dados do Angular:

6.5.1- Akita:

O Akita é um padrão de gerenciamento de estado, construído sobre o RxJS, que leva a idéia de vários armazenamentos de dados do Flux e as atualizações imutáveis do Redux, juntamente com o conceito de streaming de dados, para criar o modelo de armazenamento de dados observável.

6.5.2- Fogo angular:

A biblioteca oficial do Firebase e Angular.

6.5.3- Apolo:

Apollo é uma pilha de dados para aplicativos modernos, criados com o GraphQL.

6.5.4- ButterCMS:

O ButterCMS é um CMS sem cabeça que permite gerenciar o conteúdo usando nosso painel e integrá-lo ao seu aplicativo angular.

6.5.5- Formamente:

Formamente é uma biblioteca de formulários dinâmica (alimentada por JSON), construída sobre os Formulários Reativos Angulares.

6.5.6-Meteoro:

Use Angular e Meteor para criar aplicativos JavaScript completos para dispositivos móveis e desktop.

6.5.7- ngrx:

Extensões reativas para angular.

6.5.8- ngx-api-utils:

O ngx-api-utils é uma biblioteca enxuta de utilitários e auxiliares para integrar rapidamente qualquer API HTTP (REST, Ajax e qualquer outra) com o Angular.



6.5.9- NGXS:

NGXS é um padrão de gerenciamento de estado + biblioteca para Angular. O NGXS é modelado após o padrão CQRS implementado popularmente em bibliotecas como Redux e NgRx, mas reduz o boilerplate usando recursos modernos do TypeScript, como classes e decoradores.

6.5.10- rx-angular:

O RxAngular é uma biblioteca reativa que fornece extensões para o desenvolvimento de aplicativos de alto desempenho e escaláveis. Ele aborda os problemas do gerenciamento local do estado, desempenho da renderização e arquitetura reativa.

6.5.11- Validadores de formulários reativos da RxWeb:

Os validadores de formulários reativos da RxWeb fornecem todos os tipos de validação complexa, condicional, de campo cruzado e dinâmica em formulários reativos baseados em validador, formulários reativos baseados em modelo e formulários orientados por modelo.

6.6- Componentes e interface do Angular:

6.6.1- ag-Grid:

Um datagrid para Angular com recursos de estilo corporativo, como classificação, filtragem, renderização personalizada, edição, agrupamento, agregação e pivotação.

6.6.2- Alyle UI:

Design mínimo, um conjunto de componentes para Angular.



6.6.3- Amexio - Extensões angulares:

O Amexio é um rico conjunto de componentes angulares equipados com HTML5 e CSS3 para Design Web responsivo e mais de 80 temas de design de materiais incorporados. Amexio tem 3 edições, padrão, corporativa e criativa. O Std Edition consiste em componentes básicos da interface do usuário, que incluem grade, guias, entradas de formulário e assim por diante. Enquanto o Enterprise Edition consiste em componentes como Calendário, Guias em Árvore, Logins de Mídia Social (Facebook, GitHub, Twitter e assim por diante) e o Creative Edition está focado na criação de sites elegantes e bonitos. Com mais de 200 componentes / recursos. Todas as edições são de código aberto e gratuitas, com base na Licença Apache 2.

6.6.4- Projeto angular e material 2.0 para Bootstrap 5:

Componentes, modelos e plugins gratuitos para o Bootstrap 5 mais recente, com o estilo de acordo com o Material Design 2.0 (diretrizes introduzidas em 2020)

6.6.5- Projeto angular e material para Bootstrap 4:

Componentes, modelos e plugins gratuitos para o Bootstrap 4 mais recente, com o estilo de acordo com o Material Design (diretrizes introduzidas em 2015)

6.6.6- Componente angular do gráfico de Gantt:

Um componente moderno e poderoso do gráfico de gantt para Angular

6.6.7- Material angular:

Componentes de design de material para angular

6.6.8- Kit de ferramentas angular da interface do usuário:

Kit de ferramentas angular da interface do usuário: 115 componentes da interface do usuário com manutenção profissional, variando de uma grade robusta a gráficos e muito mais. Experimente gratuitamente e crie aplicativos angulares mais rapidamente.

6.6.9- Angular-Slickgrid:

Angular-SlickGrid é um invólucro da biblioteca de datagrid SlickGrid, rápida e personalizável, com temas do Bootstrap 3,4



6.6.10- Design de formigas móvel de angular (ng-zorro-antd-móvel) :

Um conjunto de componentes de interface do usuário móvel de classe empresarial baseados no Ant Design Mobile e Angular

6.6.11- Projeto de formiga de angular (ng-zorro-antd) :

Um conjunto de componentes de interface do usuário de classe empresarial baseados em Ant Design e Angular

6.6.12- Serra de vaivém (chinês) :

O quebra-cabeças fornece um conjunto de componentes da Web baseados no Angular. Ele está apoiando o desenvolvimento de todas as aplicações do Big Data Product of ZTE (<https://www.zte.com.cn>).

6.6.13- Material Blox:

Uma biblioteca leve de Design de materiais para Angular, baseada nos Componentes de materiais do Google para a Web

6.6.14- Componentes de carbono angulares:

Uma implementação angular do Carbon Design System para IBM

6.6.15- Sistema de Design de Clareza:

Diretrizes UX, estrutura HTML / CSS e componentes angulares trabalhando juntos para criar experiências excepcionais

6.6.16- CoreUI para Angular:

Biblioteca de componentes de interface do usuário de código aberto para criar aplicativos modernos, bonitos e responsivos na metade do tempo com componentes de interface do usuário de alto desempenho e fáceis de personalizar.

6.6.17- DevExtreme:

Mais de 50 componentes da interface do usuário, incluindo grade de dados, grade de pivô, agendador, gráficos, editores, mapas e outros controles multiuso para criar aplicativos da Web altamente responsivos para dispositivos de toque e desktops tradicionais.



6.6.18- JS essencial 2:

Essential JS 2 for Angular é uma coleção moderna de componentes angulares verdadeiros baseados em TypeScript. Ele tem suporte para compilação Ahead Of Time (AOT) e Tree-Shaking. Todos os componentes são desenvolvidos desde o início para serem leves, responsivos, modulares e amigáveis ao toque.

6.6.19- FancyGrid:

Biblioteca de grade angular com integração de gráficos e comunicação com servidor para Enterprise.

6.6.20- Ignite a interface do usuário para Angular:

O Ignite UI for Angular é um kit de ferramentas angular livre de dependência para a criação de aplicativos da web modernos.

6.6.21- jqWidgets:

Componentes angulares da interface do usuário, incluindo grade de dados, grade de árvores, grade de pivô, agendador, gráficos, editores e outros componentes multiuso

6.6.22- Kendo UI:

Uma biblioteca de nível profissional de componentes angulares da interface do usuário, escrita em TypeScript, que inclui nossa grade de dados, TreeView, gráficos, editores, DropDowns, DatePickers e muito mais. Os recursos incluem suporte para compilação AOT, tree shaking para alto desempenho, localização e acessibilidade.

6.6.23- Componentes da comunidade material:

Componentes materiais fabricados pela comunidade que incluem Color Picker, Timer Picker, Speed Dial e Scrollspy.

6.6.24- adaptador material-dayjs:

Uma implementação DayJS do DateAdapter @angular / material que resulta em tamanhos de pacote menores que o seu homólogo MomentJS.

6.6.25- Mosaic - Angular UI Components:

Positive Technologies UI componentes baseados em Angular.



6.6.26- Nebular:

Sistema de temas, componentes de interface do usuário, autenticação e segurança para seu próximo aplicativo Angular.

6.6.27- ng-bootstrap:

A versão Angular da biblioteca Angular UI Bootstrap. Esta biblioteca está sendo construída do zero no Typescript usando o framework Bootstrap 4 CSS.

6.6.28- ng-relâmpago:

Componentes e diretivas nativos do Angular para o Lightning Design System.

6.6.29- Serviço de Cookies NGX:

O NGX Cookie Service é uma Biblioteca Angular que pode ser usada para ler, definir e excluir cookies do navegador.

6.6.30- Carregador Esqueleto NGX:

O NGX Skeleton Loader é uma solução leve e amigável para A11Y para esqueletos de carregamento rápido que se adaptam automaticamente aos seus aplicativos Angular.

6.6.31- ngx-bootstrap:

Diretivas nativas angulares para Bootstrap.

6.6.32- ngx-smart-modal:

Manipulador modal angular inteligente, leve e rápido para gerenciar modais e dados em qualquer lugar.

6.6.33- NGX-YOUI:

NGX-YOUI é uma biblioteca de interface do usuário angular que pode ser altamente personalizada com conteúdo e estilos. Você pode integrar facilmente com outras estruturas CSS e biblioteca de interface do usuário, como Tailwind, Bootstrap, PrimeNG, Angular Material etc.



6.6.34- Onsen UI:

Componentes da interface do usuário para aplicativos móveis híbridos com ligações para Angular e AngularJS.

6.6.35- PO UI:

PO UI, Componentes da interface do usuário brasileira, focada na facilidade e produtividade de aplicações angulares.

6.6.36- Faces Primeiros:

PrimeNG é uma coleção de componentes ricos da interface do usuário para Angular.

6.6.37- Sencha para Angular:

Crie aplicativos da Web modernos mais rapidamente com mais de 115 componentes de interface do usuário pré-criados. Experimente gratuitamente e faça o download hoje.

6.6.38- UI de qualidade simples:

A interface do usuário de qualidade simples (SQ-UI) é um kit de interface do usuário flexível e facilmente personalizável, com o objetivo de fornecer a máxima eficiência com o mínimo possível de sobrecarga. Impulsionado pela ideia de que deve ser estritamente "para desenvolvedores por desenvolvedores", cada nova versão do recurso inclui funcionalidades exigidas pelos desenvolvedores que a estão usando.

6.6.39- Componentes inteligentes da Web:

Componentes da Web para angular. Componentes angulares sem dependência para a criação de aplicativos da Web modernos e compatíveis com dispositivos móveis.

6.6.40- UI de Taiga:

O Taiga UI é um kit de interface do usuário angular totalmente acessível, composto por várias bibliotecas de base e vários complementos. Temos mais de 130 componentes, mais de 100 diretivas, dezenas de tokens, utilidades e ferramentas. Eles são facilmente personalizáveis e bem projetados.



6.6.41- Verdadeiramente UI:

O TrulyUI é um Quadro de interface do usuário angular desenvolvido especialmente para aplicativos de desktop baseados em componentes da Web que usam as maiores tecnologias do mundo.

6.6.42- Vaadin:

O design de materiais inspirou componentes da interface do usuário para a criação de ótimos aplicativos da web. Para celular e desktop.

6.6.43- Wijmo:

Controles de interface do usuário de alto desempenho com o suporte angular mais completo disponível. Os controles do Wijmo são todos escritos no TypeScript e têm zero dependências. O controle FlexGrid inclui marcação declarativa completa, incluindo modelos de célula.

6.7- Desenvolvimento entre plataformas com Angular:

6.7.1- Elétron:

Plataforma de elétrons para angular.

6.7.2- Iônico:

A Ionic oferece uma biblioteca de componentes e ferramentas HTML, CSS e JS otimizados para dispositivos móveis para criar aplicativos Web nativos e progressivos altamente interativos.

6.7.3- MiniPrograma:

Plataforma MiniProgram para Angular.

6.7.4- NativeScript:

O NativeScript é como você cria aplicativos nativos para iOS e Android entre plataformas com Angular e TypeScript. Obtenha 100% de acesso a APIs nativas via JavaScript e reutilize pacotes do NPM, CocoaPods e Gradle. Código aberto e apoiado pela Telerik.



7- Conclusão:

Com tudo o que foi relacionado acima, podemos afirmar que os Frameworks Angular e Fondation são de imensa ajuda para qualquer desenvolvedor. Com destaque para o Angular que sem dúvida é um dos frameworks mais bem posicionados do mercado, tanto pela sua fácil adaptação com todos os dispositivos e browsers , mas também por ter uma rica biblioteca que configura um ambiente perfeito para os desenvolvedores ter a liberdade de criar , com diversas feramente e recursos ao seu alcance.

Frameworks são recursos que devem ser bem pensados e gerenciados, para poder aproveitar melhor possível do funcionamento do mesmo. Pois como foi ilustrado eles apesar de facilitar a vida dos desenvolvedores, se não usados corretamente podem despende tempo e fazer com que o código fique mais longo sem ter necessidade.

Tanto o Fondation quanto o Angular nos mostram tipos de frameworks , que funcionam de diferentes formas ,podendo ajudar mais o desenvolvedor em determinadas partes do código JS , ou no BackEnd , ou no FrontEnd com os códigos HTML e CSS por exemplo.

Nota-se também uma diferença de estruturação entre os frameworks apresentados , enquanto o Fondation não faz referencia a Single Page o Angular é especializado nesse tipo de página, e tem diversas features voltadas para a mesma , ainda assim também possui um imenso repertório com as demais possibilidades de sites e aplicações.

Enquanto o Fondation é mais voltado a desenvolvimento Web , principalmente ao HTML o Angular além de englobar esse aspecto também consegue ser eficaz com relação as demais perspectivas do mercado , como as aplicações Web e até Mobile, tendo uma abrangência maior.

A união desses Frame Works pode ser benéfica se bem usada e bem planejada. Entretanto deve-se pensar muito bem as decisões a se tomar e as features a implementar, para não deixar o sistema muito carregado, tendo consumos desnecessários.

Assim conclui-se que Ambos os Frameworks apresentados são extremamente úteis para o profissional , qualificado e que sabe como ,onde e quando deve-se usar cada recurso e cada Framework.



8- Referências Bibliográficas:

1. <https://blog.betrybe.com/framework-de-programacao/o-que-e-framework/>
2. <https://www.hostinger.com.br/tutoriais/frameworks>
3. <https://get.foundation>
4. <https://get.foundation/sites.html>
5. <https://blog.eduonix.com/web-programming-tutorials/top-15-websites-and-apps-built-with-angularjs/>
6. <https://angular.io>
7. <https://angular.io/features>
8. <https://angular.io/resources?category=development>



Trabalho Desenvolvimento Web

Docente: Prof Moacir Degasperi JR

Discente:

Rodrigo Luis Tavano Bosso – PC3005623

Frame Works Foundation & Angular



FIM

