

Uma Análise Comparativa entre os Frameworks Javascript Angular e React

**João Gabriel Colares de Camargos, José Flavio Coelho, João Paulo Aramuni, Humberto
Fernandes Villela**

Fundação Mineira de Educação e Cultura (FUMEC)
R. Cobre, 200 - Cruzeiro, Belo Horizonte - MG, 30310-190 - Brazil

joaogabriel.colares, joseflaviocoelho, joaopauloaramuni@gmail.com, humberto.vill
ela@fumec.br

Resumo.

O presente artigo realiza uma análise comparativa entre frameworks Javascript, React e Angular. O objetivo da escrita é apontar qual biblioteca é mais completa, mostrando dentre as opções qual agrega mais valor em um projeto. Para tanto, foram levantados aspectos de real importância no decorrer de um projeto para servirem de parâmetros na comparação, após analisar diversas publicações e estudos sobre o tema, chegamos a apenas 16 que atendiam aos critérios de inclusão e exclusão de artigos determinados pelos autores deste estudo. Os 16 artigos foram selecionados e classificados de acordo com sua contribuição. Constatou-se após análise que apesar de semelhantes, React é mais completo e proporciona dentro dos parâmetros utilizados um melhor resultado dentro de projetos web.

Abstract.

This article performs a comparative analysis between Javascript, React and Angular frameworks. What is written is to make the library more complete, indicating options that add more value to a project. In order to do so, aspects of real importance were raised during the course of a project to serve as parameters in the comparison, while at the same time addressing the various studies and thematic, they reach only 16 that met the criteria of exclusion and exclusion of articles. . The 16 titles were selected and classified according to their contribution. It was found after an analysis

that despite its characteristics, is more complete and easier to be used to improve the results of web projects.

1. Introdução

Este estudo trata da análise comparativa entre dois Frameworks JavaScript que vem se destacando na comunidade Web Brasileira, o Angular e o React.

Segundo Stackoverflow (2018), o maior site de perguntas e respostas exclusivos para desenvolvedores, o JavaScript hoje é a linguagem mais popular do mundo, com 69,8% da preferência entre os profissionais da área. Informação que é reforçada pela Agrela (2018).

Com as novas necessidades do mercado, as empresas estão cada vez mais se preocupando com suas presenças digitais, trocando assim seus sistemas locais por aplicações web e aplicativos mobile.

O desenvolvimento dessas aplicações exigem sites leves, que tenham uma boa experiência do usuário (UX) e páginas web interativas, tornando assim o JavaScript uma boa opção na hora de escolher em qual linguagem programar. Para Kleverton (2019) “UX é importante porque coloca a perspectiva do usuário como a espinha dorsal de qualquer fluxo de experiência ...”.

O termo UX “User Experience” foi popularizado por Don Norman enquanto trabalhava na Apple, desde então o foco em usabilidade passou ser cada vez mais frequente em projetos de desenvolvimento, como informado Macedo (2019).

A UX, experiência de usuário e a UI, interface do usuário, hoje em dia são primordiais quando se fala de desenvolvimento de sites, aplicações web e aplicativos mobile. Os usuários esperam ter facilidade, rapidez e boa experiência ao interagir com essas aplicações. Por outro lado estão os responsáveis pelas páginas que esperam uma boa manutenibilidade e um grande leque de opções proporcionados durante o desenvolvimento.

O crescimento na escolha pelo Javascript tem se mostrado contínua, além do mercado de trabalho o JavaScript vem sendo utilizado por universidades de todo mundo na iniciação de jovens profissionais, pela sua versatilidade e resiliência como demonstrado por Francisco (2017).

Outro fator que pode ser considerado como causa do crescimento da aceitação da linguagem é o grande número de frameworks existentes, a lista é extensa, como relatado por Javascripting (2019).

Os Framework são diversas funções prontas, ferramentas, que formam a estrutura base de um projeto, facilitando e agilizando o seu desenvolvimento. Ele é de extrema importância para aumentar a comunidade de uma linguagem e torná-la ampla.

Com tantas pessoas optando pela utilização do Javascript e com tantas opções de frameworks no mercado, é cada vez mais recorrente na hora de iniciar um projeto dúvidas como: Qual Framework é o melhor? Qual Framework utilizar? Quais fatores devo considerar para esta decisão? Qual possui a menor curva de aprendizado ?

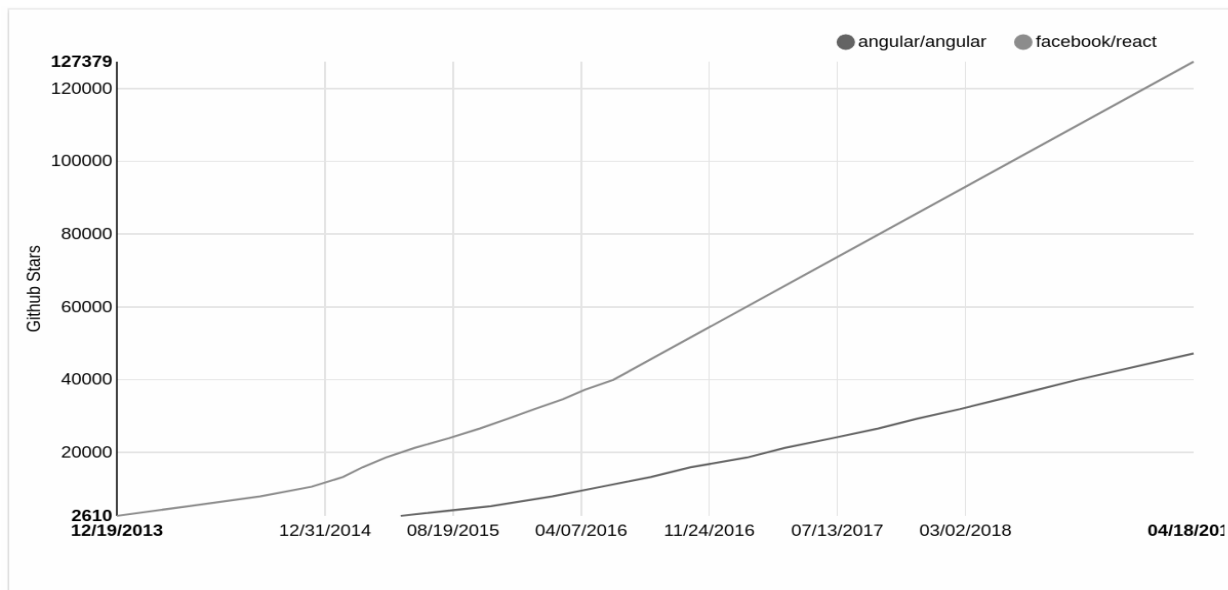
É importante entender que não é possível de forma generalista definir que um Framework é melhor do que outro, mas procurar entender mediante a cenários comuns encontrados no mercado, qual a tecnologia mais indicada para aquela ocasião.

Para escolher quais bibliotecas analisar algumas informações foram levadas em consideração, como quais são as favoritas dos profissionais da área, empresas que mantêm as bibliotecas, pioneirismo na área, estrelas no GitHub e fluxo de perguntas no site StackOverFlow.

O Angularjs é um framework com 58 mil estrelas no GitHub, 250 mil perguntas feitas no StackOverFlow, é atualmente mantida pelo Google e um dos mais utilizados pelos profissionais. Apesar de existirem duas versões diferentes, sua curva de aprendizado é pequena .

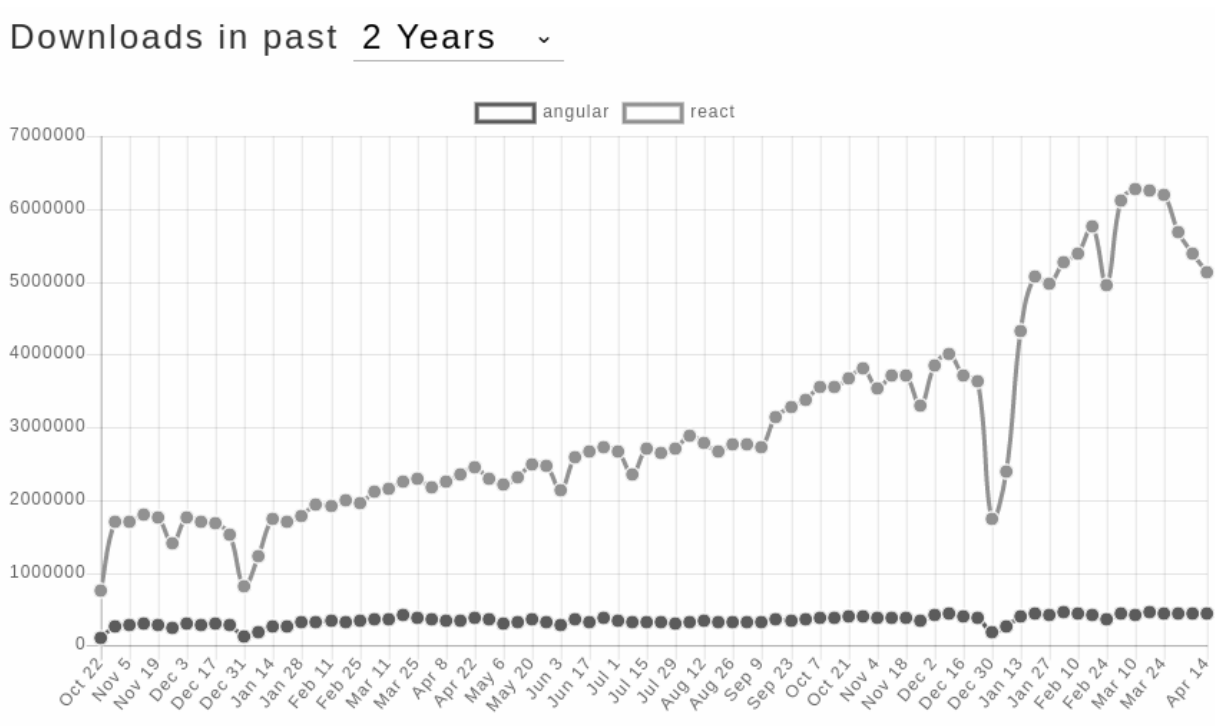
O React é uma biblioteca com 91 mil estrelas no GitHub, 78 mil perguntas feitas no StackOverFlow, foi desenvolvida pelo Facebook e tem uma grande comunidade que a defende, apesar de ter uma curva de aprendizado maior, levando em consideração o Angularjs.

Figura 1 - Histórico de Stars do Angular e React no Github



Fonte: STARHISTOR, 2019

Figura 2 : Histórico de downloads pelo npm packages



Fonte : NPMTRENDS, 2019

2. Metodologia

Nesta análise comparativa entre os frameworks Javascript, Angular e React, levantar pontos importantes que devem ser utilizados na hora de escolher em qual ferramenta desenvolver o front end de uma aplicação web, considerando também os pilares da qualidade de um software, conforme descrito em Pilares. (2018).

Alguns pontos foram considerados como essenciais na hora de comparar e escolher um framework para seu projeto front end. Esses pontos são curva de aprendizagem, Ambiente de desenvolvimento / Instalação, construção de templates, integração com a lógica, gerenciador de formulários, comunicação cliente servidor, recursos, facilidade no desenvolvimento, flexibilidade, tempo de carregamento e desempenho da aplicação final.

O processo de pesquisa foi conduzido seguindo o processo descrito por Peterson et al. (2008), que trabalha com cinco passos: (i) definição de questões de pesquisa, (ii) realização da pesquisa de estudos primários relevantes, (iii) triagem dos documentos, (iv) keywording dos resumos, e (v) a extração de dados e mapeamento.

3. Revisão de literatura

A comparação conduzida neste artigo consiste em uma conversa entre especialistas da área de desenvolvimento Web, autores, desenvolvedores e pesquisadores que realizaram algum estudo ou se posicionaram de alguma maneira sobre o assunto.

Tomando os pontos a serem analisados neste debate entre especialistas, foi identificado qual o framework apresentará o melhor resultado para cada cenário.

Começando então, analisando a Curva de Aprendizado, termo Curva de aprendizado, conforme descrito por Bastos (2015), foi citado pela primeira vez em 1936, por Theodore Wright que relacionou produtividade com a produção acumulada, verificando que existia um padrão de aprendizagem e que o tempo para produzir uma unidade adicional de produto diminuía.

Segundo Santos (2017), Angular é um framework bem complexo que possui alguns conhecimentos técnicos prévios sobre WebPack, módulos Node.js, npm e TypeScript. Diante disso, para o autor, a curva de aprendizado é somada pela complexidade da ferramenta com os conhecimentos prévios, portanto, possui curva de aprendizagem alta.

Para Pereira (2018) o conhecimento em ES6 ou ECMAScript 6 por parte do desenvolvedor é um passo importante que minimiza a curva de aprendizagem no ReactJS. O mesmo serve para a relação entre TypeScript e Angular, onde o conhecimento da linguagem, facilita a curva de aprendizado na ferramenta.

O TypeScript permite o código JavaScript seja desenvolvido com Orientação a Objeto. Como Daniel Schmitz (2015) define que “Se você gosta de tipar suas variáveis e métodos, criar classes, interfaces, usar Orientação a Objetos, o TypeScript foi feito para você e, claro, pode dizer adeus ao JavaScript.”. Com ele todas as classes que escrevemos são compiladas e transformadas em JavaScript puro.

Na visão de Frias (2019), as últimas atualizações do React trouxeram total integração com as versões mais modernas do ECMA Script, o que traz um desenvolvimento Web equivalente às linguagens mais comuns como Java, C++ e Python, tornando a curva de aprendizado mais simples. Sobre React, Santos (2017) cita como pré-requisitos o conhecimento em Node.js, módulos do npm, configuração do WebPack, conhecimento sobre JSX, tornado assim “a curva de aprendizado altíssima!”.

Diretamente ligada a Curva de aprendizagem tem-se o ponto de análise Facilidade no desenvolvimento, que se trata de quais recursos a ferramenta oferece para o desenvolvedor conseguir atingir seu objetivo com o menor custo possível, considerando linguagem utilizada até reaproveitamento de código possível.

Na visão de Santos (2018) Angular possui uma vantagem nesse quesito, e essa vantagem se resume ao TypeScript. Uma linguagem tipada que permite prevenir inúmeros erros de codificação, como digitação incorreta de variável ou parâmetro de função passado de forma errada. Já com React, ainda segundo autor é muito trabalhoso programar sem problemas, tendo como um ponto bom a maioria dos erros comuns que acontecem são mostrados no console com possíveis soluções.

Segundo Frias (2019) o React possui um poderoso algoritmo de manipulação do DOM, o que se torna um grande diferencial. Com o VDOM, o React, atualiza apenas os elementos necessário na tela, sem que tenha um recarregamento da tela toda, melhorando assim tempo de

carregamento e usabilidade dos usuários. Pereira (2018) define VDOM como “um conceito de programação onde existe uma representação da UI cacheada em memória, sincronizada com o DOM real do browser, com a utilização do ReactDOM”.

Ainda tratando de um critério relacionado ao começo da experiência Dev-Framework, foi analisado as ferramentas quanto a dificuldade em instalar e criar o Ambiente de desenvolvimento.

No ponto de vista de Santos (2017), preparar o ambiente com Angular é simples graças ao Angular command line interface (CLS), que com a instalação do @angular/cli diversos comandos para criar aplicações, executar testes e realizar o deploy são disponibilizados, sendo necessário apenas uma ide para codificar, opções que sobram no mercado como VSCode, SublimeText e Atom.

Diferente do React, que na opinião de Santos (2018), torna a tarefa de instalação e preparação do ambiente difícil e custosa, ele afirma que não existe um “Best Way” para iniciar o desenvolvimento, é necessário que o desenvolvedor saiba todos os componentes NPM que ele irá utilizar. Aspecto que Frias (2019) entende como uma vantagem do React, pois permite que você instale apenas o que realmente vai utilizar, deixando seu trabalho mais consciente permitindo a construção de aplicações mais leves.

A reutilização de código é um recurso muito utilizado hoje em dia, umas das maneiras de trabalhar desta maneira e com a Construção de Templates, um importante quesito que também estamos levando em consideração.

Tanto React, quanto Angular são baseadas em componentes, que são estruturas que recebem um entrada de dados, computa os dados e retorna uma interface renderizada. Os componentes, são estruturas que podem ser reutilizadas em outras páginas ou até mesmo dentro de outros componentes.

Então qual seria a diferença entre os dois na construção dos templates? House (2019) diz que “Angular coloca JS dentro do HTML. Já React colocar HTML dentro do JS”.

Sobre Templates, Santos (2017) acredita a facilidade é muito grande dentro do Angular, tendo em vista que estes templates são arquivos HTML comuns. Por outro lado, segundo o autor, com React esta construção se apresenta um pouco mais complexa, os templates são desenvolvidos

através de classes dentro de um método `Return()`, logo o template não é um HTML, mas sim um JSX, uma notação JavaScript parecida com HTML, mas que com algumas diferenças transforma uma tarefa simples como copiar e colar uma estrutura html em uma tarefa complexa.

Em direção oposta a Santos(2017), Neuhaus (2019) acredita que o React, nesse quesito, tem uma vantagem sobre o Angular, já que o Javascript é uma ferramenta mais poderosa que o HTML. Com o JSX, o React, junta os modelos de interface com a lógica do JavaScript, sendo assim uma grande vantagem na hora de desenvolvimento, tendo tudo em um lugar só, as verificações e tempo de compilação de código funcionam melhor.

O Angular, segundo Bruno (2019), “Combina declarative templates, dependency injection, end to end, tooling, entre outras ferramentas de formar a dar mais poder aos Developers para criar seus apps.”

Outro importante ponto a se levar em consideração na escolha de sua ferramenta é a flexibilidade, considerado um dos pilares da qualidade de um software (Escalabilidade), este critério consiste na capacidade de um software em trabalhar com grandes cargas.

React oferece uma maior flexibilidade para mudar uma aplicação grande, SPA, para uma arquitetura de microservices , podendo utilizar partes de um programa antigo. Já Angular é melhor para trabalhar com estruturas tipo de um SPA, pois é uma aplicação mais inchada para usar microservices.

Cory House (2016), observa que “ O JavaScript se modifica rapidamente e o React permite que você troque pequenos pedaços de sua aplicação por outras bibliotecas melhores, em vez de esperar e esperar que seu framework inove”.

Sendo assim sua flexibilidade em usar o React é maior, por conseguir utilizar em aplicações grandes ou pequenas e por conseguir injetar outras tecnologias melhores quando julgar necessário. Um ponto básico no desenvolvimento de qualquer aplicação Web é a Comunicação Cliente Servidor, falhas nesse ponto do projeto podem acarretar no fracasso de uma aplicação. Por isso levarmos em consideração este aspecto chave é muito importante.

Segundo Santos(2018) o Angular 4.3 consegue eximir de maneira bem eficiente o desenvolvedor de gerenciar as requisições e respostas, isso graças a um componente chamado HttpClient que concentra a lógica de comunicação com servidor via HTTP.

Pelo lado do React, existe o módulo Axios que é baseado em promisses, suas respostas são em formato JSON, formato que é utilizado em grande escala, facilitando assim as leituras das respostas e tornando dispensável parseamento de dados na manipulação de objetos. JSON é definido, em seu próprio site, como “uma formatação leve de troca de dados.”

Se compararmos o tamanho dos dois vemos que o Angular, por ser um framework fullstack, é mais pesado. Jean Neuhaus (2017) diz que o tamanho do Angular zipado é de 143k, já o React tem apenas 43k.

Analisando a performance, foi levado em consideração duas variáveis primordiais, tempo de carregamento e alocação de memória. O site de Stefan krause, faz essa comparação entre vários frameworks. O ambiente utilizado para os teste é um MacBook Pro 15 (2,4 GHz i7, 16GB RAM, OSX 10.12.5, Chrome 58.0.3029.110 (64bits)) e as versões dos frameworks são angular-v4.1.2-keyed e react-v15.5.4-redux-v3.6.0 .

Figura 3 : Tabela comparativa de tempo de carregamento

<u>Name</u>	angular- v4.1.2- keyed	react- v15.5.4- redux- v3.6.0
create rows Duration for creating 1000 rows after the page loaded.	193.1 ± 7.9 (1.0)	212.2 ± 14.2 (1.1)
replace all rows Duration for updating all 1000 rows of the table (with 5 warmup iterations).	197.4 ± 5.3 (1.0)	206.7 ± 7.3 (1.0)
partial update Time to update the text of every 10th row (with 5 warmup iterations).	13.0 ± 4.5 (1.0)	18.0 ± 1.6 (1.1)
select row Duration to highlight a row in response to a click on the row. (with 5 warmup iterations).	3.4 ± 2.3 (1.0)	8.7 ± 2.9 (1.0)
swap rows Time to swap 2 rows on a 1K table. (with 5 warmup iterations).	13.4 ± 1.0 (1.0)	17.1 ± 1.3 (1.1)
remove row Duration to remove a row. (with 5 warmup iterations).	46.1 ± 3.2 (1.0)	52.4 ± 1.7 (1.1)
create many rows Duration to create 10,000 rows	1946.0 ± 41.8 (1.0)	1931.7 ± 35.6 (1.0)

<u>Name</u>	angular- v4.1.2- keyed	react- v15.5.4- redux- v3.6.0
ready memory Memory usage after page load.	4.8 ± 0.0 (1.0)	4.9 ± 0.1 (1.0)
run memory Memory usage after adding 1000 rows.	10.9 ± 0.1 (1.0)	10.8 ± 0.1 (1.0)

Fonte: STEFANKRAUSE, 2019

O resultado dessa análise são as duas imagens anteriores, pode ser analisado que apesar da diferença o tempo de processamento, analisado em milissegundo, entre os dois serem praticamente imperceptível aos nossos olhos, o Angular tem o processamento melhor. Porém quando analisamos a alocação de memória em MB vemos que os dois são iguais.

4. Conclusão

Atualmente, existem vários frameworks disponíveis no mercado, fazendo com que tenha várias possibilidades de escolha de qual seria a mais viável para o desenvolvimento de um sistema. Principalmente, quando o tema é desenvolvimento de Front End de aplicações web. Dessa forma, na análise comparativa, foi levantado as maiores qualidades e diferenças dos dois maiores frameworks da atualidade, o Angular e ReactJs.

De uma forma geral, ambos são eficientes, mas, considerando parâmetros comuns do desenvolvimento de projetos de software, que são o prazo de entrega, curva de aprendizado e flexibilidade da ferramenta, pode-se constatar que o Reactjs seria a melhor opção.

Dessa forma, há uma tendência em se utilizar o Reactjs, por apresentar uma curva de aprendizado menor quando se comparado com os demais frameworks. Além do que, apresenta uma grande comunidade global o apoiando e sua massa de materiais didáticos, do nível básico ao avanço disponíveis na internet é enorme. Sua capacidade de reaproveitamento de código, utilizando seus componentes compilados em JavaScript, aumenta a performance na hora de desenvolver, ocasionando uma redução no tempo de entrega dos projetos.

Portanto, o Reactjs apresenta qualidades promissoras para sua utilização, se tornando uma ferramenta ampla para qualquer tipo de necessidade já que é flexível para integrar a outras bibliotecas e funcionalidades disponíveis no mercado. Além disso é compatível com sua extensão para desenvolvimento de aplicações mobile, o React Native.

O mercado requer constantemente progressos, nos exigindo demandas e necessidades diferentes, fazendo com que tenha espaço para vários frameworks, e tornando o número destes ainda maior, porém, como foi abordado, os que se destacam são os que têm maiores comunidades.

5. Referências

AGRELA, Lucas. Estas são as 20 linguagens de programação mais usadas. 2018. Disponível em: <<https://exame.abril.com.br/tecnologia/estas-sao-as-20-linguagens-de-programacao-mais-usadas/>>. Acesso em: 07 mar. 2019.

BASTOS, Ercília. O que é a curva de aprendizagem? 2015. Disponível em: <<https://www.portal-gestao.com/artigos/7692-o-que-%C3%A9-a-curva-de-aprendizagem.html>>. Acesso em: 16 abr. 2019.

BRUNO, Pedro. A batalha épica: Angular vs React - quem ganhará? Disponível em: <<https://www.eddisrupt.com/blog/aprender-a-programar/angular-vs-react>>. Acesso em: 21 abr. 2019.

FRANCISCO, Thomas Claburn In San. Stanford Uni's intro to CompSci course adopts JavaScript, bins Java: Java's days are numbered – but it's a very large number. 2017. Disponível em: <https://www.theregister.co.uk/2017/04/24/stanford_tests_javascript_in_place_of_java/>. Acesso em: 13 mar. 2019.

FRIAS, Thiago. React vs Vue vs Angular: qual escolher. 2019. Disponível em: <<http://blog.geekhunter.com.br/react-vs-vue-vs-angular-qual-escolher/>>. Acesso em: 23 abr. 2019.

HOUSE, Cory. Angular 2 versus React: There Will Be Blood. Disponível em: <<https://medium.freecodecamp.org/angular-2-versus-react-there-will-be-blood-66595faafd51>>. Acesso em: 18 abr. 2019.

JAVASCRIPTING. The definitive source of the best JavaScript libraries, frameworks, and plugins. Disponível em: <<https://www.javascripting.com/>>. Acesso em: 13 mar. 2019.

K., Kleverton. O que é UX – User Experience. 2019. Disponível em: <<https://www.hostinger.com.br/tutoriais/ux-o-que-e-user-experience/>>. Acesso em: 22 jan. 2019.

KRAUSE, Stefan. Results for js web frameworks benchmark – round 6. Disponível em: <<https://www.stefankrause.net/js-frameworks-benchmark6/webdriver-ts-results/table.html>>. Acesso em: 24 abr. 2019.

MACEDO, Paula. Origens de User Experience. Disponível em: <<https://brasil.uxdesign.cc/origens-de-user-experience-9d23f67f7c77>>. Acesso em: 25 mar. 2019.

NEUHAUS, Jens. Angular vs. React vs. Vue: A 2017 comparison. Disponível em: <<https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176>>. Acesso em: 20 abr. 2019.

NPMTRENDS. Angular vs react. 2019. Disponível em: <<https://www.npmtrends.com/angular-vs-react>>. Acesso em: 14 abr. 2019.

PEREIRA, Manacés. Angular 6 vs ReactJS—A web sempre ganhando. 2018. Disponível em: <<https://medium.com/manacespereira/angular-6-vs-reactjs-a-web-sempre-ganhando-93f89fbc3735>>. Acesso em: 23 abr. 2019.

PILARES da qualidade de software. 2018. Disponível em: <<https://docs.microsoft.com/pt-br/azure/architecture/guide/pillars>>. Acesso em: 16 abr. 2019.

REIS, Vinicius. Entenda de uma vez por todas o que é ReactJS, Angular 2, Aurelia e VueJS. 2017. Disponível em: <<https://medium.com/by-vinicius-reis/o-que-e-react-ng2-aurelia-vue-e34b0c77b5a1>>. Acesso em: 18 abr. 2019.

SANTOS, Daniel. Tabela de comparação entre Angular, React + Redux e Vue.js. 2017. Disponível em: <<https://medium.com/@daniel.dia/compara%C3%A7%C3%A3o-entre-angular-react-redux-e-vue-js-a256d0fce8e0>>. Acesso em: 10 abr. 2019.

SCHMITZ, Daniel. Diga olá ao TypeScript e adeus ao JavaScript. 2015. Disponível em: <<https://tableless.com.br/diga-ola-ao-typescript-e-adeus-ao-javascript/>>. Acesso em: 14 dez. 2015.

STACKOVERFLOW. Most-popular-technologies. 2018. Disponível em: <<https://insights.stackoverflow.com/survey/2018/#most-popular-technologies>>. Acesso em: 10 mar. 2019.

STARHISTORY. **Star history**. 2019. Disponível em: <<https://star-history.t9t.io/#angular/angular&facebook/react>>. Acesso em: 20 abr. 2019.

STEFANKRAUSE. JS Frameworks Benchmark. 2019. Disponível em: <<https://www.stefankrause.net/js-frameworks-benchmark6/webdriver-ts-results/table.html>>. Acesso em: 20 abr. 2019.