

ANÁLISE COMPARATIVA ENTRE AS TECNOLOGIAS DE FRONT-END REACT, ANGULAR E VUE

COMPARATIVE ANALYSIS BETWEEN THE FRONT-END TECHNOLOGIES REACT, ANGULAR AND VUE

Sandy Hoffmann

Graduanda em Ciência da Computação no Instituto Federal Catarinense Campus Blumenau
manglesan9228@gmail.com

Luana Alves Pinto

Graduanda em Ciência da Computação no Instituto Federal Catarinense Campus Blumenau
luanaalvesp2811@gmail.com

Luiz Ricardo Uriarte

Doutor em Engenharia de Produção pela Universidade Federal de Santa Catarina
luiz.uriarte@ifc.edu.br

Resumo:

Diante da popularidade do JavaScript e do crescimento dos *frameworks* e bibliotecas voltadas ao desenvolvimento *front-end*, a presente pesquisa visa fazer a comparação entre React, Angular e Vue, avaliando critérios de popularidade e de performance. Esse estudo tem o propósito de mostrar ao leitor as características de cada tecnologia, mostrando seus pontos positivos e negativos. Após um breve resumo para conhecer as principais características das tecnologias é feito a avaliação de performance, na qual o React obteve destaque em downloads no NPM com mais de 705 mil downloads, ficando em primeiro lugar, seguida por Vue e Angular. Na pesquisa de formulário proposta nesse artigo, o React também ganhou destaque em questões que tratam de reconhecimento e uso. Nos testes de desempenho, o Vue conseguiu o melhor tempo independente do número de elementos inseridos. Assim conclui-se que diante dos testes baseados nos critérios limitados por este artigo, o React é considerado o mais popular e o Vue com o melhor desempenho.

Palavras-Chave: *React. VueJS; Angular; Framework; Biblioteca.*

Abstract:

Facing Javascript's popularity and the growth of libraries and frameworks focused on front-end development, this research has the objective of comparing React, Angular and Vue, grading their popularity and performance. The research has the purpose of showing the reader the features of each technology presented, showing their pros and cons. After a short summary in order to know the main features of each technology it's done an evaluation of their performance, in which React got first place, highlighted for its 705 thousand downloads from NPM, followed

by Vue and Angular respectively. In the research done by a form proposed in this article, React also stood out when it came to questions related to its recognition and use. During the performance tests, Vue was the one with the best time, regardless of the amount of inserted elements. With this, it's concluded that with the tests made with the criteria previously limited by this article, React is considered the most popular meanwhile Vue has the best performance.

Keywords: React; VueJS; Angular; Framework; Library.

1. INTRODUÇÃO

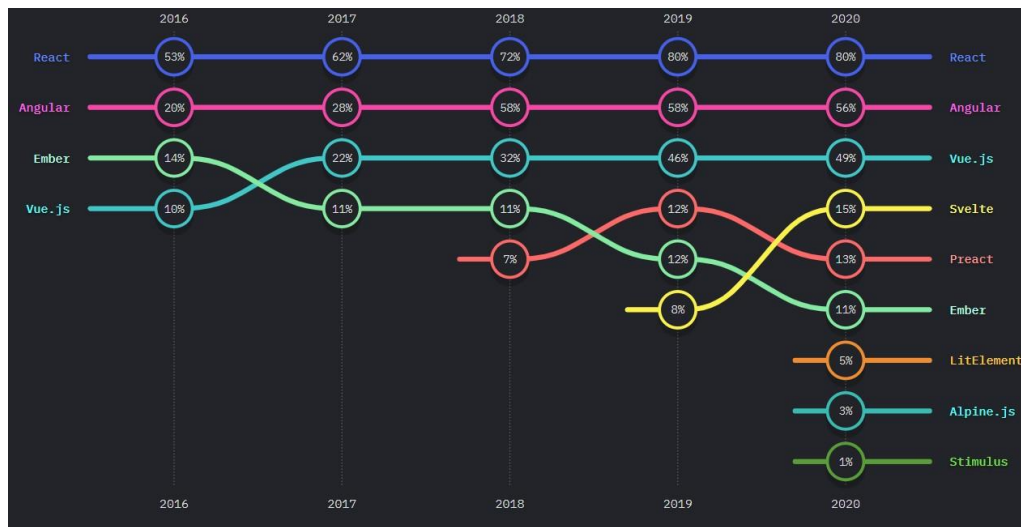
O JavaScript é uma linguagem de programação muito popular entre os programadores de todo o mundo, ocupando o primeiro lugar na pesquisa “Size of Programming Language Communities in Q3 2021” feita pelo SlashData (2021), com aproximadamente 16,4 milhões de desenvolvedores.

Deste modo, com a comunidade ativa, há também uma quantidade significativa de pacotes feitos para o uso das aplicações, a fim de facilitar processos comuns em determinados tipos de programas. De acordo com o NPM (Node Package Manager - Gerenciador de Pacotes Node), em 2019 foram lançados mais de 1 milhão de pacotes elaborados pela comunidade. Esses pacotes podem ser classificados em *frameworks* ou bibliotecas, dependendo da sua complexidade.

Segundo Sandip Roy (2022), um *framework* serve como uma fundação para construir um *software*, provendo funcionalidades necessárias para se construir toda uma aplicação, conseguindo trabalhar em conjunto com bibliotecas. Já uma biblioteca pode ser definida como um conjunto de métodos que servem para cumprir determinado propósito, facilitando a implementação de alguma funcionalidade específica.

Dentre os *frameworks* e bibliotecas existentes, tendo em vista a pesquisa “Front End Frameworks” da State of JS (2020) em relação ao uso pelos desenvolvedores, o top 3 fica com React, Angular e Vue, como pode ser visto na Figura 1.

Figura 1 - Ilustração de rankings em relação ao uso de *frameworks* nos projetos - “Front End Frameworks”



Fonte: State of JS (2020).

Diante dessa análise, a presente pesquisa irá fazer uma comparação entre às três tecnologias, utilizando critérios de avaliação como: popularidade e performance.

Desta forma, o objetivo do presente trabalho é mostrar ao leitor características dos três pacotes, permitindo uma análise sobre qual é a biblioteca/*framework* que atende mais pontos positivos para o objetivo de uso desejado pelo programador.

2. CRITÉRIOS DE AVALIAÇÃO

Para produzir o presente trabalho, foi pensado alguns critérios de avaliação, a fim de fazer uma comparação entre as ferramentas estudadas neste artigo. Os critérios escolhidos foram popularidade e performance, em seguida será explicado sobre cada um.

2.1 POPULARIDADE

Na pesquisa feita pelos autores (MOSTRÖM & RYRBERG, 2022), foram realizadas algumas perguntas em uma entrevista. Uma das perguntas sucedeu a quais fatores técnicos e

não técnicos afetam a escolha de *frameworks* web, muitos responderam sobre a popularidade dos mesmos.

Diante desse fato, será analisada a popularidade dos *frameworks* utilizando de três pontos de vista diferentes a fim de obter um resultado mais conciso e abrangente. A primeira análise será feita em cima do índice de downloads semanais e anuais retirado das suas respectivas páginas no NPM (Node Package Manager - Gerenciador de Pacotes Node). O NPM é o gerenciador de pacotes, nativo do Node JS, contando com mais de um milhão de pacotes JS (NPM, 2022). A quantidade de downloads do começo do ano até então (12/11/2022), também será considerada. Esses dados serão obtidos utilizando o npm-stat, que utiliza os dados do NPM e faz estatísticas que visam mostrar o índice de downloads dos pacotes em determinados períodos.

O segundo indicador será a análise das *webpages* do GitHub das respectivas tecnologias. O GitHub é uma plataforma onde é possível guardar repositórios, trabalhar coletivamente, visitar repositórios públicos e dar estrelas, entre outras coisas. Atualmente há mais de 83 milhões de usuários, com cerca de 200 milhões de projetos na plataforma (GitHub, 2022). No GitHub um repositório pode ser público, permitindo a qualquer usuário ter acesso ao código e especificações daquela página, ou privado, que permite a apenas uma parte de usuários selecionados conseguir ver e acessar os dados relacionados àquele projeto. Todas as três tecnologias abordadas neste artigo possuem suas *webpages* no GitHub, de modo público.

Tendo esse fato em vista, será avaliado o número de estrelas de cada uma, e o número de repositórios públicos que utilizam dessa tecnologia no GitHub. Na documentação do GitHub (2022) consta que as estrelas servem para salvar os repositórios que o usuário achar interessante e descobrir outros repositórios relacionados com o que ele avaliou, também dando um *feedback* positivo ao conteúdo daquele repositório específico. Para ver o histórico de estrelas do repositório selecionado será utilizado o star-history, que mostra a trajetória de crescimento da quantidade de avaliações conforme os anos de GitHub. O GitHub também consegue detectar quantos repositórios públicos possuem dependência daquele repositório específico que estamos analisando, permitindo ver o quanto o uso da tecnologia está influenciando os projetos da comunidade.

A terceira forma abordada é um formulário distribuído de maneira online pelo LinkedIn que possui algumas respostas de maneira direta ao usuário sobre as tecnologias abordadas pela pesquisa. O LinkedIn é uma rede social voltada ao mercado profissional, possuindo mais de 850 milhões de usuários (LinkedIn, 2022). Por se tratar de uma rede voltada a profissionais na área de trabalho, a pesquisa foi lançada em forma de postagem, utilizando o Google Forms, de modo a obter os dados respondidos e levantar uma análise.

Algumas perguntas feitas pelo formulário foram: você trabalha no ramo tecnológico, você cursa/cursou faculdade na área da computação, qual das tecnologias comentadas (React, Vue e Angular) você já ouviu falar, quais dessas tecnologias você já utilizou e qual dessas tecnologias é utilizada na empresa em que você trabalha. Em tópicos seguintes, será mostrado mais detalhadamente o formulário e explicado os dados coletados.

2.2 PERFORMANCE

A baixa performance de uma tecnologia está entre os principais fatores técnicos que podem pesar para que profissionais deixem de usar um *framework* (MOSTRÖM & RYRBERG, 2022). Isso se dá pelo fato da performance interferir diretamente na relação do sistema com o usuário, e ter o risco de aumentar conforme a aplicação aumenta os fatores de tamanho e complexidade.

Deste modo, será posto à prova as três tecnologias na questão da sua performance em renderização. O processo de renderização possui dois principais tipos, o CSR (Client Side Rendering - Renderização no lado do cliente) e SSR (Server Side Rendering - Renderização no lado do servidor). Na renderização SSR, o servidor é responsável por responder com documentos HTML as requisições feitas pelo cliente, processando os dados e enviando de modo que o navegador apenas precise exibir. Em contrapartida, no caso da CSR, o responsável pela lógica da página é o cliente, ficando para ele resolver o HTML com marcações, processando o próprio para a exibição (SCHNEIDER, 2016).

Por padrão, segundo a documentação das mesmas, as tecnologias Vue, Angular e React possuem a dinâmica CSR (apesar de ser possível configurar para a alteração do modelo de

renderização), logo a avaliação de performance será voltada para essa renderização em específico.

Para esses testes, será utilizado um mesmo modelo de base, um gerenciador de contatos, como pode ser visto na Figura 2. Esse gerenciador terá como foco a função de adicionar contatos, assim tabelando os tempos que essa tarefa leva para: 10^1 , 10^2 , 10^3 , 10^4 , e 10^5 de linhas de dados. A cada teste, a página é atualizada.

Figura 2 - Lista de contatos.



Fonte: Elaborado pelos pesquisadores, (2022).

O monitoramento desse tempo irá ser obtido pelas extensões fornecidas pelas próprias tecnologias. O navegador utilizado para os testes será o Google Chrome na sua versão mais recente (Versão 107.0.5304.107) 64 bits. No Chrome Web Store (gerenciador de extensões para o Google Chrome) iremos utilizar as extensões fornecidas para monitorar a performance dos componentes. No caso do React, a extensão fornecida será a React Dev Tools (versão 4.25.0), fornecida pelo Meta, que conta com mais de 3 milhões de usuários (Chrome Web Store, 2022). Para o Vue será utilizado a extensão Vue.js devtools (versão 6.4.5), fornecida pelo vuejs.org, contando com mais de 1 milhão de usuários (Chrome Web Store, 2022). Por último,

para o Angular será utilizado a extensão Angular DevTools (versão 1.0.6), fornecida pelo Angular, contando com mais de 200 mil usuários (Chrome Web Store, 2022).

3. REACT

No seu site oficial, React é definido como uma biblioteca JavaScript para construir interfaces para o usuário. Segundo Hunt (2016), que era gerente de engenharia no Instagram quando o React começou a fomentar, um dos principais problemas que a equipe enfrentava era manter a interface do usuário sincronizada com os eventos que aconteciam nas páginas. Inicialmente só era possível fazer isso utilizando recursos do DOM, como *event listeners*. Porém, essa prática era bastante complicada, por isso foi necessário se planejar uma tecnologia que pudesse suprir essa dificuldade.

É comum ouvirmos que o React é um *framework*, pois ele é amplo e baseado em componentes, facilitando assim a sua manutenção. Com tudo, a ferramenta JavaScript React, é uma biblioteca para o desenvolvimento de aplicações *front-end*, sendo assim, usada para criar interfaces de usuários (UI) para grandes e pequenos projetos. (UZAYR et al., 2019)

Na documentação do site do React (2022), a biblioteca pode ser definida em três características principais: “Declarativo”, “Baseado em Componentes” e “Aprenda uma vez, use em qualquer lugar”.

Por se basear no paradigma declarativo, o React consegue permitir ao usuário dizer o que quer que aconteça, e não necessariamente citando o passo a passo. Isso permite ao código ficar mais fácil de entender, e mais limpo. Esse conceito é contrário ao paradigma imperativo, em que há a necessidade de citar o passo a passo do código.

O segundo pilar do React é ele ser baseado em componentes, logo é possível reutilizar elementos passando parâmetros de diferentes tipos de dados, tendo funções de maneira encapsulada, e cuidando dos estados de mudança de forma individual. Isso aproveita o conceito de reusabilidade de código, permitindo que não precise haver trechos de códigos repetidos no programa.

O terceiro pilar do React é o lema “Aprenda uma vez, use em qualquer lugar”, isso se dá pelo fato do React trabalhar com outras tecnologias, não precisando descartar o seu código já existente.

3.1 SINTAXE

O React utiliza da sintaxe JSX (JavaScript XML), nomeado como uma extensão de sintaxe para o JavaScript. Com o uso do JSX é criado os elementos do React, juntando JavaScript e HTML. Portanto, é comum ver expressões como a Figura 3, onde uma tag HTML pode ser posta em uma constante de JS. Por ser uma expressão, há a possibilidade de colocar JSX dentro de *ifs*, *loops* e quaisquer recursos da linguagem JS e retorná-los por funções.

Figura 3 - Exemplo de uma expressão.

```
const contato = <div>Contato 01</div>
```

Fonte: Elaborado pelos pesquisadores, (2022).

Um exemplo do uso da sintaxe do React JSX pode ser visto na Figura 4, ilustrando o conceito do JSX de misturar elementos de HTML com expressões de JS. No *return* da função *Contatos*, é feito uma verificação para confirmar que a propriedade *Contatos* veio para a função e se ela estiver presente é feito uma iteração de seus elementos, retornando seus dados dentro de *divs*, em tempo de execução.

Figura 4 - Exemplo da sintaxe do React.

```
export function Contatos(props){
  return(
    <>
    {
      props.contatos && props.contatos.map((contato) => {
        return (
          <div key={contato.id} className="contato">
            <div>
              <div className='imgContato' />
            </div>
            <h3>{contato.nome}</h3>
            <p>telefone: {contato.telefone}</p>
            <p>email: {contato.email}</p>
          </div>
        )
      })
    }
  </>
)
```

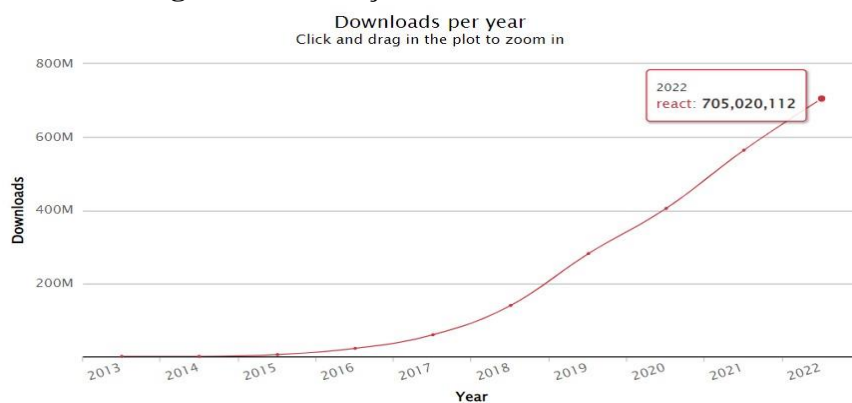
Fonte: Elaborado pelos pesquisadores, (2022).

3.2 AVALIAÇÃO

3.2.1 Popularidade

Na sua página oficial no NPM, do período de 06/11/2022 a 12/11/2022 houve cerca de 18.850.523 downloads. Utilizando do site npm-stat (2022), o pacote React teve em 2022 (até o dia 12/11/2022) mais de 705 milhões de downloads, já ultrapassando sua quantidade em 2021 que foi cerca de 564 milhões, como pode ser visto na Figura 5.

Figura 5 - Ilustração da taxa de downloads.



Fonte: npm-stat (2022)

Já o repositório do React no GitHub, conta com 198 mil estrelas e está sendo usado em 12.236.988 repositórios públicos (visto em 12/11/2022, GitHub).

3.2.2 PERFORMANCE

Para medir a performance do React foi utilizado o React Dev Tools, na funcionalidade *Profiler*. O resultado da avaliação de desempenho perante a adição de dados está exposta na Tabela 1, onde é feito uma média aritmética simples das primeiras três tentativas dos testes.

Tabela 1 - Tentativas de desempenhos.

| Número Elementos | 1° Tentativa | 2° Tentativa | 3° Tentativa | Média Final |
|------------------|--------------|--------------|--------------|-------------|
| 10 ¹ | 2.8ms | 2.8ms | 2.9ms | 2.83ms |
| 10 ² | 17.8ms | 18.3ms | 16.6ms | 17.57ms |
| 10 ³ | 103.2ms | 109.4ms | 98.9ms | 103.8ms |
| 10 ⁴ | 737.7ms | 753.9ms | 775.2ms | 755.6ms |
| 10 ⁵ | 7146.4ms | 7292.9ms | 7115.8ms | 7185.03ms |

4.VUE.JS

VueJS é um *framework* JavaScript voltado para aplicações web. Sua criação se deu em 2014 por Evan You , em busca de um *framework* que permitisse mais interatividade para os seus projetos.

Segundo a página de documentação do VueJS, o *framework* possui uma renderização declarativa, ou seja, utiliza o HTML permitindo utilizar sintaxes de *template* dentro das *tags*. Isso significa que o que se encontra dentro das *tags* pode ser controlado pela mudança de estados da linguagem JS.

Vue também é progressivo, permitindo que aplicações que comecem em uma escala menor consigam ir crescendo conforme o ritmo da aplicação necessitar. Isso faz com que seu uso seja ampliado de pequenas aplicações a grandes *softwares*. Também possuindo reatividade, Vue permite ao navegador detectar mudanças no código enquanto o programador está fazendo alterações em tempo real.

Os componentes desse *framework* funcionam em apenas um arquivo, com a extensão `.vue`, encapsulando a lógica em JS, a estilização e o HTML em seu conteúdo, porém caso o usuário prefira há também a possibilidade de fazer os componentes em JS puramente. Algumas das vantagens de usar a arquitetura que engloba tudo em um arquivo citadas na documentação são: *templates* pré-compilados e otimização de tempo de compilação, influenciando na melhora da performance.

4.1 SINTAXE

O Vue utiliza a sintaxe de *templates* baseados em HTML, permitindo ao programador incorporar seus dados ao componente alvo. É efetuado uma compilação dos modelos de *template* usando JS para assim permitir a exibição final ao usuário.

A inserção de novos dados ao *template* do Vue é feito utilizando chaves duplas, e nesse *framework* temos o uso de “diretivas”. As diretivas no Vue são funções que realizam determinadas ações pré-programadas pelo Vue, começando com “-v”, e afetando o elemento DOM. As diretivas também são programáveis a cargo do usuário, permitindo a sua personalização.

Um exemplo de código usando a sintaxe de *templates* pode ser visto na Figura 6, onde se define o código utilizado para mostrar um contato em `Vue.js`.

Figura 6 - Exemplo da sintaxe do Vue.

```
<script setup>
defineProps({
  contatos: {
    type: Array,
    default: () => [],
    required: true,
  },
});
</script>

<template>
  <div v-for="contato in contatos" :key="contato.id" class="contato">
    <div>
      <div className="imgContato"></div>
    </div>
    <h3>{{ contato.nome }}</h3>
    <p>telefone:{{ contato.telefone }}</p>
    <p>email: {{ contato.email }}</p>
  </div>
</template>
```

Fonte: Elaborado pelos pesquisadores, (2022).

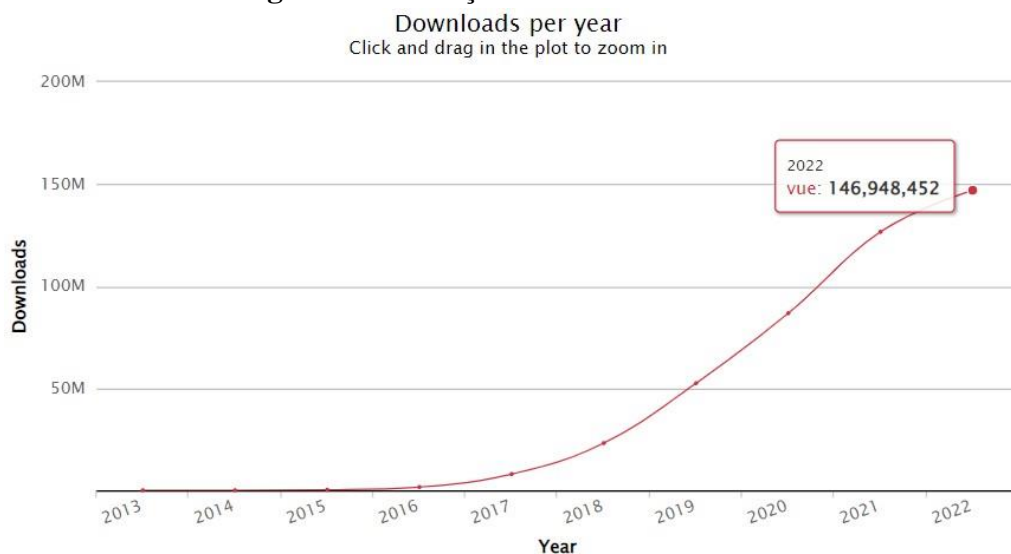
Há também a possibilidade de programar usando a função `render`, que utiliza mais recursos JS. Para isso se usa o conceito de *vnode*, que utiliza de *hyperscript* para criar os HTML desejados. Segundo o site de documentação do *hypertext* essa tecnologia serve para responder a eventos e manipular os elementos DOM (Document Object Model).

4.2 AVALIAÇÃO

4.2.1 Popularidade

Na página oficial do Vue no NPM, no período de 06/11/2022 a 12/11/2022 houve um total de aproximadamente 3.939.509 downloads. Utilizando o site *npm-stat* (2022), o *framework* teve em 2022 (até o dia 12/11/2022) cerca de 147 milhões de downloads, já ultrapassando sua quantidade em 2021, que foi cerca de 126 milhões. A Figura 7 ilustra esse crescimento ao decorrer dos anos.

Figura 7 - Ilustração da taxa de downloads.



Fonte: npm-stat (2022)

O repositório do Vue no GitHub conta com cerca de 201 mil estrelas, porém não foi possível encontrar a quantidade de pessoas que possuem repositórios públicos dependentes dessa biblioteca na sua página (visto em 12/11/2022, GitHub).

4.2.2 PERFORMANCE

A performance do Vue foi medida utilizando o Vue.js devtools, utilizando da sua medição de performance. Para a funcionalidade de adição de dados, o resultado pode ser observado na Tabela 2, onde são feitas 3 tentativas para os respectivos números de elementos e após isso é tirado uma média aritmética simples para os resultados.

Tabela 2 - Tentativas de desempenhos.

| Número Elementos | 1° Tentativa | 2° Tentativa | 3° Tentativa | Média Final |
|------------------|--------------|--------------|--------------|-------------|
| 10 ¹ | 1.8ms | 1.7ms | 1.6ms | 1.70ms |
| 10 ² | 5.5ms | 5.6ms | 5.8ms | 5.63ms |

| | | | | |
|-----------------|----------|----------|----------|----------|
| 10 ³ | 49.4ms | 56.8ms | 48.4ms | 51.53ms |
| 10 ⁴ | 349.9ms | 440.4ms | 404.8ms | 398.37ms |
| 10 ⁵ | 3357.7ms | 3201.8ms | 3213.0ms | 3257.3ms |

5. ANGULAR

Angular foi desenvolvido por Misko Hevery e Adam Abrons no ano de 2009, e é um framework voltado para o desenvolvimento de plataformas web (Angular, 2022). Seu objetivo é o desenvolvimento de aplicações com menos código, sendo mais objetiva, reusável e expressiva (Branas, 2014).

O *framework*, segundo seu site, pode ser utilizado para desenvolver aplicações web e native, de modo a contemplar de smartphones a computadores desktops. Outro ponto é a escalabilidade de dados, assegurando a possibilidade de crescer suas aplicações.

A divisão de um componente no Angular, é dividida em 3 arquivos principais, um HTML para servir como *template* da aplicação, um TypeScript, um arquivo de CSS que possui a estilização do componente, e um arquivo de especificação de teste. Por padrão, essa organização interna de componente pode ser criada utilizando o comando “ng generate component NomeComponente”, criando os arquivos padrões como ilustra a Figura 8.

Figura 8 - Padrões de arquivos.

```

✓ contatos
  # contatos.component.css
  <> contatos.component.html
  TS contatos.component.spec.ts
  TS contatos.component.ts
  
```

Fonte: Elaborado pelos pesquisadores, (2022).

5.1 SINTAXE

O *framework* Angular utiliza a sintaxe de *template*, utilizando a marcação HTML. A maioria das *tags* HTML são válidas no Angular, e ele soma com algumas funcionalidades próprias do *framework* para ajudar a fazer a manutenção dos elementos DOM. Os dados são colocados nos *templates* com o uso de chaves duplas, e o Angular possui diretivas semelhantes ao Vue, para adicionar comportamentos aos seus componentes, como a iteração ilustrada na Figura 9, onde a diretiva `*ngFor` itera os elementos da lista de contatos.

Figura 9 - Exemplo da sintaxe do Angular.

```
<div class="Conteudo">
  <div *ngFor="let contato of contatos" id={{contato.id}} class="contato" >
    <div>
      <div class='imgContato' ></div>

      </div>
      <h3>{{contato.nome}}</h3>
      <p>telefone: {{contato.telefone}}</p>
      <p>email: {{contato.email}}</p>
    </div>
  </div>
```

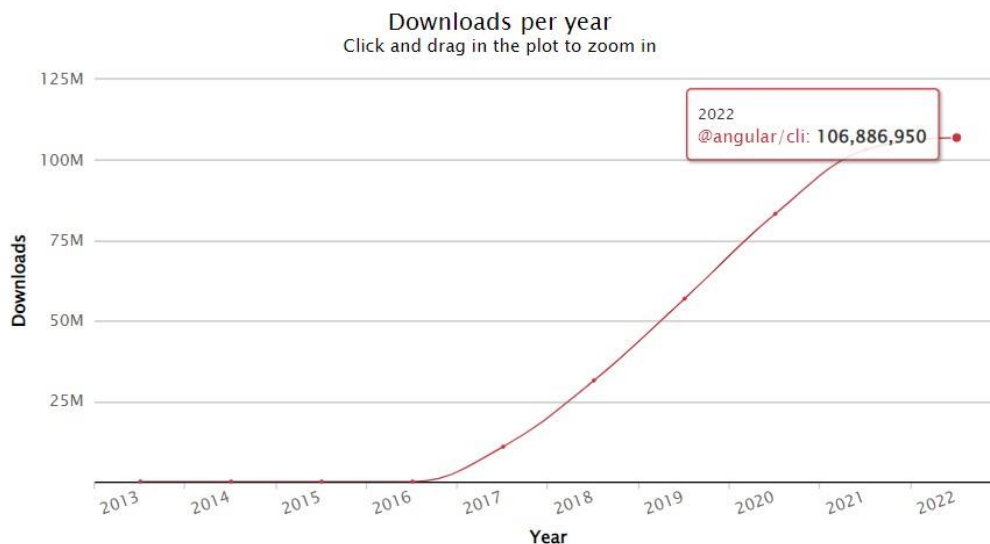
Fonte: Elaborado pelos pesquisadores, (2022).

5.2 AVALIAÇÃO

5.2.1 Popularidade

O *framework* angular possui funcionalidades divididas no site do NPM, porém esta pesquisa terá como foco os downloads do pacote Angular CLI, sendo o utilizado como base para a criação dos projetos e aplicações. Em sua página oficial @angular/cli no site do NPM, no período de 06/11/2022 a 12/11/2022, houve um total de 2.555.422 downloads. Usando o site npm-stat (2022), o *framework* teve do início do ano de 2022 até o dia 12/11/2022 cerca de 106 milhões de downloads, ultrapassando a quantidade do ano de 2021, que foi cerca de 103 milhões. A Figura 10 ilustra o crescimento dos downloads desta parte da biblioteca do Angular.

Figura 10 - Ilustração da taxa de downloads.



Fonte: npm-stat (2022)

O repositório do Angular no GitHub possui cerca de 84,8 mil estrelas, e é utilizado em cerca de 2.141.284 repositórios públicos (visto em 12/11/2022, GitHub).

5.2.2 Performance

Os testes de performance do Angular foram feitos utilizando a extensão Angular DevTools, utilizando de sua funcionalidade *Profiler*. Na Tabela 3 pode se observar o resultado do teste de adição de elementos, feito então 3 tentativas é tirado para resultado a média aritmética simples desses valores.

Tabela 3 - Tentativas de desempenhos.

| Número Elementos | 1° Tentativa | 2° Tentativa | 3° Tentativa | Média Final |
|------------------|--------------|--------------|--------------|-------------|
| 10 ¹ | 2.1ms | 2.1ms | 2.0ms | 2.06ms |
| 10 ² | 11.8ms | 11.7ms | 12.8ms | 12.1ms |
| 10 ³ | 71.1ms | 69.0ms | 68.3ms | 64.47ms |

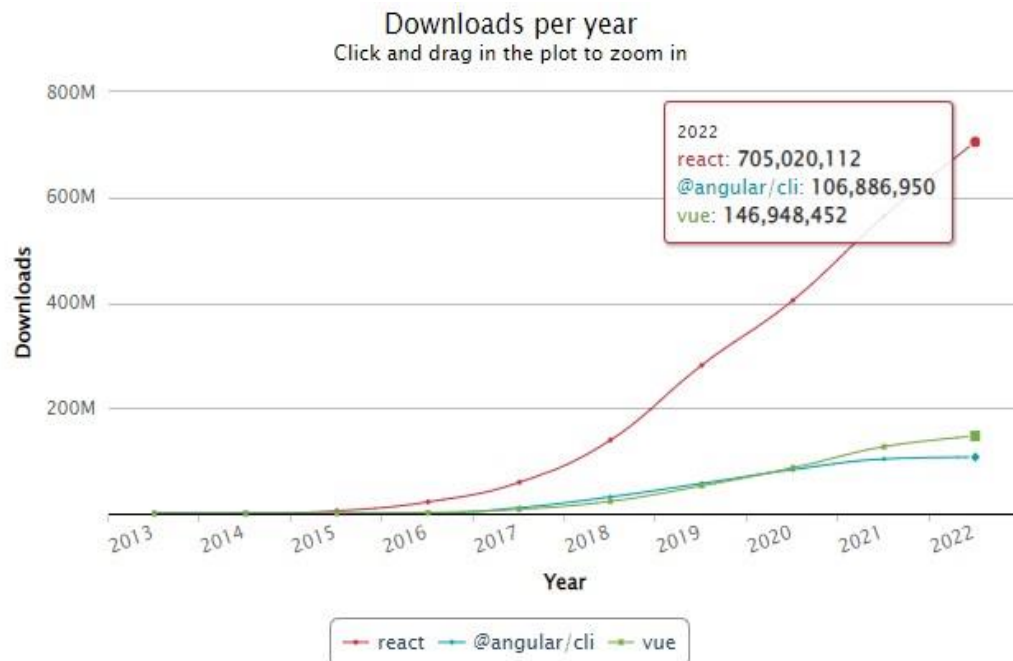
| | | | | |
|-----------------|----------|----------|----------|-----------|
| 10 ⁴ | 443.6ms | 427.9ms | 444.6ms | 438.70ms |
| 10 ⁵ | 4412.6ms | 3820.7ms | 4374.0ms | 4202,43ms |

6. COMPARAÇÃO DOS RESULTADOS

6.1 POPULARIDADE

No site do NPM, o *framework* que mais obteve downloads nesse ano de 2022 (até 12/11/2022) foi o React com mais de 705 milhões de downloads. Em segundo lugar está o Vue, com mais de 146 milhões, seguido pelo @angular/cli com cerca de 106 milhões. Um gráfico detalhado do crescimento do número de downloads dos pacotes ao longo dos anos pode ser visto na Figura 11, retirada do npm-stat.

Figura 11 - Crescimento de downloads das ferramentas.

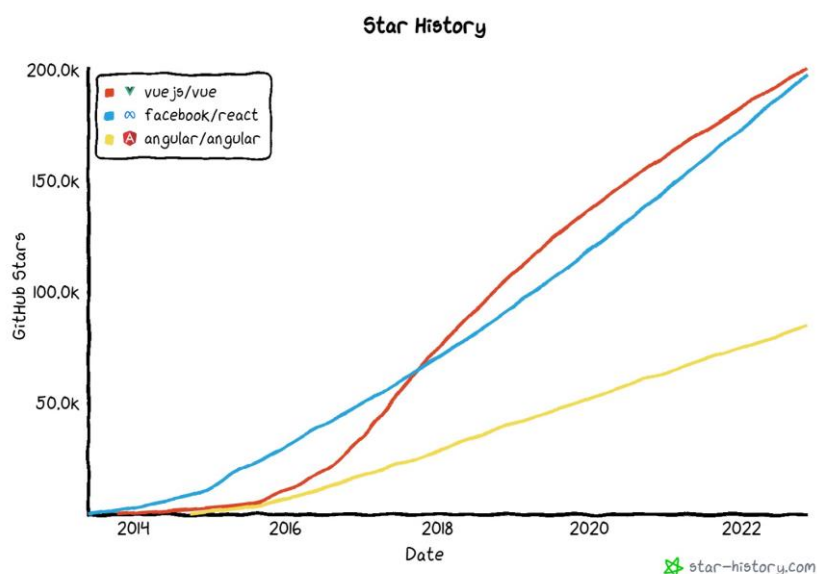


Fonte: npm-stat (2022)

Já no que diz respeito aos seus repositórios no GitHub, o React está na frente no requisito de uso em repositórios públicos, com 12.346.988 repositórios, enquanto o Angular possui 2.141.284 repositórios (visto em 12/11/2022, GitHub). Esse dado não foi encontrado para o Vue em sua página, logo não foi possível fazer a análise desse quesito para o *framework*.

No que diz respeito ao número de estrelas no GitHub, o Vue conta com 201 mil estrelas, ultrapassando o React que possui 198 mil, enquanto o Angular ocupa a terceira posição com 84,8 mil estrelas (visto em 12/11/2022, GitHub). Na Figura 12, um gráfico detalha o crescimento de estrelas de 2014 a 2022.

Figura 12 - gráfico com o crescimento de estrelas de 2014 a 2022.



Fonte: star-history (2022)

A Pesquisa de Popularidade aceitou respostas entre os dias 15 de outubro de 2022 a 15 de novembro de 2022, obtendo 45 respostas neste período. Das respostas obtidas, 41 pessoas trabalhavam no ramo tecnológico, e 37 cursaram/estão cursando alguma faculdade na área da computação.

Nas perguntas acerca das tecnologias abordadas na pesquisa, de 44 respostas, 100% ouviram falar do React, enquanto o Angular ficou com 90,9% e o Vue com 68,2%.

Das tecnologias abordadas, de 37 respostas, o React já foi utilizado por 75,7% dos usuários, enquanto o Angular por 43,2% e o Vue por 27%.

A última pergunta do Formulário obteve 34 respostas, e perguntava qual tecnologia estava inclusa no ambiente de trabalho dos usuários. O React obteve 70,6%, seguido do Angular com 47,1% e do Vue com 17,6%.

Conforme os dados apresentados pode se perceber que no NPM, o React lidera o número de downloads, desde 2016. No GitHub o React também está na frente no número de Repositórios usados, porém perde para o número de estrelas que o Vue conta em sua página. Na pesquisa de popularidade o React também se destaca, sendo a tecnologia mais reconhecida, e utilizada pelo usuário e por seu ambiente de trabalho. Diante desses fatos, o React se mostra em destaque na parte de popularidade até o momento dessa pesquisa e o Vue mostra uma boa comunidade apoiadora no GitHub por seu número de estrelas em repositório.

6.2 PERFORMANCE

O teste de performance foi realizado partindo de um mesmo sistema, uma página onde é possível listar e adicionar contatos. Para tal, foi implementado o mesmo modelo de página utilizando as três tecnologias, React, Angular e Vue, a fim de testar os tempos de renderização das mesmas.

Foi utilizado para medição desse tempo, as plataformas de Avaliação de Rendimento das tecnologias, sendo assim: para o React, Vue e Angular, foi respectivamente utilizado React Dev Tools (versão 4.25.0), Vue.js devtools (versão 6.4.5) e o Angular DevTools (versão 1.0.6).

O primeiro teste foi o de inclusão de linhas, indo de 10 a 10⁵ contatos incluídos de uma só vez. Para o resultado do mesmo foram feitas três tentativas e tirado a média aritmética simples para obter os resultados. Após cada tentativa a página era recarregada.

Diante dessa avaliação, o Vue conseguiu resultados melhores que o React e o Angular em todos os números de elementos propostos. Já o React levou mais tempo do que o Angular para renderizar o número de elementos propostos, como pode ser visto na Tabela 4.

Tabela 4 - Avaliação.

| Número Elementos | React | Vue | Angular |
|------------------|-----------|----------|-----------|
| 10 ¹ | 2.83ms | 1.70ms | 2.06ms |
| 10 ² | 17.57ms | 5.63ms | 12.1ms |
| 10 ³ | 103.8ms | 51.53ms | 64.47ms |
| 10 ⁴ | 755.6ms | 398.37ms | 438.70ms |
| 10 ⁵ | 7185.03ms | 3257.3ms | 4202,43ms |

7. CONSIDERAÇÕES FINAIS

A fim de comparar as três tecnologias, React, Angular e Vue, foram abordados os critérios de popularidade e performance. No critério de popularidade, o React teve destaque nos requisitos propostos, atingindo o maior número de downloads no NPM e de repositórios públicos no GitHub. Em suma, na pesquisa feita pelo artigo, o React atingiu a maior porcentagem nas perguntas que tinham como escolha uma das três tecnologias, sendo elas “Qual das tecnologias você já ouviu falar?”, “Qual das tecnologias você já utilizou?” e “Qual das tecnologias é utilizada na empresa em que você trabalha?”. Diante desses fatos, o React conseguiu se mostrar o que possui mais popularidade dos três no cenário proposto nesta pesquisa.

Na questão dos testes de performance, o VueJS se mostrou mais eficiente nos testes de inserção de elementos, conseguindo ficar na frente independente da quantidade de elementos propostos na inserção.

Diante desses fatos, e no propósito de testes dessa pesquisa, o React se mostrou mais popular e o VueJS mais eficiente.

REFERÊNCIAS

- ANGULAR. Angular. 2022. Disponível em: <<https://angular.io/>>. Acesso em: 15 nov. 2022.
- ANGULAR DevTools. 2022. Disponível em: <11nq.com/OkcJV>. Acesso em: 15 nov. 2022.
- BAELDUNG. The Difference Between a Framework and a Library. 2022. Disponível em: <<https://www.baeldung.com/cs/framework-vs-library>>. Acesso em: 5 set. 2022.
- BRANAS, Rodrigo. AngularJS Essentials. 2014. Disponível em: <<http://www.webshac.com/Angular/AngularJS%20Essentials.pdf>>. Acesso em: 12 set. 2022.
- CINCOVIĆ, Jelica; PUNT, Marija. Comparison: Angular vs. React vs. Vue. Which framework is the best choice? [S. l.], 4 fev. 2020. Disponível em: <11nq.com/0xHdy> Acesso em: 14 nov. 2022.
- REACT: Facebook's Functional Turn on Writing JavaScript. Practice, [S. l.], p. 56-62, 1 dez. 2016. DOI 10.1145/2980991. Disponível em: <https://sci-hub.se/https://dl.acm.org/doi/10.1145/2980991>. Acesso em: 15 nov. 2022.
- GITHUB. 2022. Disponível em: <<https://github.com/>>. Acesso em: 15 nov. 2022.
- MENG, M., Steinhardt, S., & Schubert, A. (2017). Application Programming Interface Documentation: What Do Software Developers Want? Journal of Technical Writing and Communication, 48(3), 295–330. doi:10.1177/0047281617721853
- MOSTRÖM, Mathilda et al. How to choose a web development framework: Analyzing best practices on the adoption of web frameworks. [S. l.], 16 fev. 2022. Disponível em: <<https://www.diva-portal.org/smash/get/diva2:1682657/FULLTEXT01.pdf>> Acesso em: 14 nov. 2022.
- NASCIMENTO, THIAGO DA SILVA. AVALIAÇÃO DE DESEMPENHO DE RENDERIZAÇÃO DE PÁGINAS WEB: UM ESTUDO DE CASO COM TECNOLOGIA
- JAVASCRIPT. 2018. Monografia (Bacharel em Ciência da Computação) - Universidade Federal do Maranhão, [S. l.], 2018. Disponível em: <11nq.com/UkITW> Acesso em: 14 nov. 2022.
- NPM. react. 2022. Disponível em: <<https://www.npmjs.com/package/react>>. Acesso em: 15 nov. 2022.
- NPM. vue. 2022. Disponível em: <<https://www.npmjs.com/package/vue>>. Acesso em: 15 nov. 2022.

NPM. @angular-cli. 2022. Disponível em: <<https://www.npmjs.com/package/@angular/cli>>.

Acesso em: 15 nov. 2022.

NPM-STAT. 2022. Disponível em: <<https://npm-stat.com/>>. Acesso em: 15 nov. 2022.

REACTANGULARVUE. 2022. Disponível em:

<<https://github.com/SandyHoffmann/ReactAngularVue>>. Acesso em: 15 nov. 2022.

REACT Developer Tools. 2022. Disponível em: <11nq.com/GvCro>. Acesso em: 15 nov.

2022.

REACTJS. React. 2022. Disponível em: <<https://reactjs.org/>>. Acesso em: 15 nov. 2022.

SATROM, Brandon. Choosing the Right JavaScript Framework for Your Next Web

Application. [S. l.], 1 jan. 2018. Disponível em: <11nq.com/PIO1S> Acesso em: 14 nov. 2022.

SCHNEIDER, ADOLFO HENRIQUE. Desenvolvimento web com Client Side Rendering: combinando Single Page Application e serviços de backend. 2016. Monografia (Bacharel em

Ciência da Computação) - UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, [S. l.],

2016. Disponível em: <11nq.com/mcqjN> Acesso em: 14 nov. 2022.

SlashData. State of the developer nation. Size of Programming Language Communities in Q3. 2021. Disponível em: <11nq.com/hRWDI>. Acesso em: 10 set. 2022.

STAR-HISTORY. 2022. Disponível em: <<https://star-history.com/>>. Acesso em: 15 nov. 2022.

State of JS. Front-end Frameworks. 2020. Disponível em: <11nq.com/Yxy5f>. Acesso em: 5 set. 2022.

UZAYR, B. Sufyan; CLOUD Nicholas; AMBLER Tim. JavaScript Frameworks for Modern Web Development: the essential frameworks, libraries, and tools to learn right now. 2. ed.

Apress, 2019.

VUE.JS devtools. 2022. Disponível em: <11nq.com/buuW7>. Acesso em: 15 nov. 2022.

VUEJS. Vue.js. 2022. Disponível em: <<https://vuejs.org/>>. Acesso em: 15 nov. 2022.