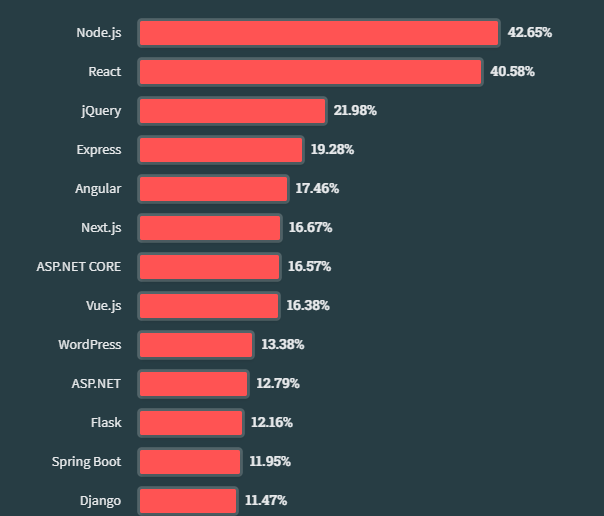
tecnologias mais buscadas e usadas pelos programadores no ano de 2023 de acordo com o maior site de programadores do mundo stackoverFlow, aproximadamente 71.802 pessoas participaram da votação

O Angular é um grande framework mantido pelo google desde 2016 que vem substituindo seu antecessor o AngularJs que apresentava muitos problemas de performance e muitas dificuldades durante o desenvolvimento.

Com isso o Angular é o segundo framework front-end mais popular para a construção de sistemas, perdendo apenas para o React em popularidade. Por que o angular foi escolhido para o nosso projeto ? Primeiramente porque ele é mantido pela Google o'que dá uma segurança que ele será mantido por muito tempo até ser substituído, e tem muitas melhorias novas vindas com as versões 16 e 17 do Angular que trazem novos mundos de possibilidades de fazer a mesma coisa com menos código, mais clareza e desempenho, além disso o nosso site se encaixa perfeitamente no contexto de sites SPA(single page Applications) que seria sites de página única, onde o'que somente mudaria seriam os componentes da página, e outra metodologia de apresentação usada seria a PWA (progressive web applications) que em resumo são sites que podem emular o mesmo comportamento de um aplicativo móvel.

Benefícios do angular:

* renderização no lado do servidor (SSR) que permite que o html seja renderizado no lado do servidor, isso oferece um melhor desempenho e menor tempo de carregamento da página.
* fluxos de dados bidirecionais que permitem a troca de dados entre pai pra filho, o'que facilita muito a troca de dados dinâmicos entre componentes
* padrão **observable** integrado com a biblioteca javascript rxjs que permite que muitos componentes que dependem de um mesmo dado, tenha sua “atenção” para um lugar centralizado que despacha esses dados para quem precisa quando esses mesmos dados mudam.
* roteamento simplificado
* injeção de dependência já é resolvida pelo próprio core do angular
* carregamento lento parcial com o @defere carregamento lento nas próprias rotas
* uso de diretivas que permitem alterar o comportamento do DOM, pode ser reutilizado muitas vezes
* Arquitetura robusta, que indica aonde estão cada componente ou módulo, que ajudam a manter uma alta consistência durante o tempo de desenvolvimento.

Lado ruim do angular

* Como ele pode ser carregado no lado do servidor, pode ocorrer uma incompatibilidade com as API´s da Web em tempo de desenvolvimento, em tempo de compilação elas funcionam normalmente.
* alta curva de aprendizagem

Angular vs React

Aqui veremos o'que os dois tem de melhor e realizar uma breve comparação entre os dois

**criador e quem mantém:**

Angular: google e comunidade

React: criada pelo facebook e mantida pela comunidade

**componentes em cada um:**

Angular: um framework completo para projetos de médio/grande porte, com integração a CLI oficial (suporte para linha de comando), atualmente é construídos com base nos componentes independentes que importam **somente** aquilo que vão usar, sejam serviços externos, diretivas ou módulos do core do angular.

React: uma biblioteca Javascript para projetos de qualquer escala, com integração com CLI não oficial (suporte para linha de comando), é baseado em componentização que indica que os componentes devem ser leves e reutilizáveis e de fácil manutenção para facilitar o desenvolvimento.

**linguagens usadas em cada um:**

Angular: typescript.

React: javascript

**gerenciamento de estado (importantíssimo para nós):**

React: no react tem várias formas de gerenciar estados com o “useSate”, ou “props”, além de outras ferramentas com o Redux ou o MobX que impõe uma série de padrões a serem seguidos que podem aumentar a complexidade do projeto.

Angular: no angular pode-se usar fácilmente somente o padrão observable que quando um dado mudar ele notifica todos os consumidores daquele dado, mas na versão 16 do angular temos as novas implementações dos “sinais angulares” que facilitam e muito e podem substituir o padrão observable, agora a única coisa que precisa ser feita é centralizar esse sinal e quando alguém mudar ele o próprio angular core atualiza os lugares onde esse sinal é usado, isso permite que escrevemos menos código com o mesmo efeito que resulta em maior.

**benefícios gerais:**

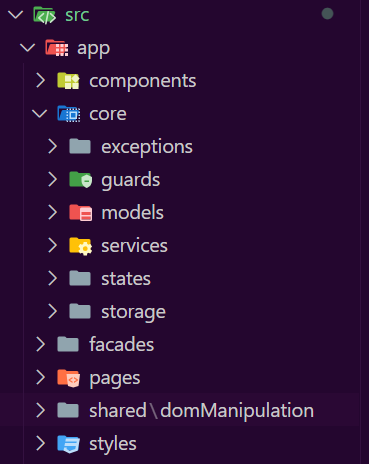
Angular: reatividade poderosa com a biblioteca rxjs que é muito usada, signals e outros módulos do próprio angular como formulários relativos, operadores pipe para mudança no html, diretivas que podem ser criadas e utilizadas, opinião forte sobre sua arquitetura que mantém uma lata consistência.

React: componentização do sistema que deixa ele mais leve e adequado para qualquer tipo de projeto, várias bibliotecas da comunidade que auxiliam para o melhoramento de todo seu ecossistema, sua flexibilidade para trocar de ferramentas de acordo com suas necessidades durante o desenvolvimento.

Por que o angular

Angular foi escolhido por ser um framework completo e com a segurança que ele seria adequado para o nosso projeto que é de médio porte, se encaixa muito bem no modelo de projeto onde o nosso seria uma SPA e PWA onde o angular tem uma excelente performance e é ideal para isso, além de claro um outro ponto muito específico que foi crucial para a tomada de decisão que seria o uso do typescript, isso foi de extrema importância por causa da tipagem que ele traz, facilita o desenvolvimento e fica mais claro o'que cada função deverá retornar e o'que cada parâmetro deveria ser, ter essa clareza é muito importante para manter uma boa legibilidade do código, o react tem sim o typescript mas não é o modelo principal deles e sim o javascript, outro ponto também a considerar é a complexidade onde temos nos arquivos “.JXS” que tem o html e javascript/typescript junto e pode ter os estilos juntos também, no angular temos 1 componente com 4 arquivos, 1 para o template html, outro para a lógica do componente, um para estilo e o último para teste, essa separação de responsabilidades também foi ponto que nos fez escolher o angular,

a organização de pastas e arquitetura que impõem também foi crucial, ajudando a dividir o'que seria cada componente e a qual “modulo” ele pertenceria, por exemplo você tem um modulos de serviços que obtém dados dos usuários de uma api externa, claramente é esperado que esse serviço esteja em services module ou agrupado em services pasta, e é justamente isso que o angular nos obriga a fazer para manter um projeto de fácil leitura e entendimento. a arquitetura de pastas de assemelhava a isso aqui por exemplo



tudo separado de acordo com a sua importância, no core ficavam os componentes mais importantes, pages as páginas do site, e em components os componentes reutilizados mais de 1 vez, facades seria uma camada abstrata para os componentes fazerem suas requisições para as services de forma indireta sem conhecê-las totalmente isso foi feito para desacoplar o codigo burro do inteligente.

e por que não o Vue.js

o Vue é inspirado fortemente no angular, seja na sua estrutura inicial, com sua CLI que é muito semelhante, junta também a ideia de template html, lógica de componente e estilos do componente tudo dentro de um mesmo arquivo, ele junta boas práticas dos dois frameworks mais populares, mas não foi escolhido por que ele não tem uma base, um ecossistema muito sólido, apesar de ser muito usado não vemos grandes empresas como google, facebook usando o Vue é recebendo suporte como o Angular recebe do google e React do facebook, e de novo caímos no problema de responsabilidade única, um mesmo arquivo cuidando de 3 ou mais preocupações diferentes.

fontes do site <https://www.browserstack.com/guide/angular-vs-react-vs-vue>

<https://blog.geekhunter.com.br/react-vs-vue-vs-angular-qual-escolher/#Quais_as_peculiaridades_do_Angular_2>