

O Fibonacci App: Documentação técnica

Neste documento são apresentados [Detalhes Técnicos](#) relativos a implementação do [Fibonacci App](#). Vários desses detalhes de programação são colocados em prática no dia a dia da construção de aplicativos da startup [Finisher.Tech](#).

Conteúdo

1. Licença e código fonte	2
2. Detalhes Técnicos	3
3. Execução local	4
4. Documentação	4
4.1. Processos de geração	4
5. Deploy no GitHub Pages	5
6. TODO	6

Autor deste documento: [Paulo Jerônimo](#).
Última atualização: 2022-03-18 17:11:29 -0300.

Versões online:

Em HTML: <https://finisher.tech/fibonacci-app/docs/technical.pt.html>

Em PDF: <https://finisher.tech/fibonacci-app/docs/technical.pt.pdf>

1. Licença e código fonte

O aplicativo "Fibonacci App" é *open source* e distribuído sob a **licença MIT**. Seu **código fonte** está **inteiramente disponível** em dois repositórios Git: um completo (e privado) e outro público.

O repositório completo:

É utilizado pelos desenvolvedores e **mentorados** da **Finisher.Tech**. Nele estão os **commits mais granulares** realizados nessa aplicação. Contudo, tanto esse repositório quanto o público contemplam os **Detalhes Técnicos** descritos abaixo.

Essa é uma das formas usadas pela startup para *atrair desenvolvedores interessados em seus projetos* de forma que eles entrem para sua equipe. Esse repositório completo inclui, também, *branches* com alternativas de soluções ou algoritmos.

O repositório público (<https://github.com/finishertech/fibonacci-app>):

Contempla commits menos granulares que talvez sejam (ou serão) explicados em vídeos disponíveis nos canais **Finisher.Tech** e **Saúde e Cripto** no YouTube.

Veja a **lista das funcionalidades** que serão disponibilizadas em próximos commits no **repositório público** dessa aplicação.

2. Detalhes Técnicos

Estes são os assuntos cobertos durante a construção do [Fibonacci App](#) que já estão disponíveis em commits disponíveis em seu [repositório público](#):

1. Construção e otimização de algoritmos:
 - a. Implementação não recursiva da [série de Fibonacci](#).
 - i. Várias outras implementações (recursivas ou não) estão [repositório completo](#).
 - b. Busca dos números de Fibonacci que são anterior ou posterior a um número qualquer.
2. Desenvolvimento em **JavaScript e React**:
 - a. Inicialização do projeto: utilizando o [Create React App](#) (CRA).
 - b. Organização estruturada das pastas do projeto.
 - c. Implementação utilizando *Test Driven Development* (TDD) e [Jest](#).
 - d. Codificação de *design* responsivo e uso de [CSS Grid Layout](#).
 - e. Codificação de páginas e componentes em React.
 - f. Uso de [React Hooks](#): *useState*, *useEffect*, *useCallback*, *useMemo*, etc.
 - g. Uso do [React Router](#) (em sua versão 6).
 - h. Uso do [react-device-detect](#) para detecção de dispositivos e adequação desse app a dispositivos móveis.
 - i. Internacionalização utilizando o [react-i18next](#).
 - j. Seleção de temas (*Light* ou *Dark*) através de ajuste dinâmico de [CSS Custom Properties](#).
 - k. Criação de componentes específicos para separar as funcionalidades de internacionalização e de seleção de tema.
 - l. Desenho da função Fibonacci usando o [Katex](#) e [react-katex](#).
3. Deploy em produção:
 - a. Feito no [GitHub Pages](#) através do *package gh-pages*.
4. [Documentação](#).

3. Execução local

```
$ npm install  
$ npm start
```

4. Documentação

Esse documento técnico, e todos os outros que foram gerados em <https://github.com/finishertech/fibonacci-app/blob/main/docs>, são construídos de forma programática. O termo técnico para isso em inglês é *Docs as Code*. Os documentos neste projeto são:

1. Escritos no [formato AsciiDoc](#).
 - a. Contém definições para visualização em diferentes formatos (HTML, PDF) e no ambiente do GitHub ([env-github](#)).
 - b. A equação matemática da função Fibonacci é escrita [LaTeX](#) e convertida para uma imagem no formato [SVG](#) (arquivo [fibonacci.svg](#)). Esse arquivo é, então, usado na geração de PDFs.
 - c. Os links referenciados nos documentos são escritos a partir de um arquivo escrito no formato [YAML](#) ([fibonacci-refs.yaml](#)). A partir dele as referências são geradas no formato AsciiDoc.
2. Gerados nos formatos HTML e PDF através de um [script escrito em Bash](#) que utiliza o [Docker](#) para executar o [Asciidoctor](#).

4.1. Processos de geração

Há dois processos de geração dos documentos.

No *default*, sua geração é feita para o acesso via <http://localhost:3000/fibonacci-app/docs>. Essa URL é a padrão ao iniciar o [Fibonacci App](#) localmente através do `npm start`. Nesse caso, o commando para geração é este:

```
$ docs/build.sh
```

Outro processo é para a atualização/ visualização dos documentos de forma *detached*, ou seja, sem a necessidade de o [Fibonacci App](#) estar em execução, mas com um servidor HTTP em execução em <http://localhost:3000>. Essa será a URL base para os documentos no caso de você executar estes comandos:

```
$ docs/build.sh detached  
$ serve -s public/docs &> /dev/null & ① ②
```

① Inicia um servidor HTTP local (instalado via `npm install -g serve`) em background.

② Veja [alternativas p/ execução de um servidor HTTP](#) utilizando diferentes linguagens de

programação.

5. Deploy no GitHub Pages

```
$ npm run deploy
```



Esse procedimento só será executado com sucesso se você possuir permissão para alterar a [branch gh-pages](#) da aplicação!

6. TODO

- **Refactorings:**

- Separar os componentes em `FibonacciNumbers.js` em diferentes arquivos e utilizar o `useContext` (Feito).
- Utilizar o [Styled Components](#).
- Melhorar o *design* do frontend (estilização no geral).

- **Novas funcionalidades:**

- Salvar os números de Fibonacci que uma pessoa utiliza em um *backend* ([Firebase](#) ou [Supabase](#)) que estará acessível (para leitura) para todos.
 - Possibilitar a pesquisa dos números Fibonacci registrados por um usuário específico.
 - Um usuário não poderá editar as descrições associadas a números cadastros por outros usuários.
- Desenhar a espiral de Fibonacci (com o [p5.js](#) ou o [d3.js](#)) conforme a navegação do usuário pelos números da série na página Numbers.
 - Esse desenho deverá ser interativo (obedecendo a navegação do usuário nos números da série).
- Possibilitar que o usuário se registre utilizando seu email ou telefone.
 - Se o usuário for brasileiro e informar um CEP, o aplicativo consulará o [ViaCEP](#) para auxiliar o usuário a entrar com seu endereço completo.
- Exigir o login do usuário para que ele possa registrar seus números.
 - Utilizar JWT para autenticação do usuário e, também, OAuth para possibilitar login social.
- Acrescenter algumas rotas externas (ou que estejam em `/public`). Exemplos:
 - <https://finisher.tech/fibonacci-app/public/technical.pt.html> ← deve enviar o usuário para a página `technical.pt.html` localizado na pasta `public` junto aos fontes do projeto. Da mesma forma, <https://finisher.tech/fibonacci-app/public/xpto.json> deve possibilitar o browser apresentar esse arquivo, caso ele exista na pasta `public`.
 - <https://finisher.tech/fibonacci-app/github> ← deve redirecionar o usuário para o [repositório público](#).
- **Premium (para os [mentorados](#)):**
 - Versão que altera o *backend* para uso da [Blockchain](#) do [Ethereum](#).
- **Bugs conhecidos:**
 - Ao se efetuar uma troca da linguagem é possível notar que o botão que indica qual o fator de incremento/decremento não troca de vírgula para ponto (ou vice-versa).