

Fibonacci App: Documento técnico

O aplicativo web "[Fibonacci App](#)", desenvolvido por [Paulo Jerônimo](#), tem o intuito de apresentar a [Série de Fibonacci](#), para o público alvo da [mentoria da Finisher.Tech](#), de uma forma programável (em [JavaScript e React](#)). [Detalhes Técnicos](#) de programação, que são colocados em prática no dia a dia da construção de aplicativos da [Finisher.Tech](#), são apresentados nesse aplicativo.

Este documento que você está lendo foi gerado em 2022-03-05.
Seu código fonte talvez já esteja mais atualizado e disponível [neste link](#).

Conteúdo

1. Licença e código fonte	1
2. Detalhes Técnicos	2
3. Execução Local	3
4. Deploy no GitHub Pages	3
5. TODO	4

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 da [Finisher.Tech](#) e seus mentorados. 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 que usamos para *atrair desenvolvedores interessados em nossos projetos* a entrarem para nossa equipe. Esse repositório completo inclui, também, *branches* com alternativas de soluções ou algoritmos.

O repositório público:

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 de funcionalidades](#) que serão disponibilizadas em próximos commits no [repositório público](#) dessa aplicação.

2. Detalhes Técnicos

Estes são assuntos cobertos durante a construção [dessa aplicação](#) e 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 intenso 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 tema *light* ou *dark* através de ajuste dinâmico de [CSS Custom Properties](#).
 - k. Criação de componentes específicos que separam as funcionalidades de internacionalização e de seleção de tema.
3. Deploy em produção:
 - a. Feito no [GitHub Pages](#) através do *package gh-pages*.
4. Documentação:
 - a. Uso do [Asciidoctor](#) e do [Docker](#).
 - b. Ajustada para a visualização em diferentes formatos (*backends*): HTML, PDF e GitHub.

3. Execução Local

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

4. 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!

5. TODO

- **Refactorings:**

- Separar os componentes em `FibonacciNumbers.js` em diferentes arquivos e utilizar o `useContext`.
- 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 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 (utilizando o p5.js) conforme a navegação do usuário pelos números da série na página Numbers.
- Possibilitar que o usuário se registre utilizando seu email ou telefone.
 - Se o usuário for brasileiro e informar um CEP, o aplicativo consultará 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.