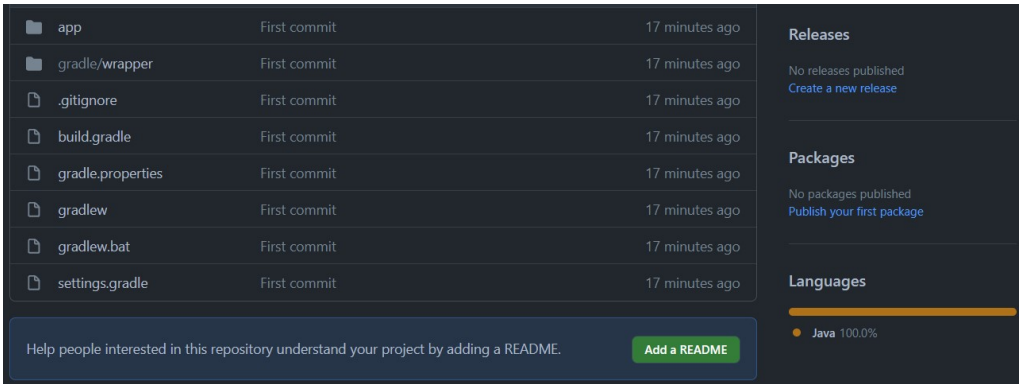
Artigos > **Programação**

Como escrever um README incrível no seu Github

Camila Fernanda Alves
Atualizado em 1 de Junho

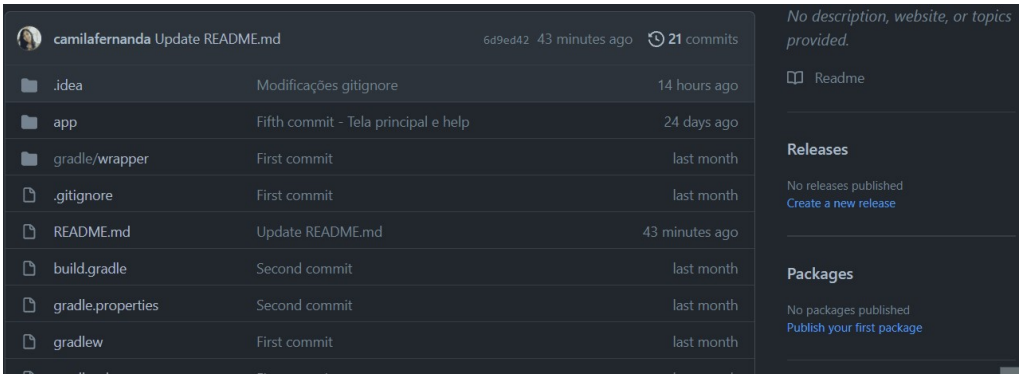
COMPARTILHE

Tendo em vista o *mesmo projeto* em repositórios remotos no GitHub, a famosa rede social da pessoa que desenvolve, qual das duas abordagens abaixo é mais atrativa para você?



"

Repositório: GlicoCare2



"

Repositório: GlicoCare

Com certeza a *segunda opção*, não é mesmo? Pois na primeira opção nem iríamos saber direito do que se trata o projeto. E o diferencial do segundo repositório foi a adição de um arquivo **README** bem bacana que deixou o projeto mais atrativo e explicativo.

Confira neste artigo:

- O que é README? Leia-me, o que seria isso?
- Por que fazer um README?
- O que é bacana de se ter em um README
- Título e Imagem de capa
- Badges

- Índice
- Descrição do Projeto
- Status do Projeto
- Funcionalidades e Demonstração da Aplicação
- 🔨 Funcionalidades do projeto
- Acesso ao projeto
- Tecnologias utilizadas
- Pessoas Contribuidoras
- Pessoas Desenvolvedoras do Projeto
- Licença
- Referências de README
- Partiu colocar a mão na massa?

O que é README? Leia-me, o que seria isso?

O README é um arquivo com extensão .md, ou seja, ele é escrito em **Markdown** que é uma linguagem de marcação utilizada para converter o texto em um HTML válido. Caso queira saber mais sobre, temos esse artigo que explica muito bem como funciona e como escrever anotações com essa linguagem.

Nas plataformas de repositórios remotos, como o GitHub, a função desse arquivo é apresentar informações do projeto, como:

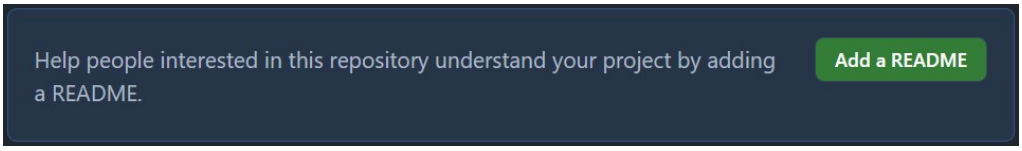
- Descrição do seu projeto;
- Funcionalidades;
- Como os usuários podem utilizá-lo;
- Onde os usuários podem encontrar ajuda sobre seu projeto;
- Autores do projeto.

Por que fazer um README?

Depois de trabalhar bastante e desenvolver seu projeto bacana, provavelmente você irá subir para o GitHub e mostrar para comunidade o que você fez.

Para isso, é legal documentar o projeto para quem for visitar seu repositório saiba do que se trata, assim como foi visto no primeiro exemplo.

E podemos fazer isso por meio do **README**, que é o primeiro arquivo a ser visto, ou seja, é a *porta de entrada para o seu projeto*. O próprio GitHub faz essa recomendação ao subir algum projeto *sem* esse arquivo:



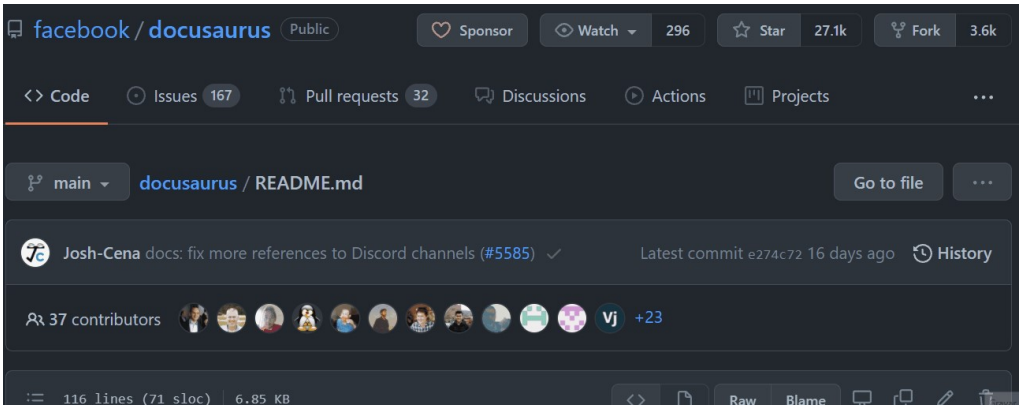
> *Tradução: Ajude as pessoas interessadas neste repositório a entender seu projeto adicionando um README.*

Além disso, algumas pessoas utilizam o perfil GitHub como portfólio, se esse for o seu caso, é interessante apostar em arquivos README para deixar seus projetos mais atrativos até mesmo para recrutadores, como explicado nesse artigo da Jornada de um estagiário. Caso esse não seja o seu objetivo, é válido destacar que nem sempre queremos que os usuários se interessem ou colaborem com nosso repositório, nesses casos é dispensável o uso desse arquivo.

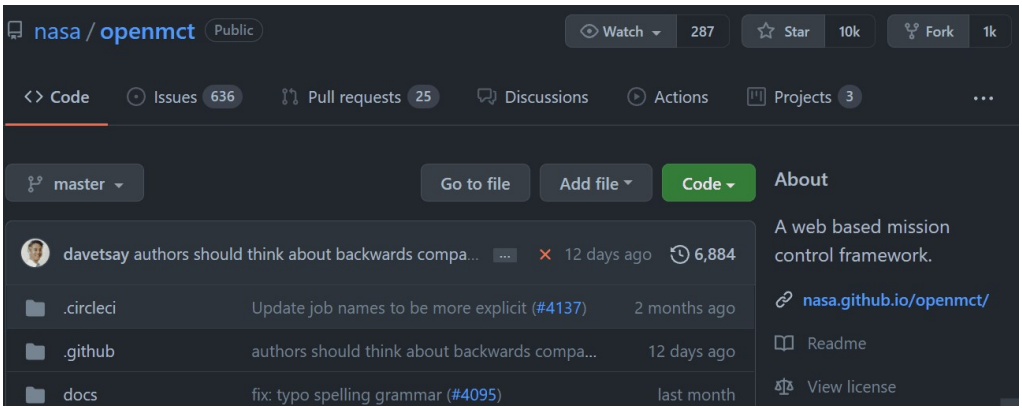
O que é bacana de se ter em um README

Analisando repositórios famosos, é possível notar a presença de arquivos README bem legais.

Como esse repositório Docusaurus do **Facebook**, que é um projeto feito para auxiliar a construir, implantar e manter sites de projetos de código aberto.



E o projeto Open MCT da **Nasa** que é utilizado para análise de dados de missões de espaçonaves, bem como planejamento e operação de sistemas de rover experimentais.



É possível notar que os dois *não* possuem um padrão exatamente igual, cada um possui sua identidade. Porém, conseguimos ver alguns pontos que podemos destacar que são legais de se ter no seu README, como:

- Título e Imagem de capa;
- Badges;
- Índice;
- Descrição do Projeto;
- Status do Projeto;
- Funcionalidades e Demonstração da Aplicação;
- Acesso ao Projeto;
- Tecnologias utilizadas;
- Pessoas Contribuidoras;
- Pessoas Desenvolvedoras do Projeto;
- Licença.

Título e Imagem de capa

Assim que você adicionar um README, ele já irá iniciar com o título sendo o nome do seu repositório. Mas você pode mudar ele e colocar um **nome descritivo**. Nesse momento abuse da criatividade.

Ao escolher o título, você pode colocá-lo dessa maneira:

```
# Seu título aqui
```

Ou, caso queira colocar ele **centralizado**, você pode utilizar **tags do HTML** que funcionam normalmente, dessa forma:

```
<h1 align="center"> Seu título aqui </h1>
```

Feito isso, caso queira, você pode fazer uma **capa** ou **logo do projeto** para colocar após o título. Particularmente, gosto bastante do [Canva] para fazer artes, com o plano gratuito já é possível encontrar algumas ferramentas

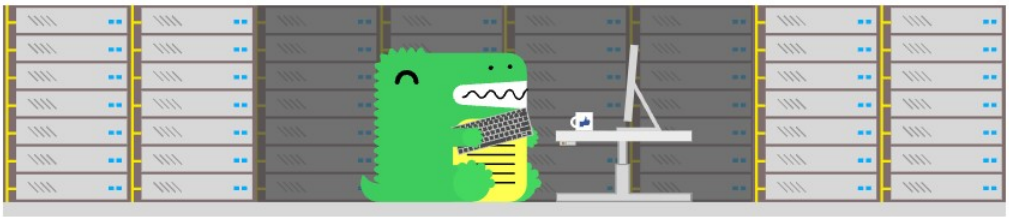
legais. Caso queira aprender mais, confira nosso Curso de Introdução ao Canva.

Eventualmente, caso a logo for uma arte do título, como no exemplo do GlicoCare, ela pode substituir o título:

GlicoCARE

Porém, pode ser utilizados os dois também, como utilizado no Docusaurus:

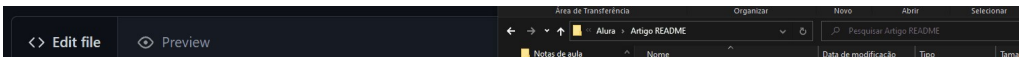
Docusaurus

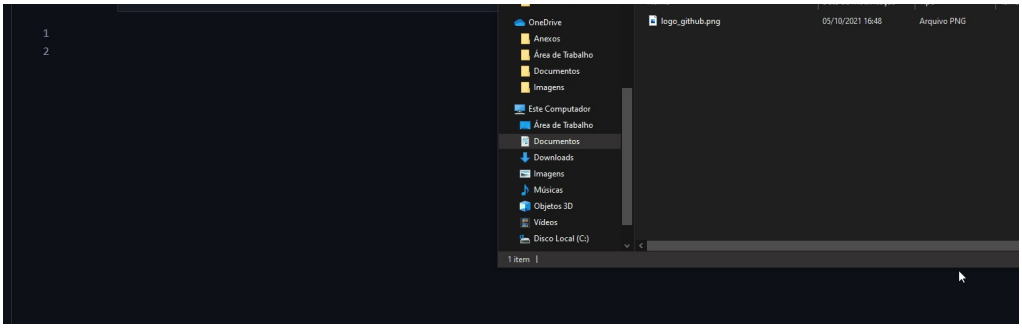


Ou CacheLib do Facebook também:

CacheLib

Mas como colocar a imagem no arquivo? Assim que você fizer ou encontrar sua arte, você pode abrir ela no explorador de arquivos e arrastá-la para o arquivo do README em edição, segue um exemplo feito no Windows:





Dentro dos colchetes `[]`, irá aparecer o nome do arquivo da imagem como descrição, mas é interessante que você descreva detalhadamente do que se trata para ajudar na **acessibilidade**. E dentro dos parênteses aparece um link da sua imagem que o GitHub gera ao converter e hospedar ela.

Outra maneira de colocar imagem, que é mais segura, seria subindo a imagem para **seu próprio repositório** ou utilizando **serviços de hospedagem de imagens** como o imgur ou o pasteboard e colocar o link gerado em markdown, dessa forma:

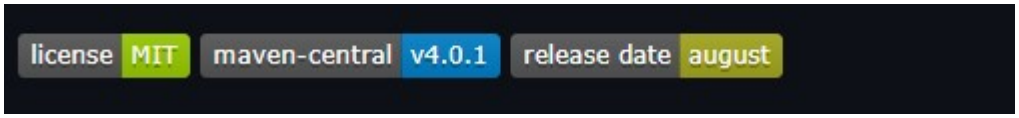
```
![descrição da imagem](url da imagem gerado pelo serviço de hos
```

Você também pode utilizar imagens disponibilizadas na internet pegando o link e colocando da mesma forma citada anteriormente, porém *não* é recomendado, pois pode ocorrer que a imagem seja deletada e seu repositório fique sem ela depois.

Badges

Alguns repositórios utilizam **badges**, que na tradução literal é distintivo, emblema ou insígnia. Seus objetivos são indicar o **estado atual do projeto**, **licença** caso tenha, **versões**, **dependências**, **testes** e entre outros.

Um exemplo do uso de badges em um repositório do Dropbox, o Dropbox Core SDK for Java 8+:



Nele foi utilizado badges para:

- **Licença**: Instituto de Tecnologia de Massachusetts (MIT);
- **Versão da ferramenta de gerenciamento de dependências, Maven**: versão 4.0.1;
- **Data da última versão do projeto**: agosto.

Caso queira fazer suas badges, você pode utilizar o Shields.io, ele fornece na página principal diversos *exemplos de Badges* e, além disso, nele você pode colar o link do seu repositório do GitHub na caixa de texto inicial, assim automaticamente ele irá *sugerir algumas Badges* para você utilizar no seu projeto, fornecendo o link da Badge para copiar e colar no seu README.

Exemplos:

1) Status do projeto:

Código gerado:

```
![Badge em Desenvolvimento](http://img.shields.io/static/v1?lab
```

Resultado:

Caso queira deixar *centralizado*, pode utilizar a tag `align` do HTML também, dessa forma:

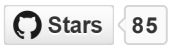
```
<p align="center">
<img src="http://img.shields.io/static/v1?label=STATUS&message=
</p>
```

2) Stars do projeto:

Código gerado:

```
![GitHub Org's stars](https://img.shields.io/github/stars/camil
```

Resultado:



Índice

O GitHub gera **automaticamente** um índice para arquivos README, tendo como base os títulos da seção. Para visualizá-lo, selecione o *ícone de menu* no canto superior esquerdo do arquivo.

Principalmente em READMEs extensos, essa é uma ferramenta excelente para navegar pelo documento, pois redireciona o usuário para o tópico selecionado.

Entretanto, caso você tenha interesse em fazer seu próprio índice para ser visualizado no arquivo, também é possível fazer em Markdown.

Exemplo:

```
# Índice

* [Título e Imagem de capa](#Título-e-Imagem-de-capa)
* [Badges](#badges)
* [Índice](#índice)
* [Descrição do Projeto](#descrição-do-projeto)
* [Status do Projeto](#status-do-Projeto)
* [Funcionalidades e Demonstração da Aplicação](#funcionalidade
* [Acesso ao Projeto](#acesso-ao-projeto)
* [Tecnologias utilizadas](#tecnologias-utilizadas)
* [Pessoas Contribuidoras](#pessoas-contribuidoras)
* [Pessoas Desenvolvedoras do Projeto](#pessoas-desenvolvedoras
* [Licença](#licença)
* [Conclusão](#conclusão)
```

Resultado:

- Título e Imagem de capa
- Badges

- Índice
- Descrição do Projeto
- Status do Projeto
- Funcionalidades e Demonstração da Aplicação
- Acesso ao Projeto
- Tecnologias utilizadas
- Pessoas Contribuidoras
- Pessoas Desenvolvedoras do Projeto
- Licença
- Conclusão

Descrição do Projeto

Como é um objetivo do arquivo README **descrever o projeto**, é legal que você apresente logo após o título ou imagem de capa e/ou Badges caso tenha, uma breve explicação do seu projeto com seu objetivo principal.

Você pode adotar uma descrição mais objetiva, como a do Dropbox Core SDK for Java 8+:

> *Tradução: Uma biblioteca Java para acessar a API Core v2 baseada em HTTP do Dropbox. Este SDK também oferece suporte ao Core API v1 mais antigo, mas esse suporte será removido em algum momento.*

Ou pode detalhar mais, como feito no projeto GlicoCare e apresentar figuras que exemplificam o projeto, caso tenha:

Status do Projeto

Caso você opte por não utilizar Badges sobre o status do projeto, é interessante que você coloque essa informação no próprio texto mesmo. Exemplo:

```
> :construction: Projeto em construção :construction:
```

Resultado:

> 🚧 *Projeto em construção* 🚧

Ou caso queira centralizar:

```
<h4 align="center">
    :construction:  Projeto em construção  :construction:
</h4>
```

Nesse exemplo foi utilizado o **emoji** `:construction:` , mas você pode utilizar em qualquer lugar do arquivo, como *antes dos subtítulos*. Nesse Gist do

Rafael Xavier de Souza, você pode encontrar uma variedade de emojis para utilizar no seu README e deixá-lo mais descontraído.

Funcionalidades e Demonstração da Aplicação

Você pode **listar as funcionalidades** do seu projeto para facilitar o entendimento do usuário. Para isso, você pode fazer dessa maneira:

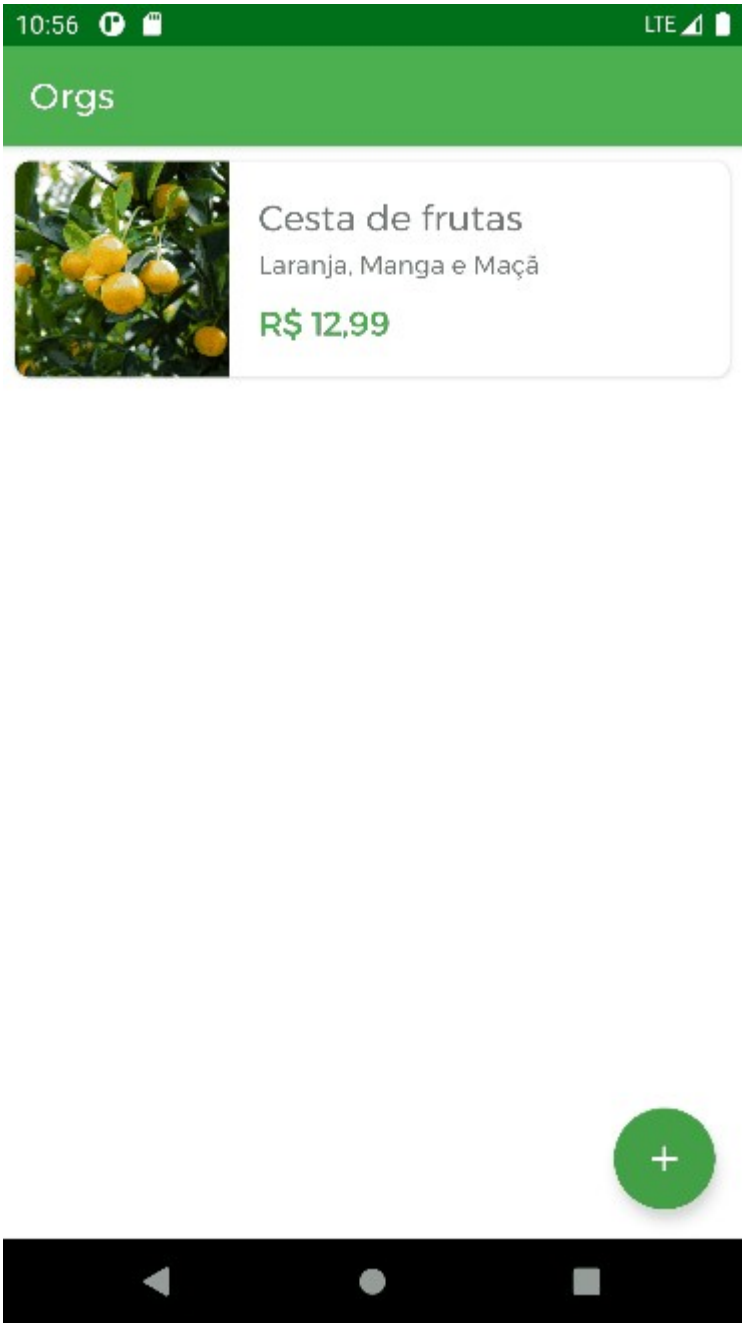
```
# :hammer: Funcionalidades do projeto

- `Funcionalidade 1`: descrição da funcionalidade 1
- `Funcionalidade 2`: descrição da funcionalidade 2
- `Funcionalidade 2a`: descrição da funcionalidade 2a relacionad
- `Funcionalidade 3`: descrição da funcionalidade 3
```

Resultado:

🔨 Funcionalidades do projeto

- `Funcionalidade 1` : descrição da funcionalidade 1
- `Funcionalidade 2` : descrição da funcionalidade 2
- `Funcionalidade 2a` : descrição da funcionalidade 2a relacionada à funcionalidade 2
- `Funcionalidade 3` : descrição da funcionalidade 3

Além disso, se for possível, é interessante apresentar as funcionalidades com um **exemplo visual do projeto**, como gif, imagens ou vídeo. Segue como exemplo o gif que foi utilizado no repositório [Android com Kotlin - Personalizando UI]():

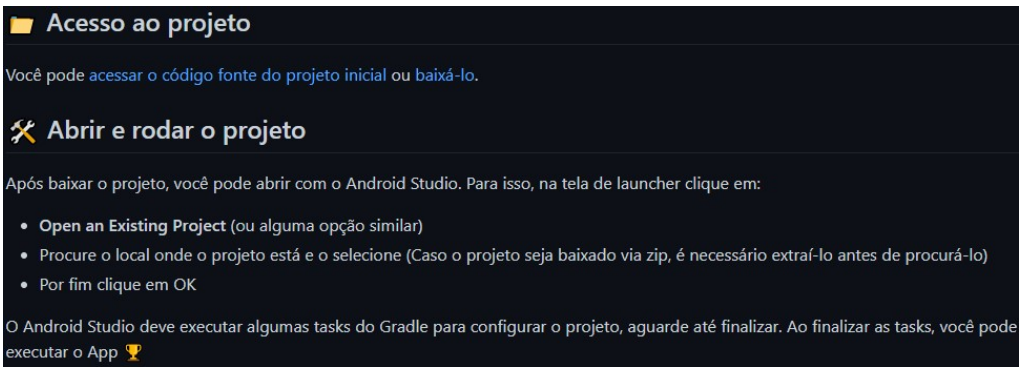


Lembrando que o procedimento para colocar gif é o mesmo adotado para imagens e você pode gravar gifs com gravadores de tela, como o Acethinker.

Acesso ao projeto

Caso o seu projeto esteja no ar com algum serviço de hospedagem, você pode **disponibilizar o link** para o mesmo. Caso contrário, você pode apostar em gifs e imagens, como citado anteriormente, bem como **indicar como o usuário pode baixar o projeto, abrir e executar**.

Novamente, tendo como exemplo o repositório Android com Kotlin - Personalizando UI que citou como o usuário pode ter acesso ao projeto, bem como abrir e rodar o mesmo:



Como pode ser feito em Markdown:

```
# 📂 Acesso ao projeto

**Indique como é possível baixar ou acessar o código fonte do pr

# 🛠️ Abrir e rodar o projeto

**Apresente as instruções necessárias para abrir e executar o pr
```

Tecnologias utilizadas

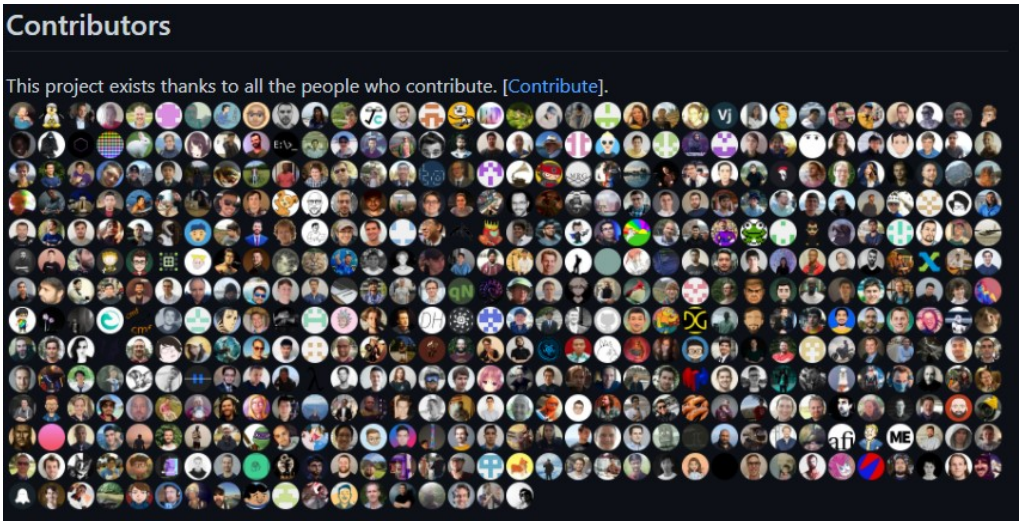
Você também pode citar as **tecnologias utilizadas no projeto**, é uma ótima forma de demonstrar o que você anda estudando nesse mar que é a tecnologia.

Você pode citar com textos, como o que foi feito no repositório Edige - POO:

- `Java 8`
- `InteliJ IDEA`
- `Paradigma de orientação a objetos`

Pessoas Contribuidoras

Caso o seu projeto tenha contribuidores, é bacana adicionar eles no README. Um exemplo fantástico disso é o Docusaurus, onde se tem as fotos de todos os contribuidores e um link para outro documento markdown com recados importantes para quem deseja contribuir também:

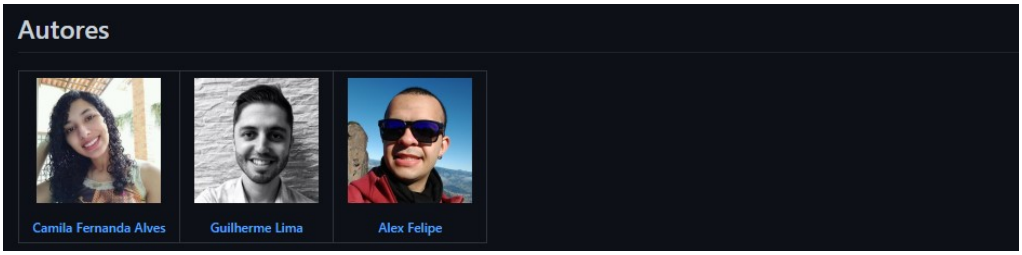


Pessoas Desenvolvedoras do Projeto

E agora entra você! É importante que você coloque sua foto também, caso não goste de fotos, vale o user padrão do GitHub ou fazer seu próprio Octocat. Além disso, você pode linkar seu usuário, para caso algum usuário queira entrar em contato ou reportar algo.

```
# Autores

| [<img src="https://avatars.githubusercontent.com/u/37356058?v=
| :---: | :---: | :---: |
```

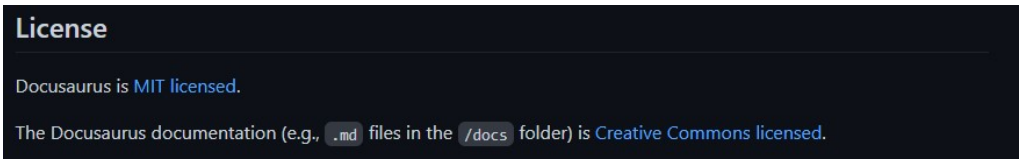
Resultado:



Licença

Geralmente, os repositórios públicos no GitHub são utilizados para compartilhar softwares de código aberto. Porém, para que um repositório seja realmente de código aberto, ele precisa obter uma licença para que outros tenham a liberdade de usar, alterar e distribuir o software.

Portanto, caso seu repositório tenha uma licença, é essencial que você coloque ela no seu README. Como feito no Docusaurus:



> *Tradução: Docusaurus é licenciado pelo MIT. A documentação do Docusaurus (por exemplo, arquivos .md na pasta ./docs) é licenciada pelo Creative Commons.*

Referências de README

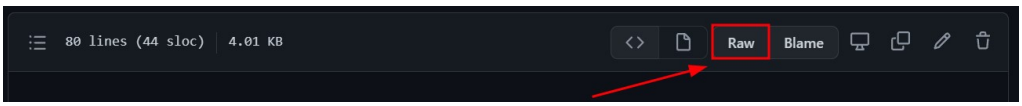
Segue todos os READMEs citados aqui, bem como alguns para se inspirar:

- Docusaurus;
- Open MCT;
- GlicoCare;
- CacheLib;
- Dropbox Core SDK for Java 8+;
- Edige - POO;
- Android com Kotlin - Personalizando UI.

E para se inspirar, segue alguns templates legais:

- Template de README em Português feito pela Diana Regina, que é um presente do artigo da autora, o qual foi referência para escrita desse;
- Template de README em Inglês do Othneil Drew.

Lembrando que para olhar o código fonte, em Markdown, clique na opção `Raw` no canto superior direito do README:



Partiu colocar a mão na massa?

Agora sim espero que você tenha notado a importância de um arquivo README e como fazer um bacana. Caso você tenha feito algum projeto conosco em imersões, cursos, challenges ou bootcamps, se liga nesse template que nossa Escola de Mobile fez para vocês utilizarem como base.

E para aprender mais sobre **Git e GitHub**, confira:

- Websérie: Git e Github para sobrevivência | com DevSoutinho
- Curso de Git e Github: Controle e compartilhe seu código
- Curso de Git e Github: Estratégias de ramificação, Conflitos e Pull Requests

E tem um video incrível do DevSoutinho, um Alura Star, que também aborda o Readme:

Camila Fernanda Alves

Sou quase engenheira pela Universidade Federal de Uberlândia e instrutora aqui na Alura. Atuei como estagiária do Scuba Team e já realizei algumas monitorias de Física, Circuitos Elétricos e até mesmo Processamento de Sinais. Atualmente, estou iniciando na produção de conteúdos relacionados a DevOps e também estou aqui para mergulhar em tecnologia com vocês e, sempre que possível, tentar ajudá-los a encontrar as respostas que precisam nos fóruns. No tempo livre, gosto de jogos da Nintendo e Valorant.

Artigo Anterior

Design patterns: Breve introdução aos padrões de projeto

Próximo Artigo

Guia pra se tornar dev web, estruturas de dados no React, Data Science e mais Alura Stars

Veja outros artigos sobre
Programação

Quer mergulhar em tecnologia e aprendizagem?

Receba a newsletter que o nosso CEO escreve pessoalmente, com insights do mercado de trabalho, ciência e desenvolvimento de software

Escreva seu email

ME INSCREVA

Nossas redes e apps

Institucional

Sobre nós

A Alura

Como Funciona

Trabalhe conosco

Para Empresas

Para Escolas

Política de Privacidade

Compromisso de Integridade

Termos de Uso

Status

Todos os cursos

Depoimentos

Instrutores(as)

Dev em <T>

Conteúdos

Alura Cases

Imersões

Artigos

Podcasts

Artigos de educação corporativa

Fale Conosco

Email e telefone

Perguntas frequentes

Novidades e Lançamentos

Email* ENVIAR

CURSOS

Cursos de Programação
Lógica | Python | PHP | Java | .NET | Node JS | C | Computação | Jogos | IoT

Cursos de Front-end
HTML, CSS | React | Angular | JavaScript | jQuery

Cursos de Data Science
Ciência de dados | BI | SQL e Banco de Dados | Excel | Machine Learning | NoSQL | Estatística

Cursos de Inteligência Artificial
IA para Programação | IA para Dados

Cursos de DevOps
AWS | Azure | Docker | Segurança | IaC | Linux

Cursos de UX & Design
Usabilidade e UX | Vídeo e Motion | 3D

Cursos de Mobile
React Native | Flutter | iOS e Swift | Android, Kotlin | Jogos

Cursos de Inovação & Gestão
Métodos Ágeis | Softskills | Liderança e Gestão | Startups | Vendas