# Guia: Como contribuir em Open Source

## Quer contribuir em open source e não sabe como e nem por onde começar? Esse pequeno guia pode te ajudar.

## Índice

* [Introdução](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "introducao)
* [O que é Open Source?](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "o-que-e-open-source)
  + [Livre redistribuição](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "livre-redistribuicao)
  + [Trabalhos derivados](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "trabalhos-derivados)
  + [Sem discriminação contra pessoas ou grupos](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "sem-discriminacao-contra-pessoas-ou-grupos)
* [Por que contribuir?](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "por-que-contribuir)
  + [Melhorar seus conhecimentos](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "melhorar-seus-conhecimentos)
  + [Ensinar e aprender](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "ensinar-e-aprender)
  + [Criar uma boa imagem e ajudar na carreira](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "criar-uma-boa-imagem-e-ajudar-na-carreira)
* [Como contribuir?](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "como-contribuir)
  + [Você gosta de design?](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "voce-gosta-de-design)
  + [Você gosta de planejar eventos?](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "voce-gosta-de-planejar-eventos)
  + [Você gosta de escrever?](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "voce-gosta-de-escrever)
  + [Você gosta de organização?](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "voce-gosta-de-organizacao)
  + [Você gosta de codar?](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "voce-gosta-de-codar)
  + [Você gosta de ajudar pessoas?](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "voce-gosta-de-ajudar-pessoas)
  + [Você pode simplesmente ajudar em outras coisas que não sejam código!](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "voce-pode-simplesmente-ajudar-em-outras-coisas-que-nao-sejam-codigo)
* [Orientando-se para um novo projeto](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "orientando-se-para-um-novo-projeto)
  + [Anatomia básica de um projeto open source](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "anatomia-basica-de-um-projeto-open-source)
* [Encontrando um projeto para contribuir](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "encontrando-um-projeto-para-contribuir)
  + [Buscas diretas no Github](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "buscas-diretas-no-github)
  + [Ferramentas para buscar projetos](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "ferramentas-para-buscar-projetos)
* [Uma checklist para antes de contribuir](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "uma-checklist-para-antes-de-contribuir)
* [Como submeter uma contribuição?](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "como-submeter-uma-contribuicao)
  + [Comunique efetivamente](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "comunique-efetivamente)
  + [Faça seu dever de casa antes](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "faça-seu-dever-de-casa-antes)
  + [Faça seus pedidos de forma curta e direta](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "faça-seus-pedidos-de-forma-curta-e-direta)
  + [Mantenha toda comunicação pública](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "mantenha-toda-comunicação-pública)
  + [Respeite as decisões do projeto](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "respeite-as-decisoes-do-projeto)
  + [Abrindo uma Issue](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "abrindo-uma-issue)
  + [Abrindo um Pull Request](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "abrindo-um-pull-request)
* [O que acontece depois?](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "o-que-acontece-depois)
  + [Você não recebe uma resposta](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "voce-nao-recebe-uma-resposta)
  + [Respondem e pedem mudanças](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "respondem-e-pedem-mudancas)
  + [Sua contribuição não foi aceita](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "sua-contribuição-nao-foi-aceita)
  + [Sua contribuição foi aceita](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "sua-contribuiçao-foi-aceita)
* [Mais alguns artigos que podem lhe ajudar](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "mais-alguns-artigos-que-podem-lhe-ajudar)
* [Conclusão](https://willianjusten.com.br/guia-como-contribuir-em-open-source/" \l "conclusao)

## Introdução

Olá pessoal, primeiro de tudo, quero agradecer ao apoio que muita gente tem dado ao meu mais novo [Curso de Git e Github na Vida Real](https://www.udemy.com/git-e-github-na-vida-real/?couponCode=PROMOJUL21) e também falar que esse post é totalmente ligado ao curso, pois é uma dúvida que nesses últimos dias surgiu para caramba, muitos me perguntando como contribuir, o que fazer para achar algo, enfim, várias coisas, então resolvi escrever algumas coisinhas, espero que seja útil. Muita coisa eu estou pegando do [guia de open source do Github](https://opensource.guide/how-to-contribute/) e outras mais fontes, fica tranquilo que vou colocando link em tudo.

Enquanto escrevo vou ouvindo uma banda que tem dominado minhas playlists nos últimos dias, a banda se chama [Naxatras](https://open.spotify.com/artist/6HN1s0JzLowapZ7nhOAJ71?si=HdwvHdUVQzqaLdrjvcOYLA) e tem um som absurdamente bom, vai por mim! Eles são da Grécia e tem um som muito muito legal, com misturas de Blues, psicodelia e até umas pitadas de música árabe em algumas músicas. A música [Space Tunnel](https://open.spotify.com/track/630AJdTUlZ7M9SlbfGpMaa?si=H4xmNgu3Qf2dkzNgTA3JWA) começa com um baixo de arrepiar! Enfim, vamos ao guia e curta esse som!

## O que é Open Source?

Muita gente acha que open source significa só o acesso ao código fonte de um projeto.

Mas Open Source não é só o acesso ao código, para que algo seja realmente considerado open source, precisa-se seguir alguns critérios, que são definidos na [opensource.org](https://opensource.org/osd), seguem alguns abaixo:

### Livre redistribuição

A licença não deve restringir qualquer parte de vender ou distribuir o software e a licença não pode exigir taxas de royalties ou quaisquer outras taxas para vendas.

### Trabalhos derivados

A licença deve permitir modificações e trabalhos derivados, e deve permitir a distribuição sob os mesmos termos da licença do software original.

### Sem discriminação contra pessoas ou grupos

Exatamente, algo que muita gente pode não saber. Mas a licença não pode discriminar nenhuma pessoa ou grupo de pessoas.

Bom, apesar da maioria já ter uma ideia do que é Open Source, talvez algumas das coisas acima sejam novidade, aproveita e abra o link [opensource.org](https://opensource.org/osd) e dá uma lida nos outros pontos. E com isso vamos a segunda pergunta mais comum.

## Por que contribuir?

Contribuir com o open source pode ser recompensador! Uma maneira de aprender, ensinar e ter uma experiência talvez nunca imaginada. E algumas razões:

### Melhorar seus conhecimentos

Se você é um programador, um designer, até mesmo um escritor, você sabe que a melhor forma de se aprender é na prática e quer melhor maneira de praticar em algo real?

### Ensinar e aprender

A comunidade open source exatamente por ser tão aberta e livre, permite que exista essa troca. Muitas coisas você pode aprender com outros projetos e também pode retornar com o que souber =)

### Criar uma boa imagem e ajudar na carreira

Pense no open source como seu grande portfólio. E se você tem projetos interessantes e que pessoas se interessam, isso pode atrair headhunters e outros devs a se espelhar em você. Mas lembre-se, você não precisa criar o "novo React". Qualquer contribuição é válida e sempre pode ajudar alguém que está começando, não tenha medo de contribuir, por mais que ache que "é algo bobo".

## Como contribuir?

A primeira e mais importante regra é:

Existem várias formas de contribuir e fazer código é só UMA delas.

É muito comum o pessoal pensar que para contribuir em um projeto open source é possível só para gênios com QI absurdamente alto e consomem café para sobreviver. E bom pessoal, essa não é a verdade. Existe lugar no open source para todo mundo! Seja um sênior que já programa há 20 anos, seja você que aprendeu Git semana passada e tá dando seus primeiros passos com programação. E por que não alguém que sequer é programador?

Seguem abaixo formas de você contribuir com o Open Source e também com a comunidade ao redor:

### Você gosta de design?

* Crie sites para projetos open source! Deixe eles sexy! Muitos programadores não tem habilidade nenhuma com cores e formas e por mais que seus projetos sejam incríveis, as vezes ficam no limbo por não terem nada que crie atração ao primeiro momento.
* Ajude a melhorar a usabilidade e/ou acessibilidade de algum projeto já existente.
* Teste, futuque, procure por bugs e os reporte! Isso é extremamente importante para nós.

### Você gosta de planejar eventos?

Ué? Pensei que estávamos falando de código.

Como eu disse mais acima, open source não é somente código, hoje isso pode ser extrapolado e por que não considerar eventos como parte? Aliás, eles são importantes para fortalecer a comunidade, criar laços, compartilhar conhecimento e muitas coisas mais! E como ajudar nisso?

* Ajude a organizar eventos e/ou meetups na sua cidade.
* Ajude as pessoas a encontrarem eventos bons e interessantes para elas.

### Você gosta de escrever?

* Ajude na documentação de um projeto (isso é muito importante).
* Escreva tutoriais sobre o projeto.
* Faça traduções de documentações.

### Você gosta de organização?

* Mostre issues duplicadas, crie novas labels para issues, organize.
* Vá atrás de issues antigas e veja se elas ainda fazem sentido, senão peça para fechá-las.
* Crie discussões dentro dos projetos para que faça ele ir para frente.

### Você gosta de codar?

* Busque por issues para contribuir
* Pergunte se pode ajudar na criação de algo para o projeto
* Automatize o processo de instalação do projeto
* Melhore os testes ou ferramentas

### Você gosta de ajudar pessoas?

* Responda perguntas sobre o projeto no Stack Overflow, Reddit, Quora, etc.
* Responda as pessoas em issues abertas
* Ajude a moderar quadros de discussão ou canais de conversação (Slack, Gitter)
* Participe do [FrontendBr](http://github.com/frontendbr/forum) e do [BackendBr](http://github.com/backend-br/forum), ajude pessoas de todo o Brasil <3

### Você pode simplesmente ajudar em outras coisas que não sejam código!

* Você pode ajudar em listas ou até criando listas de links úteis para algum estudo. [Aqui](https://github.com/sindresorhus/awesome) tem uma lista imensa com várias listas que você pode começar a ajudar hoje mesmo. Eu tenho um carinho muito grande por esse tipo de coisa e inclusive tenho 2 listas, uma sobre [SVG](https://github.com/willianjusten/awesome-svg) e outra sobre [Audio Visualização](https://github.com/willianjusten/awesome-audio-visualization).
* Existe também um projeto da h5bp sobre [Perguntas comuns em Entrevistas](https://github.com/h5bp/Front-end-Developer-Interview-Questions)

E fique calmo, ainda vou falar como encontrar coisas para contribuir, isso acima foi só uma prévia xD

## Orientando-se para um novo projeto

Quando você está "iniciando" em algum projeto, é comum que os mantenedores do projeto não vejam com bons olhos se você já chegar reclamando e/ou apontando falhas sem lhes dar nenhuma solução ou até uma "apresentação". Então, antes que você pule no projeto as cegas, faça o seu dever de casa, leia sobre o projeto, veja as issues já reportadas e pull requests mergeados e negados.

### Anatomia básica de um projeto open source

Todo projeto open source é diferente, mas existem algumas coisas que ao passar dos tempos passaram a ganhar nomenclaturas e padrões. Tendo isso em mente, nós temos alguns papeís dentro do projeto:

* Author: a pessoa/organização que criou o projeto.
* Owner: pessoa que tem os direitos administrativos do repositório (nem sempre é o autor)
* Mantenedores: os responsáveis por guiar a visão e organização do projeto. (podem ser ou não os autores do projeto)
* Contribuintes: todo mundo que já contribuiu em alguma coisa no projeto.
* Membros da comunidade: pessoas que usam o projeto e que podem ser ativos em grupos de discussão.

Temos também a documentação, que tem seguido o seguinte padrão:

* LICENSE: por definição, todo projeto open source precisa ter uma licença. Se ele não possui uma licença, ele não é open source. E essa licença que vai guiar as permissões diante daquele projeto.
* README: é a documentação de "boas-vindas" do projeto, onde, em geral, explica um pouco sobre o projeto e como começar com ele.
* CONTRIBUTING: enquanto o README ajuda as pessoas a usarem o projeto, o contributing ajuda as pessoas a contribuirem com o projeto. Ele explica o processo para contribuição e outros detalhes caso necessário. Enquanto nem todo projeto possui um arquivo de contributing, os projetos que o possuem já sinalizam que contribuições são bem vindas e aceitas.
* CODEOFCONDUCT: o código de conduta serve para definir regras para o comportamento de todos os participantes, de forma a facilitar um ambiente amigável e acolhedor.

E por último, nós temos as ferramentas que servem para organizar as dicussões:

* Issue tracker: onde as pessoas discutem sobre bugs, dúvidas e/ou melhorias a serem feitas no projeto.
* Pull Requests: onde as pessoas discutem e revisam as mudanças que estão sendo feitas dentro do projeto.
* Fóruns de discussão / Chats: alguns projetos maiores tem chats como Slack/Gitter para que as discussões mais comuns possam ser feitas lá, como dúvidas de como usar o projeto ou erros na instalação.

## Encontrando um projeto para contribuir

Outra dúvida gigante é essa, tá legal, já sei o que é, já sei como funciona, como acho algo legal para contribuir? Antes de achar que você só pode contribuir naqueles projetos que você já usa ou conhece, lembre-se que o mundo open source é enorme e muita coisa legal você nunca nem viu e nem ouviu falar, mas pode contribuir!

Existem algumas formas bem legais de achar projetos para ajudar, que são:

### Buscas diretas no Github

O Github possui algumas labels muito conhecidas e úteis que a galera usa, exatamente para buscar ajuda, alguns exemplos:

* [is:issue is:open label:beginner](https://github.com/search?utf8=✓&q=is%3Aissue+is%3Aopen+label%3Abeginner)
* [is:issue is:open label:easy](https://github.com/search?utf8=✓&q=is%3Aissue+is%3Aopen+label%3Aeasy)
* [is:issue is:open label:first-timers-only](https://github.com/search?utf8=✓&q=is%3Aissue+is%3Aopen+label%3Afirst-timers-only)
* [is:issue is:open label:good-first-bug](https://github.com/search?utf8=✓&q=is%3Aissue+is%3Aopen+label%3Agood-first-bug)
* [is:issue is:open label:"good first issue"](https://github.com/search?utf8=✓&q=is%3Aissue+is%3Aopen+label%3A"good+first+issue")
* [is:issue is:open label:starter](https://github.com/search?utf8=✓&q=is%3Aissue+is%3Aopen+label%3Astarter)
* [is:issue is:open label:up-for-grabs](https://github.com/search?utf8=✓&q=is%3Aissue+is%3Aopen+label%3Aup-for-grabs)

### Ferramentas para buscar projetos

* [Up For Grabs](http://up-for-grabs.net/" \l "/) - uma lista de projetos com issues para iniciantes.
* [Issuehub.io](http://issuehub.io/) - uma ferramenta para buscar issues no Github por label e linguagem.
* [Code Triage](https://www.codetriage.com/) - outra ferramenta muito boa para procurar repositórios por issues e linguagem.
* [First Timers Only](http://www.firsttimersonly.com/) - uma lista de bugs para pessoas que nunca contribuiram antes.
* [YourFirstPR](https://twitter.com/yourfirstpr) - um twitter que reporta issues que podem ser resolvidos por iniciantes.
* [Github Explore](https://github.com/explore/) - uma ferramenta dentro do próprio Github para lhe apresentar projetos novos e que podem ser interessantes.

## Uma checklist para antes de contribuir

Quando você estiver procurando por um projeto para contribuir, também é importante notar alguns aspectos, caso contrário, talvez seu trabalho nunca seja respondido ou receba uma resposta não tão calorosa. Aqui uma listinha:

* O projeto possui uma licença? Ela é de acordo com o que você apoia?
* Quando foi o último commit do projeto? Faz muito tempo?
* Com que frequência esses commits são feitos?
* O projeto tem muitas issues abertas sem resposta?
* Os mantenedores respondem rapidamente as issues?
* Existem discussões nas issues?
* Tem alguma issue recente?
* Existem issues fechadas para o projeto?
* Verifique as mesmas coisas para os Pull Requests
* Quando respondidas, essas issues/PR são tratadas de forma amigável?
* As pessoas recebem reviews em suas contribuições?
* Os mantenedores agradecem as pessoas por suas contribuições?

## Como submeter uma contribuição?

Tão importante quanto contribuir, é a forma com que você comunica/contribui. Para que sua contribuição seja levada a sério, siga alguns passos como:

### Comunique efetivamente

A comunicação é, em muitas vezes, até mais importante que a sua contribuição. E é importante lembrar que estamos trabalhando com pessoas e não robôs, então tente ser claro e direto. Lembrando que "direto" não significa ser rude!

Antes de abrir uma issue ou um Pull Request, tenha alguns pontos em mente.

Dê contexto. Ajude os outros a entenderem rapidamente. Se você estiver com um erro, explique o que você está fazendo e como reproduzí-lo. Se você está sugerindo uma nova ideia, explique por que você acha que seria útil para o projeto (não apenas para você!).

* Bom: "X não acontece quando eu faço Y"
* Ruim: "X está quebrado! Por favor corrija!"

#### Faça seu dever de casa antes

É OK não saber todas as coisas, mas mostre que você pelo menos tentou. Antes de pedir por ajuda, esteja ciente de ter checado no README, documentação, issues (abertas e fechadas) e procure na internet pela resposta. As pessoas vão apreciar que você ao menos demonstrou estar tentando.

* Bom: Eu não sei como implementar X. Eu chequei na documentação e não achei nenhuma menção, existe outro lugar?
* Ruim: Como eu faço X?

#### Faça seus pedidos de forma curta e direta

Assim como um email, toda contribuição, não importa quão simples ou útil, ela vai precisar ser lida por alguém. E a maioria dos projetos tem mais pedidos do que pessoas disponíveis para ajudar. Então seja conciso:

* Bom: Eu gostaria de escrever um tutorial sobre a API.
* Ruim: Eu estava dirigindo na auto-estrada outro dia e então parei num posto de gasolina e tive uma grande ideia de algo que poderíamos fazer, mas antes de eu explicar o quê, deixe me mostrar...

#### Mantenha toda comunicação pública

As vezes é tentador, mas não saia tentando falar diretamente com os mantenedores de forma privada (Facebook, Direct Messages, Email). Quando você fala publicamente, mais pessoas podem aprender e se beneficiar dessa conversa.

* Bom: (as a comment on Github) "Olá @mantenedor! Como eu devo proceder com esse PR?"
* Ruim: (como um email) "Olá, desculpe te incomodar, mas teria como dar uma olhadinha no meu PR?"

#### É ok fazer perguntas (mas seja paciente!)

Todo mundo precisa de um tempo para pensar sobre a pergunta, as vezes nem mesmo o mantenedor sabe a resposta e vai precisar pesquisar. Ao invés de continuar perguntando e perguntando, tente e estude junto, se tiver alguma informação interessante, passe adiante.

* Bom: Obrigado por olhar esse erro. Eu segui suas instruções e esse aqui é o resultado.
* Ruim: Por que você não corrige meu problema? Você não é o dono do projeto?

#### Respeite as decisões do projeto

Sim, isso deveria ser básico, mas muita gente esquece... Já vi milhares de pessoas xingando porque o React Router fez isso, por que o Angular mudou completamente, bando de \*\*\*\*. Lembrem-se, por trás de todo projeto, existem pessoas assim como você. Você pode dar seu feedback, mas nunca seja agressivo.

* Bom: Eu estou triste que você não tem como me ajudar no meu problema. Mas como você explicou, é um caso muito raro e eu entendo o porquê. Obrigado por me ouvir.
* Ruim: Vocês destruiram o projeto inteiro, vou usar Y.

### Abrindo uma Issue

Você deve abrir uma issue nas seguintes situações:

* Reportar um erro que você não pode resolver
* Propor uma nova feature ou outra ideia de projeto

Dicas para se comunicar nas issues:

* Se você viu uma issue que você quer fazer, comente na issue, faça com que a pessoa saiba que você está de olho nisso. Dessa forma vai evitar trabalho duplicado.
* Se uma issue foi aberta há muito tempo, é possível que ela já tenha sido até solucionada por alguém, então pergunte para confirmar antes que comece a trabalhar.
* Se você abriu uma issue e depois descobriu a resposta sozinho, comente na issue para que as pessoas saibam, de preferência responda sua própria pergunta e então feche a issue.

### Abrindo um Pull Request

Você deve abrir um Pull Request nas seguintes situações:

* Submeter pequenas correções (um erro de digitação, um link quebrado ou um erro bem óbvio)
* Começar a trabalhar numa contribuição que já foi pedida e você discutiu antes em alguma issue

Um Pull Request não precisa representar um trabalho finalizado. É inclusive melhor abrir um PR cedo, assim outros podem olhar e dar feedbacks durante o progresso. Só não esqueça de marcar como WIP (Work in Progress).

Se o projeto estiver no Github, aqui algumas dicas úteis de como criar o PR:

* [Faça um fork do repositório](https://guides.github.com/activities/forking/) e o clone localmente. Conecte seu local com o repositório original (upstream) o adicionando como remoto. E frequentemente pegue as mudanças do "upstream", assim você garante estar sempre atualizado com o projeto original. [Saiba mais como manter o seu repositório atualizado com esse post do Daciuk](https://blog.da2k.com.br/2014/01/19/manter-repositorio-github-forkado-sincronizado-com-o-original/).
* Crie um branch para o seu trabalho.
* Referencie qualquer issue relevante ou qualquer documentação necessária no seu PR.
* Inclua screenshots de antes e depois caso suas mudanças afetem algo visual do projeto.
* Teste suas mudanças! Rode os testes do projeto se existirem, se não existirem pense em criar. E tendo testes ou não, se assegure que tudo está funcionando através de testes manuais.

## O que acontece depois?

Depois que você submete sua contribuição, algumas coisas podem acontecer, que são:

### Você não recebe uma resposta

Felizmente você fez a listinha antes de contribuir para o projeto e viu que o projeto tinha sinais de atividade. Mas, as vezes, mesmo em projetos ativos, é possível você não receber nenhuma resposta.

Se você não recebeu nenhuma resposta mesmo depois de uma semana, é aceitável que dentro desse mesmo PR, você faça uma pergunta por alguém para revisar, lembrando de seguir as normas já faladas acima.

Se mesmo depois de pedir, ninguém responder, é possível que você nem receba mais respostas. Isso é bastante chato, mas não deixe que isso desencorage você. Acontece com todo mundo. Existem vários motivos para que você não tenha recebido uma resposta, e bom, eles estão fora do seu alcance. A dica é, tente procurar outro projeto ou forma de contribuir.

### Respondem e pedem mudanças

É comum que você receba alguns pedidos de mudança, seja pelo código não ser feito da forma que gostariam ou algo que não esteja de acordo com o projeto/escopo.

Quando alguém pedir por mudanças, responda. Eles tomaram seus tempos para revisar sua contribuição. Abrir um PR e simplesmente sumir não é algo legal. Se você não souber como fazer as mudanças, pesquise sobre, tente resolver e se não conseguir, peça por ajuda se precisar.

Se você não tem mais tempo para trabalhar no problema mais (por exemplo, a discussão está acontecendo durante meses e agora as circunstâncias mudaram), faça com que o mantenedor saiba, assim outra pessoa pode ajudar.

### Sua contribuição não foi aceita

Sua contribuição pode ou não ser aceita, isso vai depender muito dos mantenedores do projeto e o que eles acham que pode ser bom ou não para o projeto. E não se preocupe, isso acontece também. Você pode perguntar educadamente as razões para não ser aceito, porém, de forma alguma discuta ou seja hostil. Lembre-se, somos todos humanos e precisamos de educação.

### Sua contribuição foi aceita

Yeyyyy! Uma sensação incrível né? Corre já para contribuir em outra coisa! o/

## Mais alguns artigos que podem lhe ajudar

Wow! Eu sei, tem muita coisa para ser lida aqui, mas nunca é demais para ler mais coisas né? Seguem agora alguns links que podem ser interessantes para você:

* [Como escolher (e contribuir) no seu primeiro projeto open source (en)](https://github.com/collections/choosing-projects)
* [Como achar seu primeiro bug para corrigir (en)](https://medium.freecodecamp.org/finding-your-first-open-source-project-or-bug-to-work-on-1712f651e5ba" \l ".slc8i2h1l)
* [Traga bondade para o Open Source (en)](http://www.hanselman.com/blog/BringKindnessBackToOpenSource.aspx)
* [Como contribuir com Open Source](https://opensource.guide/how-to-contribute/)
* [Como achar um bug no seu código](https://8thlight.com/blog/doug-bradbury/2016/06/29/how-to-find-bug-in-your-code.html)
* [Ficando mestre em Markdown](https://guides.github.com/features/mastering-markdown/)

## Conclusão

É isso aí galera, espero que esse guia ajude um pouco vocês e que saiam várias grandes contribuições daqui! A nossa área é uma das mais legais no mundo, nós podemos ajudar e ser ajudados por pessoas que nunca vimos no mundo real, isso num é incrível?! Eu deixo aqui em separado de novo, o [Fórum do FrontendBr](https://github.com/frontendbr/forum), que é todo feito dentro do Github, ou seja, cada pergunta que você faz ou que ajuda a responder, você já está contribuindo diretamente para o Open Source!

E se você nunca usou o Git ou Github na vida, fica aqui o link do meu curso [Git e Github para Iniciantes](https://www.udemy.com/git-e-github-para-iniciantes/) que é completamente gratuito e já vai te ajudar a iniciar nesse mundo! E se você já sabe Git, mas quer melhorar ainda mais (ou se quiser agradecer pelo guia), tem também meu curso [Git e Github na Vida Real](https://www.udemy.com/git-e-github-na-vida-real/?couponCode=PROMOJUL21). E se você quiser colaborar com correções nesse guia (deve ter um monte de erro de português/digitação), é só ir no [repositório do blog](https://github.com/willianjusten/willianjusten.com.br/tree/master/posts), que é todo em Markdown e você pode colaborar hoje e agora!

https://willianjusten.com.br/guia-como-contribuir-em-open-source/