

Projeto alvo escolhido: Firefox

O *git mirror* atual e oficial da base de códigos do Firefox (também conhecida como "gecko" ou "mozilla-central") pode ser encontrada em <https://github.com/mozilla/gecko-dev>. Este repositório contém todos os *branches* que compõem o dia-a-dia de nossos desenvolvedores, incluindo o *branch* mozilla-central (nightly), mozilla-aurora (edição de desenvolvedor), mozilla-beta e mozilla-release.

Se você pretende trabalhar com outros *branches* do Mozilla, <https://github.com/mozilla/gecko-projects> pode vir a calhar. Este repositório contém os *project branches* (também conhecidos como "twigs") que é de uso variado.

Se você preferir usar o *git* diretamente com os repositórios *hg*, este [workflow](#) ajudará você a fazer isso da melhor maneira possível. Dessa maneira nós também deixamos você dar *push* diretamente do seu repositório *git* para o *mercurial*.

Existe também um *etherpad* que busca descrever como você pode usar o *git* para interagir com o mozilla-central.

Como submeter uma contribuição

Você encontrou um projeto que gosta e está pronto para fazer uma contribuição. Finalmente! Veja como realizar sua contribuição de forma correta.

Se comunicando de forma eficaz

Seja você um colaborador ocasional ou esteja tentando entrar em uma comunidade, trabalhar com outras pessoas é uma das habilidades mais importantes que você desenvolverá no open source.

Antes de abrir uma issue, pull request ou fazer uma pergunta no bate-papo, ter os pontos a seguir em mente irá ajudar a transmitir sua ideia de maneira eficaz.

Contextualize Ajude os outros a entenderem rapidamente. Se você está patinando em um erro, explique o que você está tentando fazer e como reproduzir este erro.

Se você está sugerindo uma nova ideia, explique por que você acha que ela seria útil para o projeto (não apenas para você).

“Não acontece X quando eu faço Y”

“X está quebrado! Por favor conserte isso.”

Faça, antes, sua lição de casa. Não há problema em não saber as coisas, mas mostre que você tentou. Antes de pedir ajuda, verifique o README de um projeto, a documentação, as issues (abertas ou fechadas), a lista de e-mail e procure na internet por uma resposta. As pessoas vão gostar quando você demonstrar que está tentando aprender.

“Não sei como implementar o X. Verifiquei os documentos de ajuda e não encontrei nada mencionando.”

“Como eu faço X?”

Mantenha-se conciso e direto. Assim como enviar um e-mail, todas as contribuições, por mais simples ou úteis, exigem a análise de outra pessoa. Muitos projetos têm mais solicitações recebidas do que pessoas disponíveis para ajudar. Seja conciso. Você aumentará a chance de que alguém possa ajudá-lo.

“Eu gostaria de escrever um tutorial da API”

“Eu estava dirigindo pela estrada no outro dia e parei para abastecer, e então eu tive essa ideia incrível para algo que deveríamos fazer, mas antes de explicar isso, deixe-me mostrar-lhe ...”

Mantenha toda a comunicação pública. Embora seja tentador, não procure mantenedores de forma privada, a menos que você precise compartilhar informações confidenciais (como um problema de segurança ou violação grave de conduta). Quando você mantém a conversa pública, mais pessoas podem aprender e se beneficiar com a sua troca. As discussões podem ser, em si mesmas, contribuições.

(um comentário) “@-maintainer Olá! Como devemos proceder neste PR?”

(um email) “Ei, desculpe incomodá-lo por e-mail, mas eu queria saber se você teve a chance de rever o meu PR”

Não há problema em fazer perguntas (mas seja paciente!). Todo mundo já foi novo no projeto em algum momento, e até colaboradores experientes precisam se atualizar quando olham para um novo projeto. Da mesma forma, mantenedores de longa data nem sempre estão familiarizados com todas as partes do projeto. Mostre a mesma paciência que você gostaria que eles tivessem com você.

“Obrigado por pegar este erro. Eu segui sua sugestão. Aqui está a saída.”

“Porque você não pode resolver meu problema? Este projeto não é seu?”

Respeite a decisão da comunidade. Suas ideias podem ser diferentes das prioridades ou visão da comunidade. Eles podem oferecer feedback ou decidir não seguir sua ideia. Enquanto você deve discutir e procurar uma solução, os mantenedores têm que viver com sua decisão por mais tempo do que você. Se você não concordar com a decisão deles, você pode sempre trabalhar em sua própria cópia do projeto ou iniciar seu próprio projeto.

“Estou desapontado por não poder apoiar o meu caso de uso, mas como você explicou, isso afeta apenas uma pequena parte dos usuários, eu entendo o porquê. Obrigado por ouvir.”

“Porque você não dá suporte ao meu caso de uso? Isto é inaceitável!”

Acima de tudo, mantenha a classe. O open source é composto por contribuidores de todo o mundo. O contexto se perde em idiomas, culturas, regiões geográficas e fusos horários. Além disso, a comunicação escrita torna mais difícil transmitir um tom ou humor. Assuma boas intenções nessas conversas. É normal se desfazer de uma ideia de forma educada e pedir mais contexto ou esclarecer melhor sua posição. Apenas tente deixar a internet melhor do que quando você a encontrou.

Capturando o contexto

Antes de fazer qualquer coisa, faça uma verificação rápida para garantir que a sua ideia não tenha sido discutida em outro lugar. Explore o README do projeto, os problemas (abertos e fechados), a lista de e-mails e o Stack Overflow. Você não tem que gastar horas passando por tudo, mas uma busca rápida por alguns termos-chave pode lhe dar uma boa visão.

Se você não encontrar sua ideia em outro lugar, você está pronto para fazer uma contribuição. Se o projeto estiver no GitHub, você provavelmente se comunicará abrindo uma issue ou pull request:

- Issues são como o início de uma conversa ou discussão
- Pull requests são para início dos trabalhos em uma solução
- Para comunicações leves, para esclarecimentos ou questões de como fazer, tente perguntar no Stack Overflow, IRC, Slack, ou outro canal de comunicação, se o projeto tiver.

Antes de abrir uma issue ou pull request, verifique os documentos de contribuição do projeto (geralmente um arquivo chamado CONTRIBUTING ou no README), para ver se é necessário incluir algo específico. Por exemplo, eles podem pedir que você siga um modelo ou exigir que você use testes.

Se você quiser fazer uma contribuição substancial, abra uma issue para perguntar antes de trabalhar nela. É útil olhar o projeto por um tempo (no GitHub, [você pode clicar em “Watch”](#) para ser notificado de todas as conversas) e conhecer os membros da comunidade, antes de realizar trabalhos que possam não ser aceitos.

Abrindo uma issue

Você deve normalmente abrir uma issue nas seguintes situações:

- Reportar um erro que você não pode resolver por conta própria.
- Discutir um tópico de alto nível ou ideia (por exemplo, comunidade, visão ou políticas)
- Propor uma nova função ou outra ideia do projeto.

Dicas para se comunicar sobre problemas:

- Se você vir uma issue aberta que deseja resolver, comente na issue para que as pessoas saibam que você está interessado nela. Dessa forma, as pessoas estarão menos propensas a duplicar seu trabalho.
- Se uma issue foi aberta há algum tempo, É possível que ela esteja sendo resolvida em algum outro lugar ou já tenha sido resolvida, por isso, comente para pedir confirmação antes de iniciar o trabalho.
- Se você abriu uma issue, mas descobriu a resposta mais tarde, comente na issue para que as pessoas saibam, então feche a issue. Mesmo este registro serve como documentação para o projeto.

Abrindo um pull request

Você deve normalmente abrir um pull request nas seguintes situações:

- Envio de correções triviais (por exemplo, um erro de digitação, um link quebrado ou um erro óbvio)
- Iniciar o trabalho em uma contribuição que já foi discutida, ou que você já tenha discutido em uma issue

Um pull request não precisa representar o trabalho final. Geralmente, é melhor abrir um pull request no início, para que outras pessoas possam acompanhar ou dar feedback sobre seu progresso. Basta marcá-lo com um “WIP” (Work in Progress) na linha de assunto. Você sempre pode adicionar mais commits depois.

Se o projeto estiver no GitHub, veja como enviar um pull request:

- [Faça um fork do repositório](#) e clone localmente. Conecte seu repositório local com o “upstream” adicionando-o como repositório remoto. Baixe as alterações do “upstream” com frequência para que você fique atualizado, quando enviar seu pull request, os conflitos de mesclagem serão menos prováveis. (Veja instruções mais detalhadas [aqui](#).)
- [Crie um branch](#) para suas edições.
- Referencie qualquer issue relevante ou documentação de suporte em seu PR (por exemplo, “Closes #37.”)
- Inclua imagens do antes e depois se suas mudanças incluírem diferenças no HTML/CSS. Copie e cole as imagens na mensagem do seu pull request.
- Teste suas mudanças! Execute suas alterações em relação a quaisquer testes existentes, e crie novos quando necessário. Se os testes existirem sempre verifique se suas alterações não quebram o projeto existente.
- Contribua para o estilo do projeto para melhorar suas habilidades. Isso pode significar usar recuos, ponto-e-vírgula ou comentários de maneira diferente do que você faria em seu próprio repositório, mas torna mais fácil para o mantenedor unir códigos, e para outros manter, entender e melhorar no futuro.

O que acontece depois que você envia uma contribuição

Você fez isso! Parabéns por se tornar um contribuidor open source. Esperamos que seja a primeira de muitas.

Depois de enviar uma contribuição, uma das seguintes situações ocorrerá:

Você não recebe uma resposta.

Espero que você [tenha checado o projeto em busca de sinais de atividade](#) antes de fazer uma contribuição. Mesmo em um projeto ativo, no entanto, é possível que sua contribuição não receba uma resposta.

Se você não obtiver uma resposta após uma semana, é justo responder educadamente no mesmo tópico, pedindo a revisão de alguém. Se você souber o nome da pessoa certa para revisar sua contribuição, você poderá @-mencioná-la nesse tópico.

Não contate essa pessoa em particular; Lembre-se de que a comunicação pública é vital para projetos open source.

Se você fizer uma chamada educada e ninguém responder, é possível que ninguém responda, nunca. Não é um sentimento agradável, mas não deixe que isso o desanime. Aconteceu com todos! Existem muitas razões possíveis pelas quais você não recebeu uma resposta, incluindo circunstâncias pessoais que podem estar fora de seu controle. Tente encontrar outro projeto ou uma maneira de contribuir. Bem como outros motivos, esta é uma boa razão para não investir muito tempo em fazer uma contribuição antes que outros membros da comunidade estejam engajados e responsivos.

Alguém solicita alterações na sua contribuição.

É comum que você seja solicitado a fazer alterações em sua contribuição, ou mesmo feedback sobre o escopo de sua ideia ou alterações em seu código.

Quando alguém solicitar alterações, seja responsivo. Eles levaram algum tempo para revisar sua contribuição. Abrir um PR e ir embora é má educação. Se você não souber como fazer as alterações, pesquise o problema e peça ajuda se precisar.

Se você não tiver mais tempo para trabalhar no assunto (por exemplo, se a conversa já dura meses, e suas circunstâncias mudaram), informe ao mantenedor para que ele não esteja esperando uma resposta. Alguém pode ficar feliz em assumir o controle.

Sua contribuição não é aceita.

Sua contribuição pode ou não ser aceita no final. Espero que você não tenha trabalhado muito nisso até então. Se você não tem certeza do motivo pelo qual não foi aceito, é perfeitamente razoável pedir feedback e esclarecimentos ao mantenedor. Em última análise, no entanto, você precisa respeitar que essa é a

decisão deles. Não discuta ou seja hostil. Você é sempre bem vindo para fazer uma cópia e trabalhar em sua própria versão, se você não concordar!

Sua contribuição é aceita.

Viva! Você fez uma contribuição open source com sucesso!