Git e Github são utilizados no dia a dia das pessoas que criam software por um motivo bem simples: ter uma forma fácil de gerenciar o código fonte da aplicação, do sistema, do produto.

Em um time pequeno, algumas pessoas ainda tentam tomar conta desses arquivos de maneiras um tanto questionáveis: compartilhar diretórios na rede, utilizar ferramentas como Dropbox, ou manter tudo em um servidor de FTP. Práticas totalmente desaconselhadas.

**O que é Git? um Sistema de Controle de Versão**

Apenas poder acessar o código dos outros colaboradores não é suficiente. Precisamos manter o **histórico** dos nossos arquivos. Mais: das nossas modificações, pois muitas vezes mudamos arquivos em grupo, num movimento único (um **commit**). Dessa forma, podemos voltar atrás e recuperar o estado do sistema como ele era ontem, ou no ano passado, e comparar as mudanças, para encontrar bug, para estudar otimizações.

Todos nossos arquivos, assim como seus históricos, ficam em um **repositório** e existiam vários sistemas que gerenciavam repositórios assim, como CVS e SVN. O Git é uma alternativa que tem um funcionamento mais interessante ainda: ele é distribuído e todo mundo tem uma cópia inteira do repositório, não apenas o "servidor principal". Isso trás grandes vantagens que você não precisa aprender nesse exato momento.

**O Git é um sistema de controle de versão distribuído e amplamente adotado**. O Git nasceu e foi tomando espaço dos outros sistemas de controle. Seu criador principal é o mesmo que o do Linux: Linus Torvalds, e ganhou o coração das pessoas que trabalham com open source.

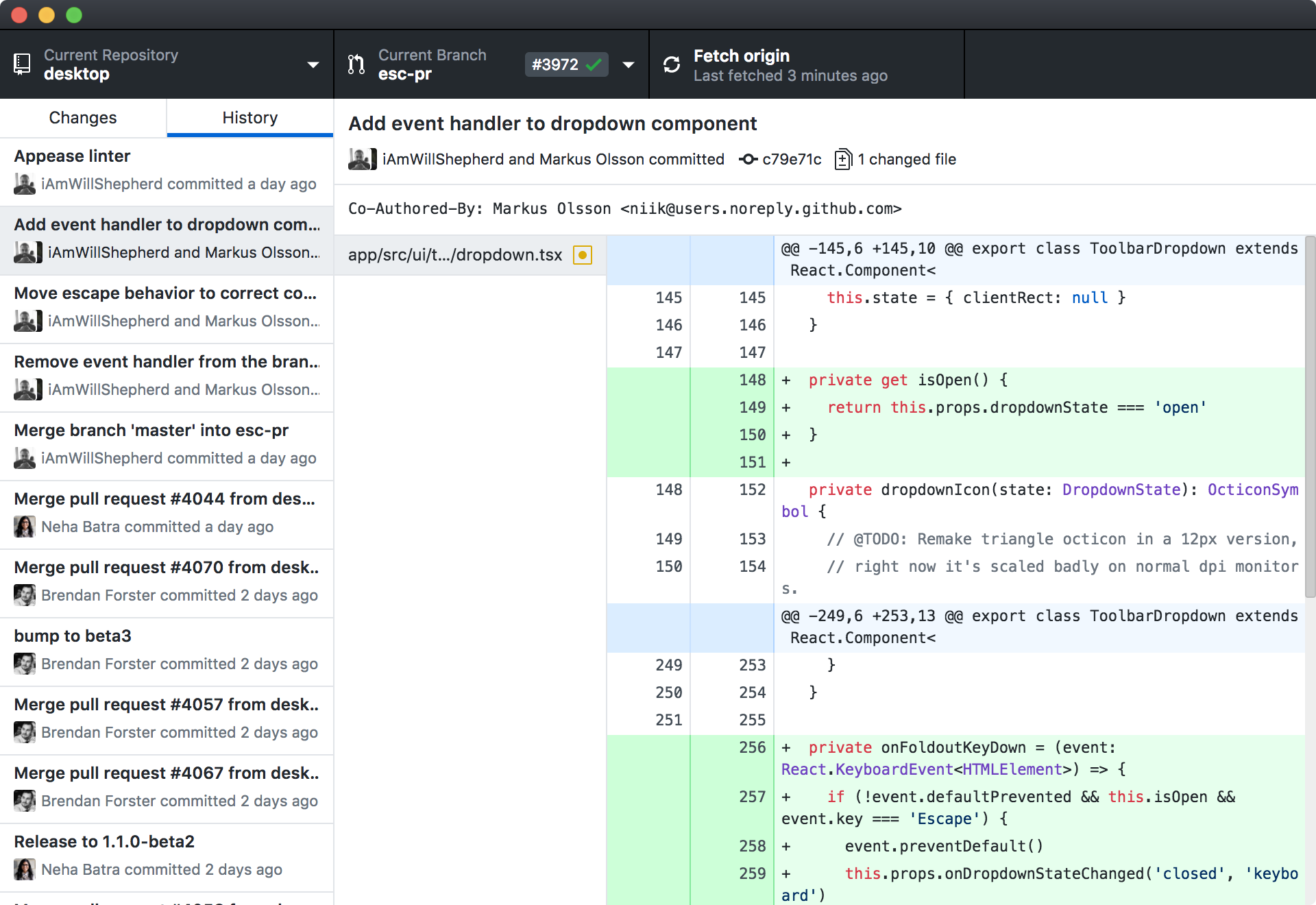
**Github: usar Git sem precisar configurar e manter um servidor**

O [Github](https://github.com) tem sim muita relação com o Git. **GitHub é uma plataforma para gerenciar seu código e criar um ambiente de colaboração entre devs, utilizando o Git como sistema de controle**. Ele vai facilitar o uso do Git, escondendo alguns detalhes mais complicados de setup. É lá que você provavelmente vai ter seu repositório e usar no dia a dia.

O sistema web que ele possui permite que você altere arquivos lá mesmo, apesar de não ser muito aconselhado, pois você não terá um editor, um ambiente de desenvolvimento e testes. Para se comunicar com o github e mexer nos arquivos do seu reposítorio, você pode usar a linha de comando, usando o comando do git e suas diretivas de commit, pull e push. Parece assustador? Há uma alternativa: usar um aplicativo desktop mais intuitivo:

**Ponto inicial: Github Desktop**

O [Github Desktop](https://desktop.github.com/) é uma casca para o comando do git, que esconde alguns termos que podem ser assustadores para quem está começando. Eu indico fortemente seu uso como primeiro passo, pois o sistema inteiro dá uma cara de "sincronizador de código". Facilita a visualizações, o envio e recebimento das modificações e os famosos conflitos de merge, que você não precisa se preocupar nesse primeiro instante.



Nesse vídeo o Felipe, da Alura, te explica como dar os passos no Github Desktop:

**Git e Github para sobrevivência**

Nessa nova websérie da Alura, o Mario Souto, conhecido como DevSoutinho, traz importantes pontos do uso dessa plataforma:

O segundo episódio também está no ar: **Como funciona o merge**? Aqui você vai entender os branches, a master e como juntamos o trabalho de diversas pessoas e equipes:

**Mais referências e conteúdo**

O Fabio Akita tem um canal que admiro muito e explica o Git e sua importância sem ser um tutorial:

Se você quer ouvir um guia para iniciantes em Git e Github, eu gravei esse episódio justo para facilitar esses passos:

Temos um [artigo tutorial](https://www.alura.com.br/artigos/comecando-com-git-aprendendo-versionar) para você fazer seus primeiros commits

E nosso [principal curso de Git e Github](https://www.alura.com.br/curso-online-git-github-controle-de-versao) é extremamente elogiado.

E a partir daí você pode entrar em merges e branches. Enquanto isso não se preocupe, fazer os primeiros pushes e pulls, sincronizando com o Github Desktop, é um excelente caminho.

**Perguntas Frequentes:**

**Como usar o Git?**

Você vai usar o Git para guardar o versionamento de todo seu sistema, de maneira segura e distribuída. Deve sempre fazer pequenos commits e 'pushar' suas modificações em grupos que fazem sentido. Do outro lado, deve fazer os 'pulls' para sincronizar com as modificações de todo time. O Git é uma ferramenta, você vai precisar de um servidor principal (mesmo ele sendo distribuído) para facilitar seu trabalho. É aí que entra o Github como sendo uma das opções já prontas.

**O que é um git commit??**

Cada commit registra um momento de vida do seu projeto. É uma foto (snapshot) daquele instante dos arquivos. Com isso você consegue se referenciar àquele momento do projeto, podendo voltar a ele, analisá-lo, ver as diferenças com um outro momento. Não se esqueça que os commits são feitos no seu repositório local e você precisa 'pushar' (empurrá-los) de volta para a origem, para que outras pessoas tenham acesso a essas fotos e atualizem-se no tempo de vida.

**Qual a diferença entre o Git é o GitHub??**

Git é a ferramenta que Linus criou. Se você quer utilizá-la, pode fazer tudo sozinho e hospedar um repositório principal para ter seu software versionado. Isso pode ser uma tarefa complicada. Algumas empresas, como Github, Bitbucket e Gitlab, já oferecem esse serviço, controlam o acesso dos usuários e oferecem ferramentas extras de gerenciamento de projetos. O Github é uma opção para que você use o Git de maneira mais simples, mas não é obrigatório, apesar de extremamente usado.

**Quais são as alternativas ao Github?**

O Bitbucket e o Gitlab são as duas principais alternativas de serviço Git além do Github.



Paulo Silveira

Paulo Silveira é CEO e cofundador da Alura. Bacharel e mestre em Ciência da Computação pela USP, teve sua carreira de formação em PHP, Java e nas maratonas de programação. Criou o Guj.com.br, o podcast do Hipsters.tech e o Like a Boss.