

BASE DE DADOS

LUKEBA ANTÓNIO MANUEL JOSÉ

O Que é GitHub, Para Que Serve e Como Usar

O GitHub é considerado é uma ferramenta essencial para engenheiros de software, com uma popularidade sem igual. Atualmente, ele acomoda mais de 25 milhões de usuários. Isso significa que há um número considerável de profissionais que estão procurando o GitHub para melhorar o fluxo de trabalho e a colaboração.

Em suma, o GitHub é um serviço baseado em nuvem que hospeda um sistema de controle de versão (VCS) chamado Git. Ele permite que os desenvolvedores colaborem e façam mudanças em projetos compartilhados enquanto mantêm um registro detalhado do seu progresso.

Para melhor entender o que é o GitHub e como ele funciona, precisamos dar uma olhada mais a fundo.

O que é um sistema de controle de versão?

Sempre que desenvolvedores criam um novo projeto eles continuam criando atualizações no código base. Mesmo depois de o projeto ser lançado é comum a atualização de versões, correção de bugs, adição de novas ferramentas, etc.

O sistema de controle de versão ajuda a acompanhar as mudanças feitas no código base. E mais, ele também registra quem efetuou a mudança e permite a restauração do código removido ou modificado.

Não há códigos sobrescritos uma vez que o Git salva múltiplas cópias no repositório. Se você ficou interessado no Git, aprenda mais aqui.

O Que é Git?

Agora que você sabe o que é GitHub, temos que entender que o Git é o coração do GitHub. “Git é um sistema de controle de versão” desenvolvido por Linus Torvalds (o criador do Linux).

Isso significa que qualquer desenvolvedor numa equipe pode gerenciar o código-fonte e seu histórico de mudanças usando ferramentas de linha de comandos de Git – desde que tenha sido concedido o acesso para isso, é claro.

Diferentemente dos sistemas de controle de versão centralizados, o Git oferece ramificações de recursos (ou *feature branches*). Isso significa que cada engenheiro de software na equipe pode separar uma ramificação de recursos que oferece um repositório local isolado para promover mudanças nos códigos.

Feature branches não afetam a ramificação principal, que é onde o código original do projeto está localizado. Uma vez que as mudanças tenham sido feitas e o código atualizado está pronto, a ramificação pode ser misturada (num processo de *merge*) com o *master branch*. É assim que as mudanças no projeto se tornam efetivas.

Por Que o GitHub é Tão Popular?

O GitHub hospeda mais de 100 milhões de repositórios, com a maior parte deles sendo projetos de código aberto. Essa estatística mostra que o GitHub está entre os clientes de Git GUI mais populares, e também porque é usado por vários profissionais e grandes empresas, como a Hostinger.

Isso acontece porque o GitHub é um projeto de gestão baseado em nuvem e uma plataforma de organização que incorpora os recursos de controle de versão do Git. Isso significa que todos os usuários do GitHub podem acompanhar e gerenciar as mudanças feitas para o código-fonte em tempo real, enquanto têm acesso a todos os outros recursos do Git disponíveis no mesmo lugar.

Além disso, a interface de usuário do GitHub é mais amigável do que a do Git, fazendo com que seja mais acessível para pessoas que possuem pouco ou nenhum conhecimento técnico. Isso significa mais membros de equipe podem ser incluídos no progresso e na gestão do projeto, fazendo com que o processo seja mais tranquilo.

Como Começar a Usar o GitHub

Você pode experimentar o GitHub com a sua equipe de graça. Existe um plano básico disponível que inclui repositórios e colaboradores ilimitados, mas oferece apenas 500 MB de espaço de armazenamento.

Para aproveitar melhor os muitos recursos do GitHub, você pode escolher um dos planos pagos que eles oferecem.

Assim que você escolher um plano, preencha os detalhes solicitados e finalize o processo de registro. A partir daí, você pode começar a explorar o que o GitHub tem a oferecer. Ao contrário do Git, o GitHub não vai exigir que você tenha conhecimentos de programação ou que use linhas de comando.

1. Crie um Repositório no GitHub

Repositório, ou *repo*, é um diretório onde os arquivos do seu projeto ficam armazenados. Ele pode ficar em um depósito do GitHub ou em seu computador. Você pode armazenar códigos, imagens, áudios, ou qualquer outra coisa relacionada ao projeto no diretório.

Para começar o processo, siga esses passos:

1. Clique em Create a repository (criar um repositório) para começar um novo projeto.
2. A seção Owner (proprietário) já vai ter o nome da sua conta. Crie um Nome de repositório. Confira se ele está definido para Public (público) para que ele tenha código aberto, e então marque a

caixa Add a README file (adicionar um arquivo LEIA-ME). Finalmente, clique em Criar repositório.

2. Crie Branches no GitHub

Ao criar *branches*, ou ramificações, você gera versões diferentes de um repositório. Quando você modifica o projeto nas *branches* de recursos, um desenvolvedor pode ver como isso vai afetar o projeto principal na hora que tudo for integrado.

Passos de como criar um *branch* de recursos:

1. Vá para o seu novo repositório. Clique no botão main (principal) e insira o nome do seu nome *branch* de recurso. Clique em Create branch (criar *branch*).

Agora você criou um novo *branch* de recurso que é idêntico ao *branch* mestre. Você pode começar a fazer mudanças livremente sem afetar o projeto principal.

3. Entenda Como Funcionam Commits no GitHub

Os Commits é como as mudanças salvas no GitHub são chamadas. Cada vez que você muda o arquivo do *branch* de recurso, você terá que executar um Commit para mantê-lo.

Aqui está como promover uma mudança e executar o Commit:

1. Acess o *branch* de recurso ao clicar em main e selecionar no menu suspenso a sua ramificação recém-criada.
2. Clique no “ícone de lápis” e comece a editar o arquivo. Assim que tiver terminado, escreva uma descrição curta sobre quais mudanças foram feitas. Clique em Commit Changes (confirmar alterações).

4. Crie Pull Requests no GitHub

Para propor as mudanças que você acabou de fazer para outros desenvolvedores trabalhando no mesmo projeto, você deve criar um pull request. São eles que fazem ser tão fácil de trabalhar junto em projetos, já que eles são a principal ferramenta de colaboração no GitHub.

Pull Requests permitem que você veja as diferenças entre o projeto original e o seu *branch* de recurso. É assim que você pede para os seus pares revisá-las. Se os outros desenvolvedores aprovarem as modificações, eles podem executar um merge pull request (solicitação de mesclagem). Isso irá aplicar as mudanças para o projeto principal.

Para fazer um *pull request*, siga os passos abaixo:

1. Clique em Pull requests -> New pull request (Solicitação de mesclagem -> Nova solicitação de mesclagem). Na seção Example comparisons (comparações de exemplo), selecione o *branch de recursos* no qual você está trabalhando há pouco.

2. Confira mais uma vez as mudanças e clique em Create pull request. Na nova página, escreva o título e ofereça uma curta descrição do trabalho que você acabou de executar — para encorajar que ele seja adicionado ao *branch* mestre. Clique em Create pull request.

Agora, os outros desenvolvedores poderão mesclar as mudanças que você fez com os arquivos originais do projeto. Se você quiser saber mais sobre como começar no GitHub, confira este [guia](#).

GitHub Não é Apenas Para Desenvolvedores

O GitHub é uma ótima plataforma que mudou o método de trabalho de desenvolvedores. Mas qualquer pessoa que deseja gerenciar seu projeto com eficiência e trabalhar com outros colaboradores também pode usar o GitHub.

Se sua equipe trabalha em um projeto que realiza atualizações constantes e você quer acompanhar como as mudanças são feitas, então o GitHub é uma ótima opção para você. Existem outras alternativas como o [GitLab](#) e [BitBucket](#), mas o GitHub deve ser levado em consideração.

Possui alguma experiência com o GitHub? Comente na seção abaixo.

O AUTOR:

Andrei L.

Jornalista e conteudista