Resumo do [Curso de Git e Github COMPLETO 2023 + Desafios + Muita Prática](https://youtube.com/watch?v=kB5e-gTAl_s)

*Este é um resumo gerado por IA. Pode haver imprecisões.*  
[Resumir outro vídeo](https://www.summarize.tech/) · [Compre summarize.tech Premium](https://www.summarize.tech/purchase)

[00:00:00](https://youtube.com/watch?v=kB5e-gTAl_s&t=0) - [00:45:00](https://youtube.com/watch?v=kB5e-gTAl_s&t=2700)

Este vídeo tutorial aborda vários aspectos do Git e do GitHub, começando com o processo de instalação e a criação de um repositório. O instrutor explica comandos como "git status", "git add" e "git commit" para controle de versão. Eles também demonstram como enviar código para um repositório do GitHub e criar ramificações para desenvolvimento separado. A importância de trabalhar em ramificações e mesclar alterações é enfatizada, juntamente com o uso de solicitações pull para revisões de código. O instrutor também explica o Git Ignore para excluir arquivos específicos do controle de versão. Ao longo do vídeo, o instrutor incentiva os espectadores a compartilhar o conteúdo e apoiar seu trabalho.

Visualizar menos

* 00:[**00:00**](https://youtube.com/watch?v=kB5e-gTAl_s&t=0) Nesta seção do vídeo, o instrutor apresenta o curso Git e GitHub, explicando como ele ajudará a transformar projetos em algo organizado e profissional. Eles enfatizam a importância do controle de versão e demonstram como o Git permite gerenciar diferentes versões de código, ajudando a evitar erros e colaborar efetivamente em uma equipe. O instrutor também menciona que o GitHub é uma plataforma popular para armazenar e compartilhar código. Eles incentivam os espectadores a compartilhar o vídeo e mostrar como instalar o Git em seus computadores.
* 00:[**05:00**](https://youtube.com/watch?v=kB5e-gTAl_s&t=300) Nesta seção, o vídeo tutorial aborda o processo de instalação do Git e do Github. O instrutor orienta os espectadores sobre como instalar o Git no computador e, em seguida, mostra como criar uma conta do Github. Depois que a conta estiver configurada, a próxima etapa é criar o primeiro repositório. O instrutor explica como inicializar um repositório usando o comando Git por meio da linha de comando. Em seguida, os visualizadores são ensinados a criar uma pasta de projeto e inicializá-la como um repositório. Além disso, o instrutor demonstra como criar um arquivo de texto com algumas linhas de código como exemplo. Ao final desta seção, os espectadores estão prontos para começar a trabalhar com o Git e o Github em seus projetos.
* [**00:10:00**](https://youtube.com/watch?v=kB5e-gTAl_s&t=600) Nesta seção, o instrutor explica o comando "git status", que é usado para verificar o status do seu repositório. Ao digitar "git status" no terminal, você receberá um relatório de quais arquivos foram incluídos no controle de versão e quais arquivos não foram. Além disso, ele mostrará se algum arquivo foi atualizado. O instrutor demonstra esse comando mostrando que ainda não há confirmações e que um arquivo não está incluído no controle de versão. Para adicionar esse arquivo ao controle de versão, o instrutor usa o comando "git add" seguido do nome do arquivo entre aspas. Depois de executar o comando, o arquivo é preparado para ser adicionado ao controle de versão. O instrutor também menciona que, se você tiver vários arquivos, poderá usar o comando "git add ." para adicionar todos os arquivos de uma só vez. Depois que os arquivos são adicionados, o próximo passo é criar a primeira versão do código usando o comando "git commit -m", seguido por uma mensagem que representa o que foi feito naquela versão. O instrutor também orienta sobre como configurar o e-mail e o nome de usuário para confirmações do git.
* [**00:15:00**](https://youtube.com/watch?v=kB5e-gTAl_s&t=900) Nesta seção, o instrutor explica que para enviar o código para o repositório em nuvem no GitHub, o comando "git push" é usado. No entanto, antes de fazer isso, é necessário definir para onde as alterações serão enviadas, o que é feito através da criação de um repositório no GitHub. O instrutor demonstra como criar um repositório e fornece opções como escolher entre acesso público ou privado, adicionar um arquivo Leiame e criar o repositório. Depois que o repositório é criado, o código pode ser enviado para a nuvem usando o comando "git push". O instrutor também menciona que diferentes versões ou ramificações podem ser criadas e, inicialmente, as alterações são enviadas para o ramo "Mestre". Por fim, o instrutor destaca que nas próximas aulas, comandos e conceitos avançados serão abordados para ajudar os alunos a trabalhar profissionalmente com o Git e o GitHub.
* [**00:20:00**](https://youtube.com/watch?v=kB5e-gTAl_s&t=1200) Nesta seção, o instrutor incentiva os espectadores a considerar fazer doações para apoiar seu trabalho se acharem o conteúdo valioso. Em seguida, ele passa a explicar como adicionar novas atualizações a um código e enviá-las ao servidor. O processo envolve modificar o código, adicionar as alterações ao controle de versão, criar uma nova versão, dar-lhe uma mensagem e enviar as atualizações. Ele demonstra como verificar o histórico de versões usando o comando "Git reflog" e mostra como reverter para uma versão anterior usando o comando "Git reset --hard" seguido pelo ID da versão desejada. Ele enfatiza a importância de ser capaz de navegar entre diferentes versões de código sem o risco de perder qualquer progresso.
* [**00:25:00**](https://youtube.com/watch?v=kB5e-gTAl_s&t=1500) Nesta seção, o instrutor alerta sobre a importância de não executar comandos de mudança de versão, a menos que necessário, e sugere consultar um superior ou analista antes de fazê-lo em um ambiente de equipe. Eles explicam o conceito de ramificações no Git, usando recursos visuais para ilustrar os diferentes caminhos do desenvolvimento. Eles destacam a ramificação Master como a ramificação principal para onde as alterações são enviadas, enquanto outras ramificações são usadas para desenvolvimento separado e posteriormente mescladas com a ramificação Master. O instrutor demonstra como criar uma nova ramificação usando a linha de comando Git Bash e explica a importância de nomear ramificações, como usar um nome que represente funcionalidade ou finalidade.
* [**00:30:00**](https://youtube.com/watch?v=kB5e-gTAl_s&t=1800) Nesta seção, o instrutor demonstra como usar o comando de checkout do Git para alternar para uma ramificação diferente a fim de trabalhar em modificações e novas funcionalidades. Eles mostram como criar uma nova ramificação e fazer alterações nessa ramificação sem afetar a ramificação atual. Eles também explicam como enviar as alterações da nova ramificação para a ramificação principal usando o comando push do Git. O instrutor enfatiza a importância de trabalhar em ramificações para evitar alterar o código estável e demonstra como mesclar alterações de uma nova ramificação para a ramificação principal usando o comando Git merge.
* [**00:35:00**](https://youtube.com/watch?v=kB5e-gTAl_s&t=2100) Nesta seção, o instrutor explica o processo de trabalhar com várias ramificações no Git. Eles demonstram como criar uma nova ramificação com base na ramificação mestre, adicionar novos recursos a essa ramificação e, em seguida, mesclar essas alterações de volta na ramificação mestre após o teste. Eles também enfatizam a importância de atualizar o código com as alterações mais recentes do servidor antes de mesclar ramificações para garantir que o código que está sendo mesclado esteja atualizado e não substitua nenhuma alteração recente feita por colegas. O instrutor fornece comandos específicos, como "git pull", "git checkout -b" e "git merge", para realizar essas tarefas de forma eficaz. No geral, esta seção oferece um guia prático passo a passo sobre o gerenciamento de filiais no Git.
* 00:[**40:00**](https://youtube.com/watch?v=kB5e-gTAl_s&t=2400) Nesta seção, o palestrante explica o processo de criação de um novo recurso e mesclá-lo com a ramificação principal usando Git e GitHub. Eles mencionam que esse processo é normalmente usado em projetos maiores ou colaborações em equipe. Eles também discutem a importância das solicitações pull, onde as alterações de código são revisadas e aprovadas por outros programadores. O palestrante demonstra o processo de criação de uma solicitação pull e explica que as alterações de código podem ser aprovadas ou rejeitadas com base nas avaliações. Eles observam que, nesse caso, como eles são a única pessoa na equipe, eles podem aprovar sua própria solicitação pull, mas em uma configuração de equipe real, várias pessoas precisariam revisar e aprovar as alterações de código.
* 00:[**45:00**](https://youtube.com/watch?v=kB5e-gTAl_s&t=2700) Nesta seção, o instrutor explica como mesclar códigos usando solicitações pull no Git e no GitHub. Eles demonstram o processo de confirmação e mesclagem de alterações de código de uma ramificação separada para a ramificação mestre. No entanto, em alguns casos, pode haver arquivos ou pastas que você não deseja incluir no controle de versão. Para excluir esses arquivos, você pode usar um processo chamado Git Ignore. O instrutor mostra como criar um arquivo Git Ignore e especifica quais arquivos ou pastas devem ser ignorados. Eles também destacam a importância de adicionar um arquivo Git Igignore, pois garante que arquivos específicos não sejam adicionados ao controle de versão. Por fim, o instrutor agradece o apoio dos espectadores e os incentiva a deixar feedbacks e doações para o curso.