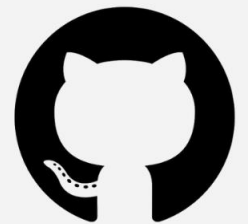


GIT E GITHUB

O ESSENCIAL

O QUE FAREMOS NESSE VÍDEO?

- Apresentaremos o git e github com intuito do aluno entender o processo de trabalho e os comandos básicos além de entender como resolver conflitos, criar branches e juntar as funcionalidades com o merge.



O QUE É GIT?

- Git é um sistema de controle de versão distribuído. Serve para controlar as versões de qualquer tipo de arquivo não necessariamente arquivos de sistemas.
- Com ele você pode manter versões diferentes de seus arquivos e projetos controlando assim seu histórico, sua evolução e podendo voltar sua versão caso algo dê muito errado!
- Com o git você pode ter outros colaboradores em seu projeto formando assim uma equipe de trabalho.



E O GITHUB?

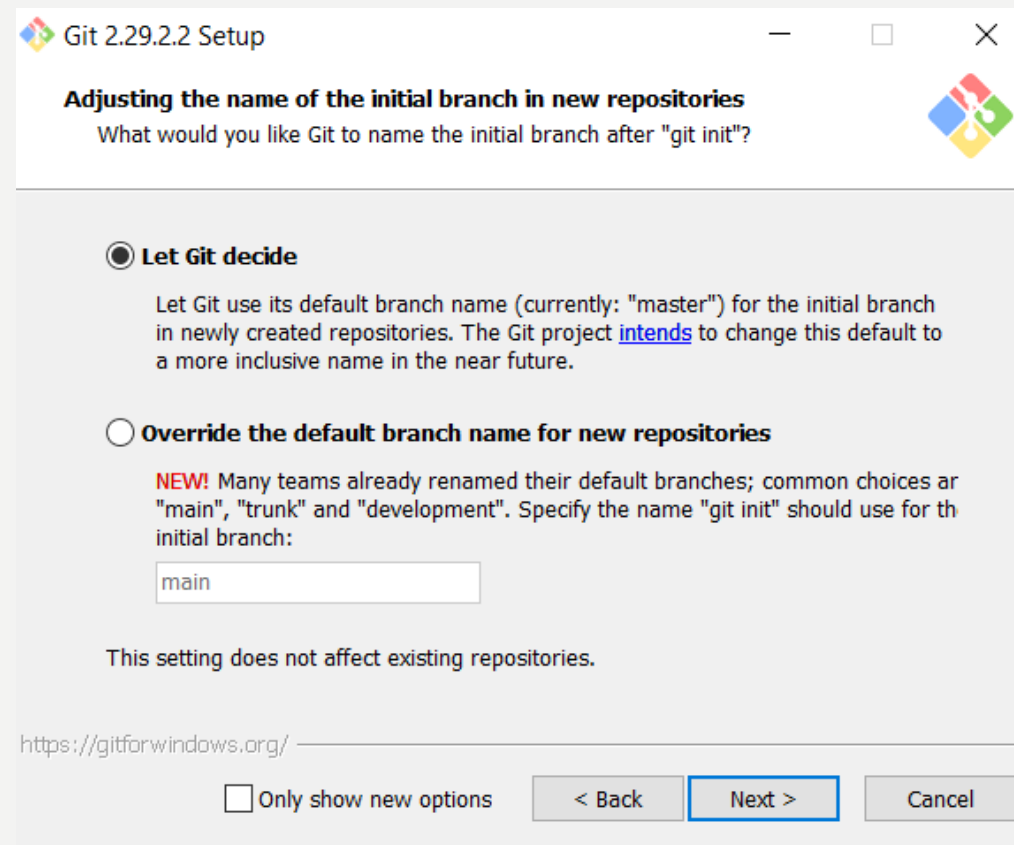
- GitHub é uma das plataformas existentes para projetos colaborativos. É como uma rede social de projetos distribuídos tendo como base de versionamento o próprio Git.
- Com o Github você pode mostrar seus projetos publicamente e pode receber colaboração ou colaborar em projetos open source ao redor do mundo. Um bom lugar para criar seu portfólio visando uma vaga futura, já que os recrutadores poderão ver seus projetos expostos publicamente.

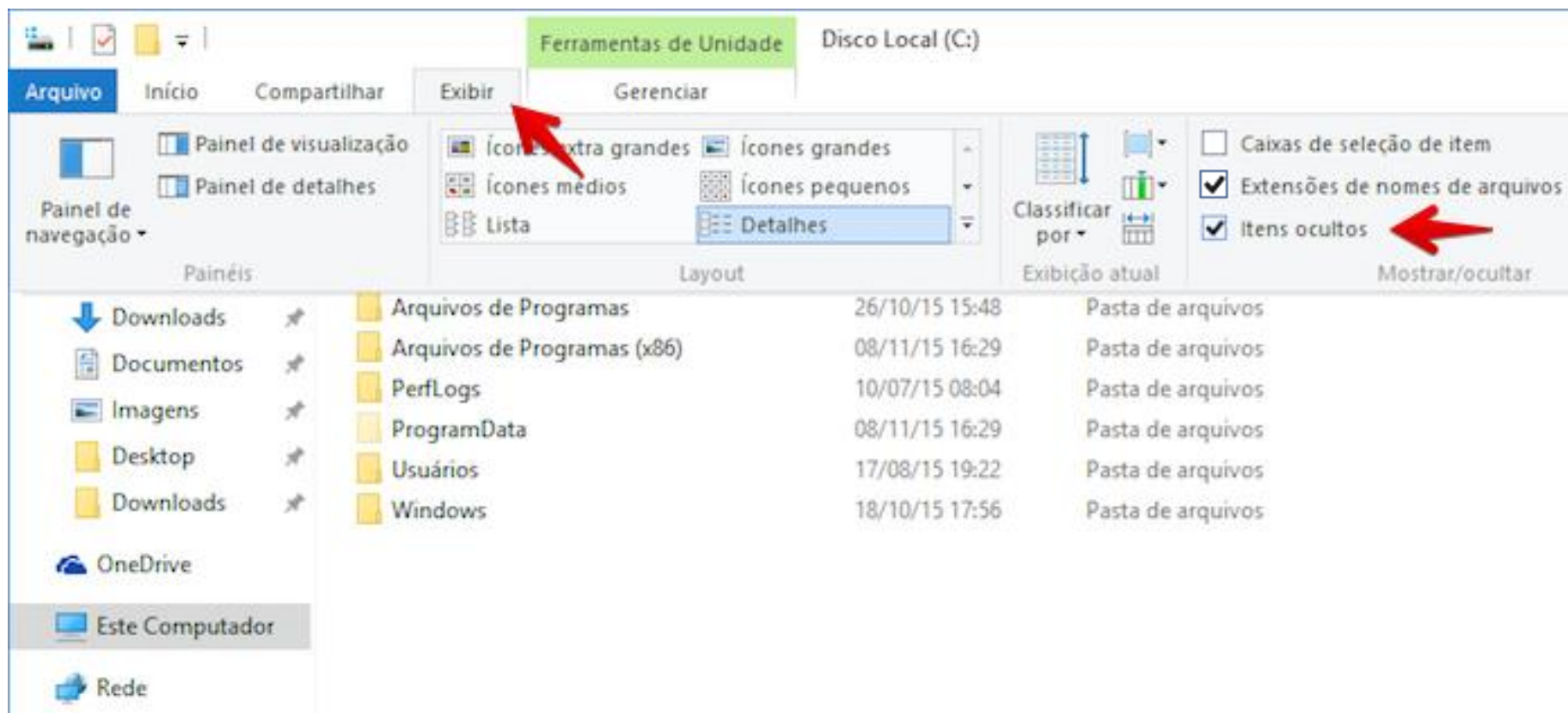


ONDE BAIXAR O GIT?

Pode ser baixado em <https://git-scm.com/>.

O instalador pode ser rodado em next, next, next como qualquer programa no Windows. Porém seria interessante utilizar a **branch main** na etapa **Adjusting the name of the initial branch in new repositories**





**DICA - EXIBA OS ARQUIVOS OCULTOS E AS
EXTENSÕES DOS ARQUIVOS!**

VOCÊ PODE EXIBIR OS ARQUIVOS OCULTOS PARA VISUALIZAR A PASTA .GIT DO SEU PROJETO

ATIVIDADE 1 – COMEÇANDO COM GIT

- Criar uma pasta de **Projeto Git**, pode ser na área de trabalho
- Iniciar o git na pasta do projeto (botão direito na pasta **Git Bash Here**)
- Mostrar os arquivos ocultos e as extensões dos arquivos
- Mostrar a diferença de quando se está em uma pasta com projeto git ou fora dela no git bash
- Criar arquivo de texto no bloco de notas
- Executar o comando *git status*

ESTÁGIOS PROCESSO GIT

WORKING DIRECTORY (DIRETÓRIO DE TRABALHO)

ARQUIVOS MODIFICADOS, EXCLUÍDOS OU ADICIONADOS

STAGING AREA (EM PREPARAÇÃO)

ARQUIVOS ADICIONADOS E PRONTOS PARA SEREM VERSIONADOS

.GIT DIRECTORY (REPOSITÓRIO)

ARQUIVOS SALVOS/VERSIONADOS



ATIVIDADE 1 – COMANDOS BÁSICOS

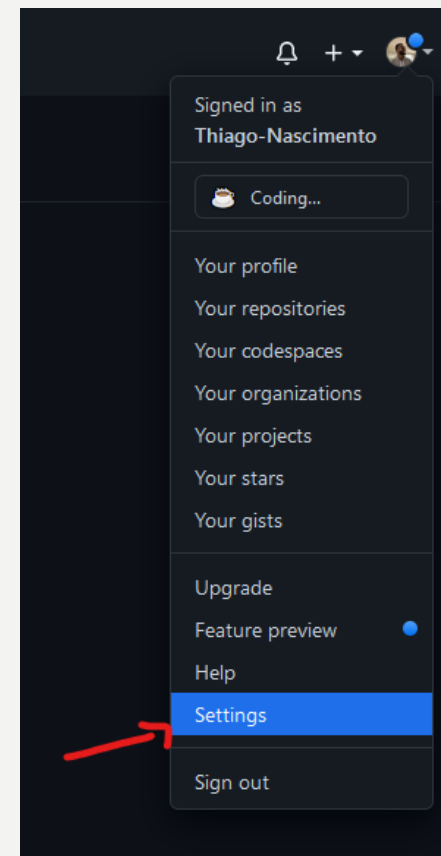
- *cd* – comando para entrar ou sair de pastas/diretórios
- *Clear* ou *cls* – limpa o console. Também existe o atalho **ctrl + I**
- *git status* – verificar o status dos arquivos
- *git add* – adicionar os arquivos alterados para o rastreamento do git
- *git commit* – adicionar os arquivos preparados para uma nova versão no git
- *git log* – histórico de versões dos arquivos
- *git branch* – criar ramos/novas funcionalidades no projeto
- *git checkout* – alterar o ramo de trabalho no projeto
- *git merge* – juntar funcionalidades
- *git clone* – clonar projetos externos
- *git remote* – repositório remoto
- *git config* – comando para configurações gerais do git. Nós veremos como alterar as configurações do usuário e email.

ATIVIDADE 2 – CRIAÇÃO DE PROJETO NO GITHUB.

- Criar conta no Github
- Criar novo Repositório
- Importar repositório existente usando os comandos do assistente do GitHub
- `git remote add origin https:// ...`
- `git branch -M main` - Cria a branch main e já altera para ela
- `git remote -v` - verbose opcional para ver se adicionou a origin corretamente
- `git push -u` - cria a branch caso não exista, só na primeira vez
- Exercitar a criação e alteração de arquivos e empurar/push para o github!
- Não esquecer de criar o arquivo Readme.md
- Postar a atividade na plataforma

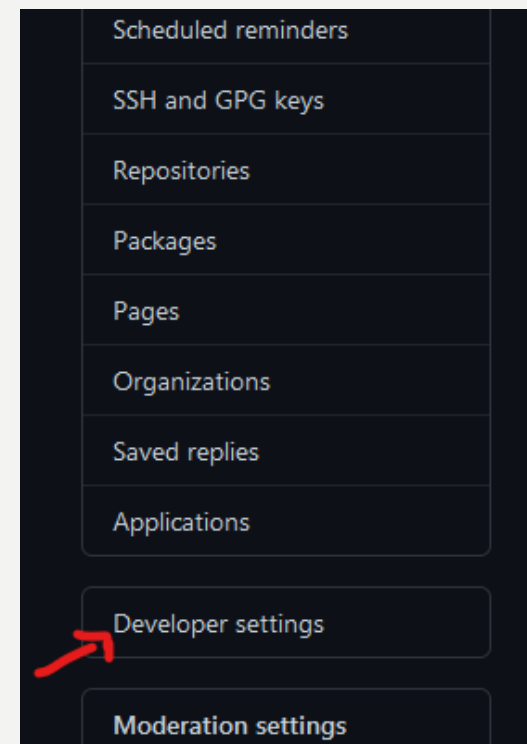
AUTENTICAÇÃO POR TOKENS GITHUB

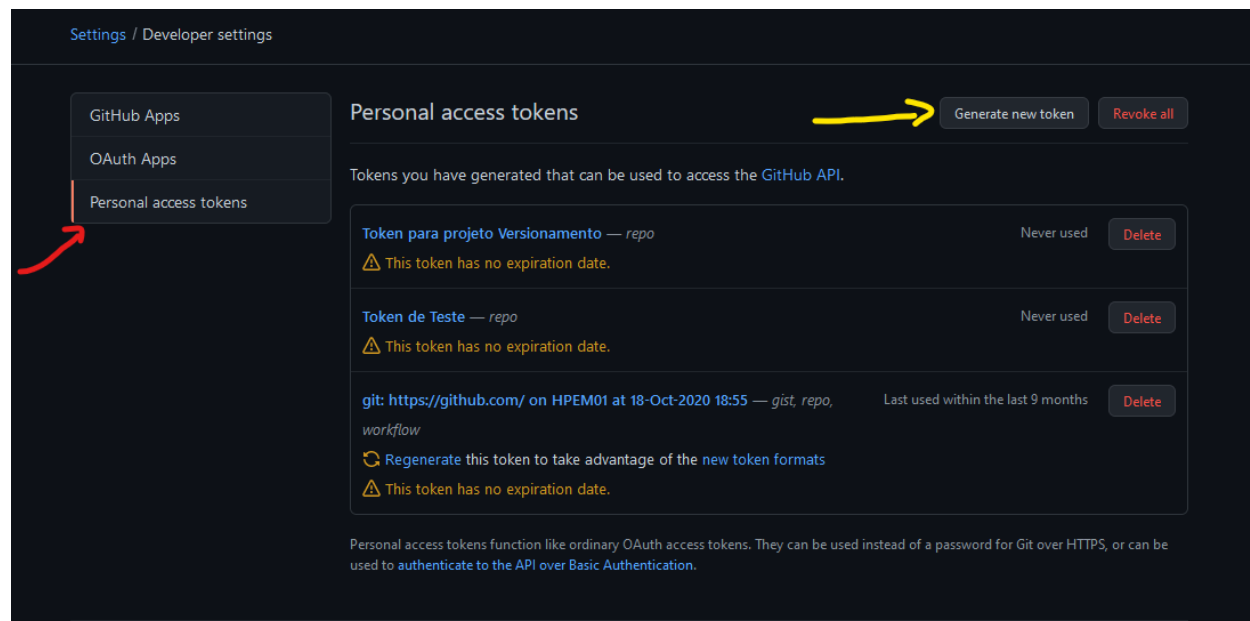
- Após a atualização do dia 14/08/2021, o Github alterou o método de autenticação para upload de alterações nos repositórios (git push), passando ser necessário a criação e utilização de um Personal Access Token (PAT) para as operações com o Github sobre o protocolo HTTPS.
- Para criar um Personal Access Token, precisamos acessar a opção Settings, pra isso clique na sua foto de perfil no lado direito do cabeçalho da página.



AUTENTICAÇÃO POR TOKENS GITHUB

- Nessa tela, selecione a opção de Personal access tokens, e depois clique no botão “Generate new token”, para iniciar a geração desse novo token de acesso pessoal.





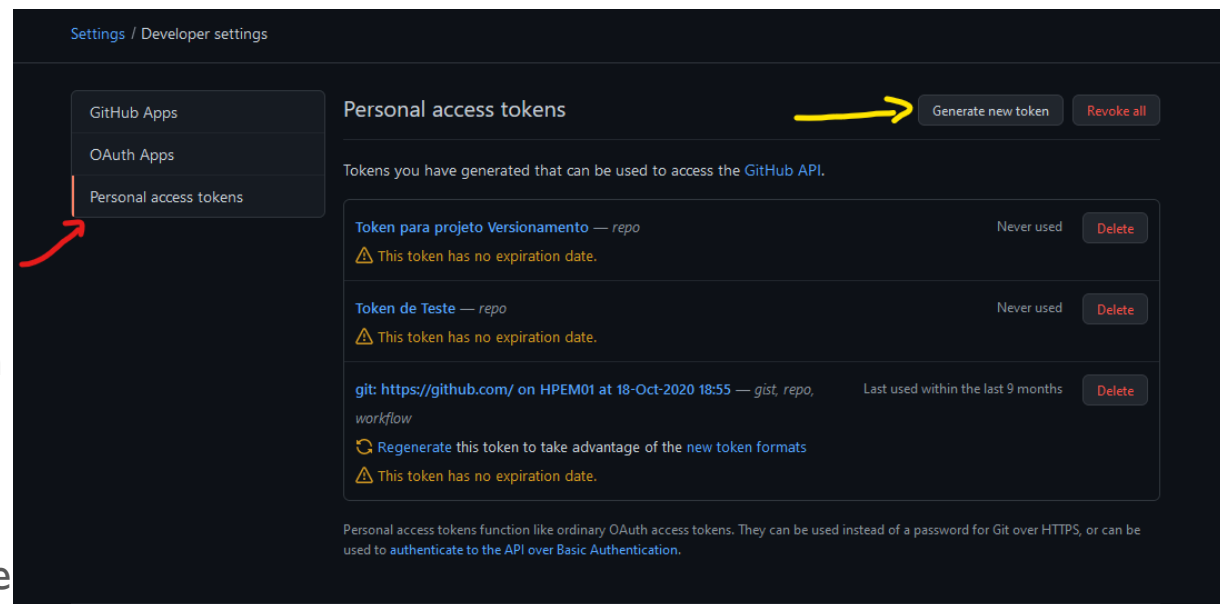
AUTENTICAÇÃO POR TOKENS GITHUB

Após acessar as configurações do perfil, clique na opção “Developer Settings”, localizada na aba de opções na lateral esquerda.

Nessa tela, selecione a opção de Personal access tokens, e depois clique no botão “Generate new token”, para iniciar a geração desse novo token de acesso pessoal.

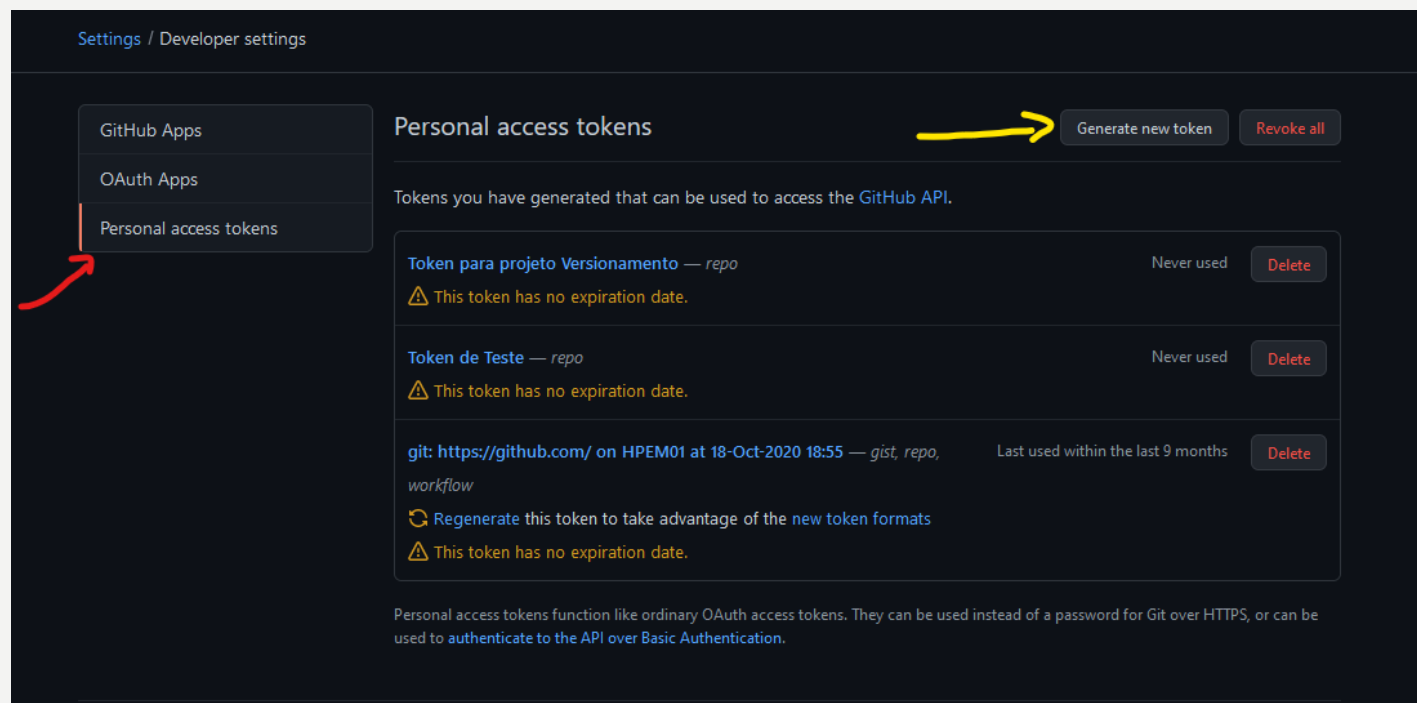
AUTENTICAÇÃO POR TOKENS GITHUB

- Nessa tela você terá as opções de configuração do token, incluindo uma breve descrição desse token (Note), o prazo de validade desse token (Expiration) e também as operações que são permitidas ao usar esse token (Scopes).
- No nosso exemplo vamos criar um token sem prazo de validade, que fará com que ele não expire, e o escopo selecionaremos a opção “repo”, que já será suficiente para as operações básicas nos repositórios.



AUTENTICAÇÃO POR TOKENS GITHUB

- Nessa tela você terá as opções de configuração do token, incluindo uma breve descrição desse token (Note), o prazo de validade desse token (Expiration) e também as operações que são permitidas ao usar esse token (Scopes).
- No nosso exemplo vamos criar um token sem prazo de validade, que fará com que ele não expire, e o escopo selecionaremos a opção “repo”, que já será suficiente para as operações básicas nos repositórios.



AUTENTICAÇÃO POR TOKENS GITHUB

- Após selecionar as configurações, clique no botão “Generate token” no final da página.

-

- Depois de gerar o token, ele será exibido na tela, copie o token e guarde-o como se fosse uma senha, depois de sair dessa tela o token não será mais mostrado, por isso é importante copia-lo.

-

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

Descrição do token

What's this token for?

Expiration *

No expiration ⌵ The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

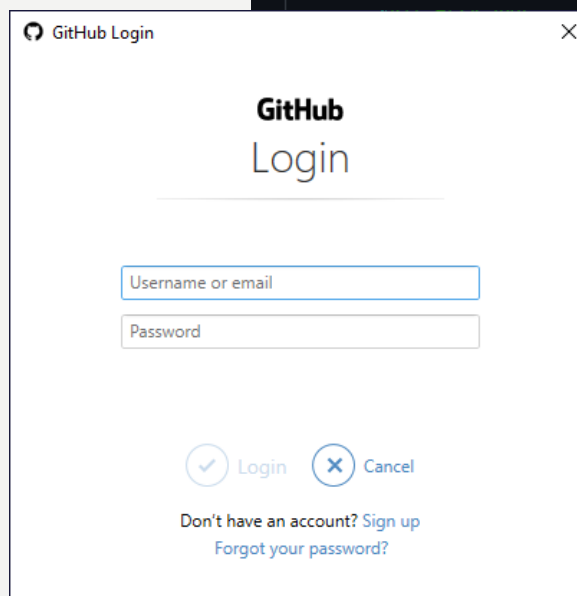
Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

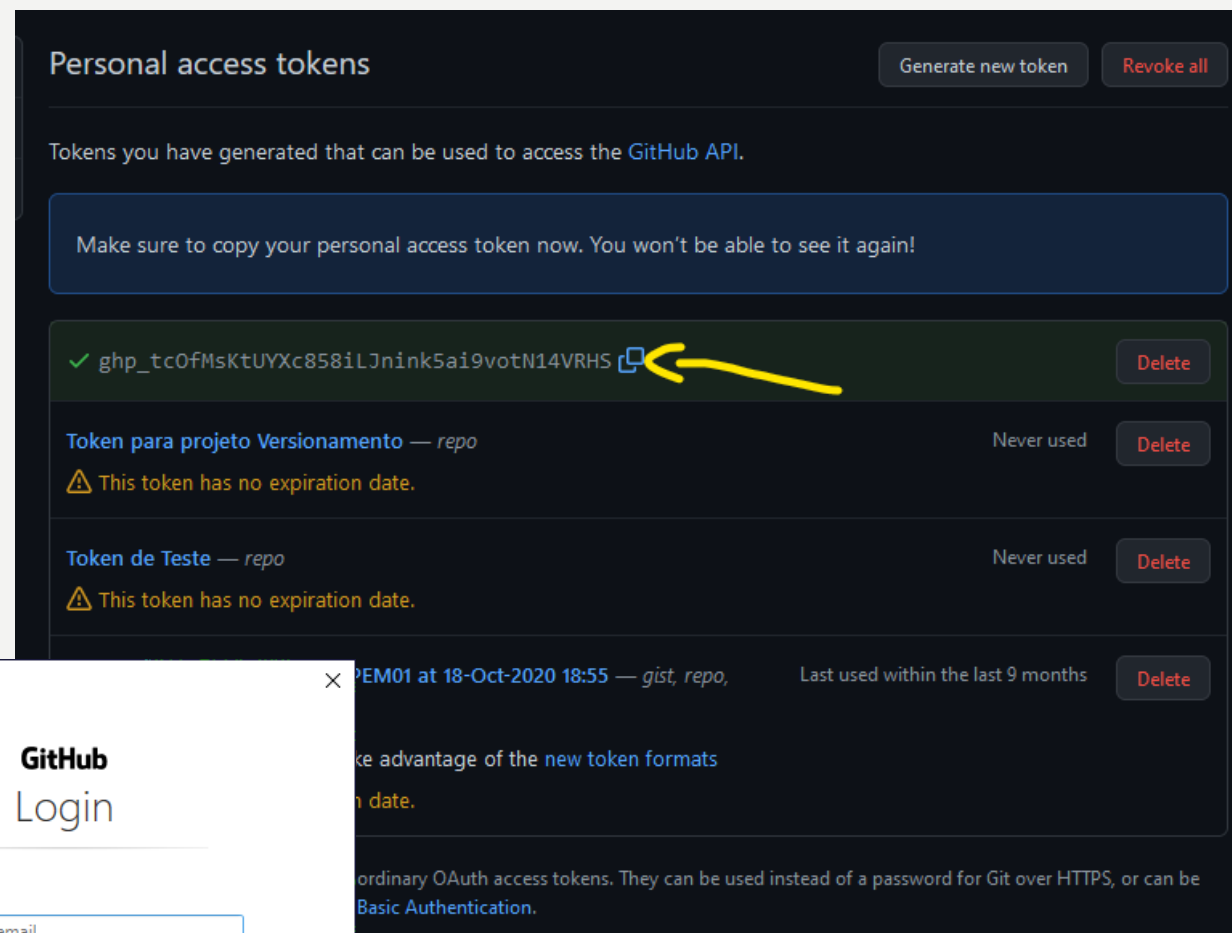
<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry

AUTENTICAÇÃO POR TOKENS GITHUB

- Agora que temos o token criado, para fazermos as operações de push no Github, quando for solicitado o usuário e senha você deverá inserir o token que criamos, no campo de senha (Password)

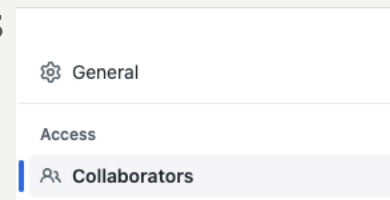
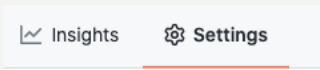


A screenshot of the GitHub Login dialog box. It has a title bar with the GitHub logo and the text 'GitHub Login'. The main content area has the text 'GitHub Login' in a large font. Below this are two input fields: 'Username or email' and 'Password'. At the bottom, there are two buttons: 'Login' with a checkmark icon and 'Cancel' with an 'X' icon. Below the buttons, there is a link that says 'Don't have an account? Sign up' and another link that says 'Forgot your password?'.

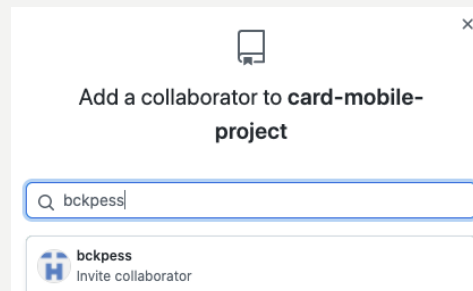
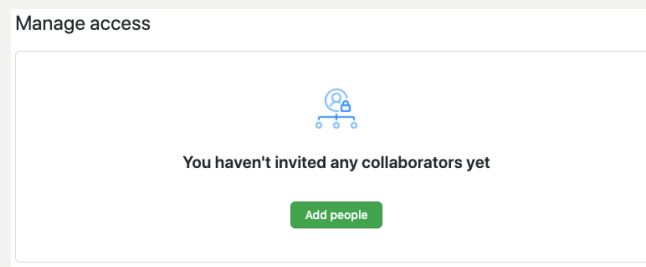


ATIVIDADE 3 – TRABALHAR EM EQUIPE

- Após o projeto criado no github, podemos adicionar outros colaboradores enviando convites.
- Após logado no github aparece a guia settings.
- Enviar Invite (convidar colaborador): Settings/Collaborators



- Clicar em Add people
- Pesquisar o usuário



ATIVIDADE 3 – TRABALHAR EM EQUIPE

- Agora é só aceitar o convite recebido por e-mail



@bckpress has invited you to collaborate on the
bckpress/GitProject repository

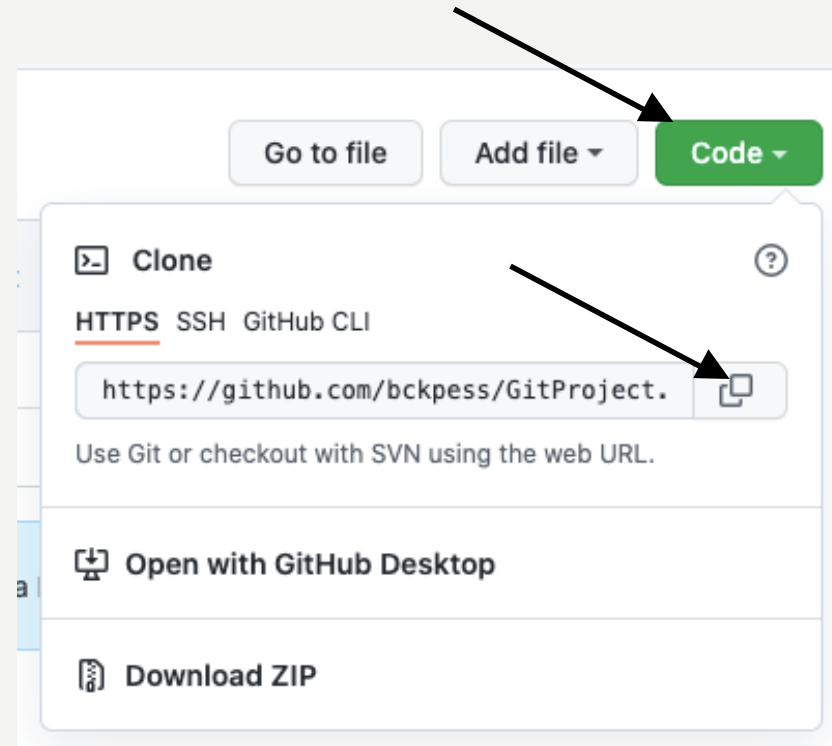
You can [accept or decline](#) this invitation. You can also head over to <https://github.com/bckpress/GitProject> to check out the repository or visit [@bckpress](#) to learn a bit more about them.

This invitation will expire in 7 days.

[View invitation](#)

ATIVIDADE 3 – TRABALHAR EM EQUIPE

- Após aceitar o convite, agora é só clonar o projeto com o comando *git clone url*



```
eduardodacosta@MacBook-Pro-de-Eduardo ~ % git clone https://github.com/bckpess/GitProject.git
```

ATIVIDADE 3 – TRABALHAR EM EQUIPE

- Cada pessoa deve trabalhar em sua Branch/Funcionalidade
- A Branch main é a principal e é utilizada para rodar o sistema, código de produção. Cada desenvolvedor deve trabalhar em sua funcionalidade e ao final enviar/merge para a branch main:
- Exemplo de Branch:
 - Main Versão final, de produção
 - Development (Para todos testarem antes de enviar para a main)
 - Login - funcionalidade em desenvolvimento
 - Cadastro de Clientes - funcionalidade em desenvolvimento

ATIVIDADE 3 – TRABALHAR EM EQUIPE

- `git branch` - mostra a branch local
- `git branch -a` - mostra todas as branches, local e remota
- `git branch login` - cria a branch login a partir da branch atual
- `git checkout login`
- criar um arquivo login e depois fazer o push e mostrar no GitHub
- Como resolver conflitos

RESOLVENDO CONFLITOS

- Conflitos são alterações feitas no mesmo arquivo e local por colaboradores diferentes, ao mesmo tempo!