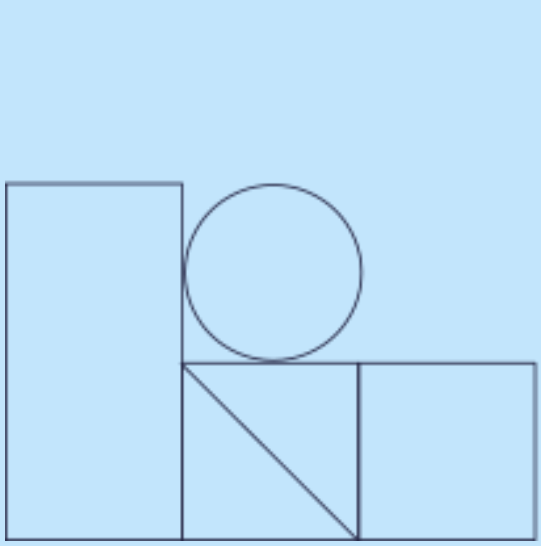
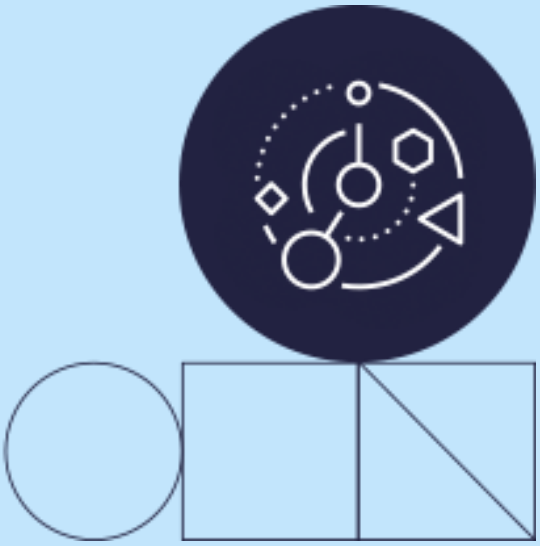
IBM SkillsBuild

IBM

Git e Github

Colaboração no Github





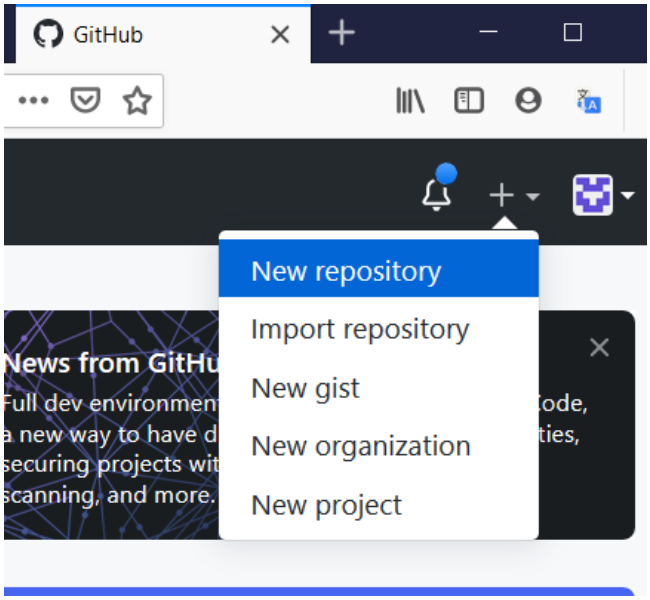
Índice

O que é colaboração em Git?

Colaboração é a forma como diferentes pessoas podem trabalhar juntas no mesmo projeto. É como criar um grupo no *GitHub*, assim como Grupos em outras redes sociais. As pessoas adicionadas à lista de colaboradores podem fazer push, *merge* e fazer outros tipos de coisas semelhantes no projeto.

Para colaborar, precisamos seguir os passos abaixo:

Passo 1: Criar um repositório



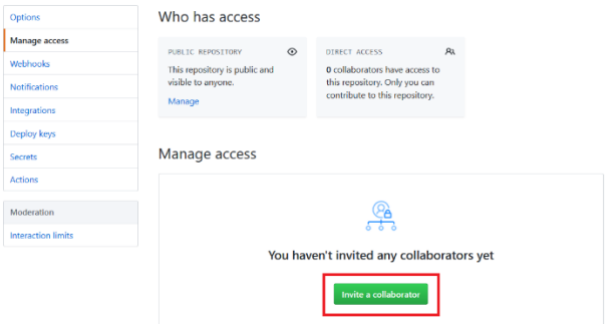
Passo 2: Adicione arquivos ao seu projeto.

Abriremos o GitBash em nosso diretório de trabalho local onde os arquivos são armazenados e executamos os seguintes comandos:

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git init

$ git add

$ git commit -m "initial commit"

$ git remote add origin remote
repository URL

$ git remote -v
```

Passo 3: Acrescentar nossos colaboradores.

Clique na configuração e, em seguida, siga os passos abaixo:



Após clicarmos em "Convidar um colaborador", preencheremos os detalhes necessários.

Nota: Para trabalhar perfeitamente em uma equipe colaborativa, é necessário conhecer as seguintes terminologias e suas aplicações.

Criar etiquetas assinadas e no assinadas

Primeiro, surge a pergunta: por que assinamos um código? A razão é que ele mostra a autoridade do código, quem o escreveu e quem deve ser culpado se surgir um erro. Para criar uma tag simples, vamos digitar:

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git tag ejemplo
```

Isto cria uma tag baseada na versão atual do repositório.

Para criar uma etiqueta anotada não assinada contendo uma mensagem, basta digitar:

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git tag -a v1 -m “Versión 1”
```

Finalmente, se quisermos assinar uma etiqueta, basta digitar:

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git tag -s Mytag
```

Para verificar sua etiqueta assinada, basta digitar:

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git tag -v mytag
```

Git checkout

O comando *git checkout* nos permite navegar entre as ramos criados pela ramo do *git*. Ele nos permite verificar os arquivos no diretório de trabalho para corresponder à versão armazenada naquele ramo.

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git checkout branch name
```

Este comando pode ser facilmente confundido com um *git clone*, mas os dois são diferentes. A diferença entre os dois comandos é que o clone trabalha para recuperar o código de um repositório remoto, alternativamente, o *checkout* trabalha para alternar entre versões de código que já estão no sistema local.

Outra terminología relacionada

push

Este comando é usado para empurrar arquivos de nossa máquina local para o repositório GitHub.

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git push -f origin master
```

fetch

Isto é útil quando se interage com um repositório remoto. Basicamente, este comando é usado para recuperar o trabalho feito por outras pessoas e atualizar nosso projeto local em conformidade.

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git fetch nombre_remoto
```

pull

Este é um método de apresentação de nossas contribuições ao projeto. Ocorre quando um desenvolvedor solicita que mudanças comprometidas em um repositório externo sejam consideradas para inclusão no repositório principal de um projeto após revisão por pares.

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git pull nombre_remoto
```

Obtenha a cópia especificada do controle remoto da filial atual e funda-a imediatamente com a cópia local. Este comando é o mesmo que *git fetch* seguido por *git merge*.

Remoção de qualquer etiqueta

Para remover quaisquer tags, navegaremos até nosso repositório GitHub local e digitaremos o seguinte comando:

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git tag -d tagName

$ git push origin :tagName
```

Busca de código

GitHub oferece aos usuários a possibilidade de buscar códigos com um repositório ou organização. Há também algumas restrições impostas por *GitHub* na busca de código. Estas restrições são impostas devido à complexidade do código de busca. Abaixo estão as buscas válidas que você pode realizar no *GitHub*.

- Busca por conteúdo do arquivo ou caminho do arquivo.
- Pesquisar os repositórios de um usuário ou organização.
- Busca por localização ou tamanho do arquivo.
- Busca por idioma.
- Busca por nome de arquivo ou extensão de arquivo.

Verificar registros anteriores

Para olhar para trás, para todos os *commits* que foram feitos com o repositório, podemos simplesmente usar:

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git log
```

Por padrão, se não forem passados argumentos com este comando. Ele exibirá os registros em ordem cronológica inversa.