

# Lembrete para uso do git

12 de abril de 2017

O texto a seguir é um breve lembrete das diversas etapas necessárias para se ter nossos projetos no **git**, baseados no [github.com](https://github.com). Dúvidas de conteúdo devem ser resolvidas pela consulta ao texto **Controle de versão.pdf** (apresentado na aula 9), que detalha os conceitos e a operação do **git** em maior profundidade, inclusive conceitualmente.

## 1. Cadastro

Todos alunos do grupo devem se cadastrar no GitHub, em <https://github.com/>

## 2. Criação do repositório

Apenas um dos alunos do grupo deve criar um repositório para o trabalho em grupo, em <https://github.com/new>

Não se deve esquecer de criar o conteúdo inicial, selecionando a opção “Initialize this repository with a README”.

## 3. Convite para o grupo

O criador do repositório deve convidar todos parceiros do grupo para que eles possam contribuir para o projeto.

Procure a guia **Settings** na página do projeto. E, dentro dela, escolha a opção **Collaborators**.

Eles serão notificados do projeto em sua página no GitHub, e devem aceitar o convite para que façam parte do projeto;

## 4. Instalação local do git

O **git** para MS-Windows pode ser baixado em <https://git-scm.com/download/win>

Para MacOS, use a URL

<https://git-scm.com/download/mac>

Para as diversas distribuições de GNU/Linux, Solaris e BSD, siga as instruções da página *Download for Linux and Unix*, em

<https://git-scm.com/download/linux>

## 5. Configuração local do git

Para configurar o **git** em seu computador, inclua os comandos que vão identificá-lo nos repositórios.

Um dos locais mais padronizados para se fazer isso é o terminal de comandos de seu computador. No MS-Windows, ele é o **cmd.exe**. No MacOS, nas distribuições GNU/Linux ele é o **shell**. Em qualquer caso, digite

```
git config --global user.name "Seu Nome"
```

onde *Seu Nome* deve ser trocado pelo nome do usuário, e depois digite

```
git config --global user.email "seu.email@mailserver.com"
```

## 6. Clonagem do repositório do [github.com](https://github.com)

Para criar uma cópia de seu repositório, hospedado no [github.com](https://github.com), clique no botão **Clone or download** na aba **Code** da página de seu projeto. Isto fará surgir uma pequena janela. Clique no botão que está à direita da caixa de texto que tem a URL iniciada por <https://github.com>

Crie um diretório que vai hospedar seu projeto **git**. Por exemplo, em sua pasta pessoal (que é mostrada quando se entra no terminal de comandos), crie o diretório **git**.

Então vá ao terminal de comandos (ou *shell*) comentado no item anterior e digite

```
git clone
```

e, sem pressionar **Enter**, cole no terminal a URL copiada do portal [github.com](https://github.com)

Finalmente pressione **Enter**. Isto vai transferir os arquivos de seu repositório no [github.com](https://github.com) para o repositório local, em seu computador.

Agora você pode trabalhar com o repositório local, manter um histórico de suas alterações dos arquivos e, finalmente, transferir arquivos do [github.com](https://github.com) para sua máquina e de sua máquina para o [github.com](https://github.com), através do comando `git`.

## 7. Criação de novos arquivos

Os novos arquivos de seu projeto devem todos ser criados no diretório que o `git` acabou de criar. Recapitulando, se você chamou o diretório para criar repositórios com o nome de `git`, como sugerido, e se seu projeto no [github.com](https://github.com) se chama Proj1, a operação `clone` criou o diretório Proj1 dentro do diretório `git`.

Crie seus arquivos de programa normalmente dentro deste diretório, através de seu editor preferido, que o `git` se encarregará de acompanhar suas versões;

## 8. Ciclo de trabalho

Nosso ciclo de trabalho no uso do `git` pode ser resumido em apenas duas etapas: trabalho no repositório local e transferência para o repositório remoto, no [github.com](https://github.com).

A questão da transferência será discutida nos itens 10 e 9, a seguir. Aqui falaremos do trabalho no repositório local.

A cada vez que um arquivo é alterado, é preciso informar ao repositório local. Isto é feito pelo comando

```
git add nome_do_arquivo
```

Naturalmente, *nome\_do\_arquivo* é um termo genérico que deve ser substituído por cada um dos arquivos que se deseja incorporar ao projeto, ou que foi alterado. O próximo passo é transferi-los para o repositório local.

Isto é feito por

```
git commit -m "Descrição da alteração efetuada"
```

Naturalmente, também aqui *Descrição da alteração efetuada* é um termo genérico que deve ser substituído por uma mensagem esclarecedora – **entre aspas** –, que descreve tão bem quanto possível as alterações feitas neste conjunto de atualizações.

Esta mensagem será essencial para entendermos as alterações efetuadas quando estivermos lendo o histórico do projeto – por exemplo, para descobrir onde surgiu um *bug* no código.

## 9. Transferência para repositório local

Antes de transferir as alterações para o repositório do [github.com](https://github.com), é preciso confirmar se algum de nossos colegas não alterou algum dos softwares. Para isso, usamos o comando

```
git pull
```

Aqui, a palavra *pull* tem o sentido de “puxar”, ou trazer, as alterações para nosso repertório local.

Se algum dos colegas tiver trabalhado exatamente onde trabalhamos, o `git` indicará um conflito de alterações. E deixará marcas no arquivo para podermos resolver o conflito.

## 10. Transferência para repositório no [github.com](https://github.com)

Finalmente, depois de terminada uma sessão de trabalho é hora de atualizar os arquivos no [github.com](https://github.com). Para isso, basta fazer

```
git push
```

Aqui, a palavra *push* tem o sentido de “empurrar”, ou levar, as alterações para o repositório comum, que fica na nuvem do [github.com](https://github.com)

Neste momento, poderá ser solicitada o *login* no [github.com](https://github.com). Use o mesmo endereço de *e-mail* fornecido antes. E, naturalmente, a mesma senha.