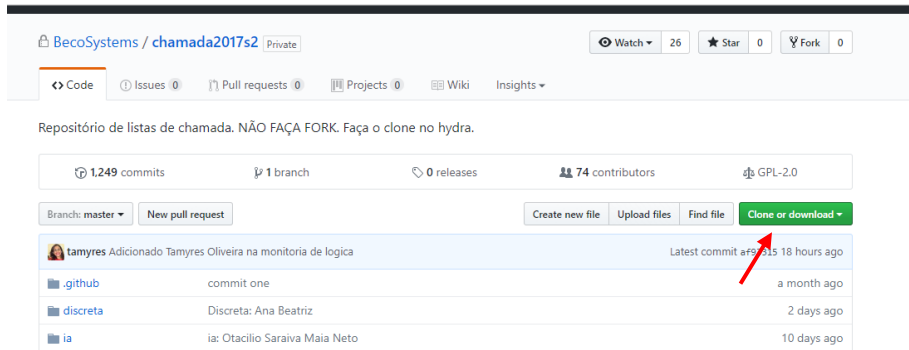


Sequência de comandos em Trabalhos do Github

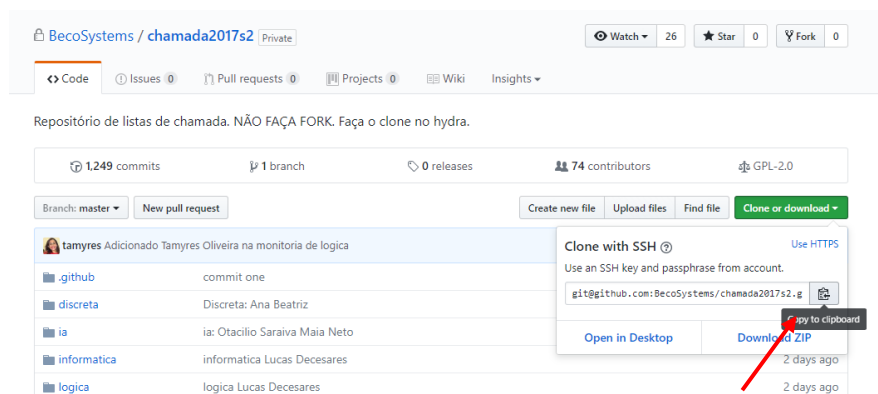
1. Clonando o repositório

1.1. Vá até o Github e clone o repositório de trabalho

Obs: NÃO FAÇA FORK



1.2. Prefira o clone com SSH



1.3. No hydra, na pasta de rascunho da respectiva disciplina do trabalho

```
$ git clone git@github.com:BecoSystems/exemplo.git
```

2. Acesso inicial ao repositório

```
$ cd exemplo (entra no repositório)
```

2.1. Neste primeiro acesso, você entrará direto no ramo master. Nos próximos acessos, ficará no último ramo que você entrou. Inicialmente temos os ramos **master** e **develop** (para mais detalhes, ler o hydrabox, ler os arquivos disponibilizados pelos monitores que já foram explicados e ir para as monitorias). É *recomendável* criar o ramo **feature-seunome**.

```
$ git co develop (vai para o ramo develop)
```

```
$ git co -b feature-seunome (cria seu ramo LOCAL)
```

2.2. No feature-seunome, você já pode fazer suas modificações no trabalho e ao finalizar, commitar normalmente

```
$ vi exn.pl
```

```
$ git cm "Sua mensagem de alteração aqui"
```

2.3. Tudo feito no feature-seunome, é hora de fazer merge com o develop e enviar para o Github.

Obs: Todas as alterações devem ser feitas com os ramos atualizados, por isso sempre lembrem de dar **git pull** no **develop** antes de fazer as modificações para não dar conflito.

```
$ git co develop (vai para o develop)
```

```
$ git merge feature-seunome (faz merge com o feature-seunome e insere as modificações)
```

```
$ git push (manda as alterações para o github)
```

3. Depois do primeiro acesso

3.1. Se tiver no develop

```
$ git pull (para atualizar)
$ git co feature-seunome (para entrar no ramo)
$ git merge develop (para atualizar o feature-seunome)
$ vi exn.pl (faz as devidas modificações)
$ git cm "Sua mensagem de alteração aqui"
$ git co develop (para ir para o develop)
$ git merge feature-seunome (envia as modificações feitas para o develop)
$ git push (manda para o github as mudanças)
```

3.2. Se tiver no feature-seunome, a única diferença vai ser que como o feature-seunome é um ramo local, não tem como atualizar ele com o **git pull**, por isso é preciso ir para o **develop** para fazer merge.

```
$ git co develop
$ git pull
$ git co feature-seunome
$ git merge develop
$ vi exn.pl
$ git cm "Sua mensagem de alteração aqui"
$ git co develop
$ git merge feature-seunome
$ git push
```

4. Release no master: só deve ser feito quando o programa estiver em boas condições de uso. Cuidado para não estourar os prazos!

4.1. Criação do ramo release-0.1

```
$ git co develop (caso esteja fora do ramo)
$ git co -b release-0.1 develop (cria o ramo LOCAL, ou seja, funciona como o feature-seunome)
```

4.2. Após quaisquer modificações no ramo **release-0.1**, é hora do merge com o **master**

```
$ git co master
$ git pull
$ git merge release-0.1
$ git push
$ git tag v0.1 (cria a tag da versão atual)
$ git push --tags (envia a tag)
```

4.3. Merge do ramo **release-0.1** com o **develop**

```
$ git co develop
$ git pull
$ git merge release-0.1
$ git push
```

4.4. Apagar o ramo release-0.1 após todos os procedimentos acima

```
$ git br -d release-0.1
```

4.5. Lembrar de atualizar o **feature-seunome** também

```
$ git co feature-seunome
$ git merge develop
```