Professor Rodrigo Bossini
https://sites.google.com/site/professorrodrigobossini/

Git – Linha de Comando – Commits e Releases

1 Introdução

Neste material iremos estudar o fluxo básico do controle de versão usando o Git na linha de comando. Quando uma versão estiver pronta para ser colocada em produção (ou entregue para o seu professor), uma release (tag) será criada.

2 Passo a passo

2.1 Certifique-se de que você possui uma conta no Github. Você pode criar uma gratuitamente visitando o Link 2.1.1.

Link 2.1.1
https://github.com/

2.2 Crie um arquivo textual usando a sua linguagem de programação preferida, como Java, por exemplo. Use qualquer IDE simples, como o Visual Studio Code. Neste material não é necessário compilar e/ou executar qualquer código. Basta fazer as alterações indicadas para que o uso do Git faça sentido. Entenda que o controle de versão se aplica a qualquer tipo de arquivo, mesmo que não envolva programação.

2.3 Considere que a primeira versão de interesse é dada na Listagem 2.3.1. Ela ainda não é algo entregável ou para ser colocado em produção. Mas trata-se de um trecho de código que deve ser guardado para que possa eventualmente ser recuperado no futuro.

Listagem 2.3.1

```
public class EntregaGit {
public static void main (String [] args){
System.out.println ("Entregando tarefas com o Git na linha de comando...");
}
}
```

2.4 O Git armazena as informações de controle de versão de arquivos em um único arquivo oculto chamado .git. Para criá-lo, execute, no terminal, o seguinte comando

git init

A partir daí, os arquivos existentes no diretório atual poderão ter sua versão controlada pelo git.

2.5 No momento, o arquivo que criamos não está sob controle de versão. Você pode verificar o status do repositório com o comando

git status

Precisamos dizer que desejamos que o controle de versão seja feito explicitamente. Para tal, usamos o seguinte comando

git add EntregaGit.java

Assim, o arquivo passa a fazer parte do controle de versão do Git e ele estará incluso no próximo commit. Verifique novamente o status do repositório com

git status

2.6 Digamos que desejamos armazenar essa versão do arquivo permanentemente, ou seja, fazer um commit. Para isso, basta usar o seguinte comando. Note que a cada commit sempre há uma mensagem associada que o descreve, a fim de que outros desenvolvedores tenham fácil entendimento sobre o que foi feito até ali.

git commit -m 'Primeira Versao'

2.7 O Git é um sistema de controle de versão local. Caso queiramos armazenar o histórico gerado por ele em um servidor remoto, precisamos adicionar uma referência ao servidor escolhido no repositório local. Para isso, o primeiro passo é criar um repositório no servidor remoto escolhido. No nosso caso, o Github. Vá até o Github e crie um repositório público. O nome do seu repositório deve ser parecido com

https://github.com/professorbossini/pessoal_guia_entrega_linha_comando

Caso esteja resolvendo os exercícios da Semana 1 de uma disciplina chamada Programação Avançada, talvez você queira chamar seu repositório de solucoes_semana1_prog_avancada, ou algo parecido com isso.

2.8 A seguir, é preciso adicionar uma referência ao repositório remoto, o que pode ser feito com o comando

git remote add origin
https://github.com/professorbossini/pessoal_guia_entrega_linha_comando.git

Note que é necessário alterar seu usuário e nome de repositório no link a ser adicionado.

Com isso, o nome "origin" está associado ao repositório remoto, o que quer dizer que sempre que fizermos um upload direcionado para "origin", o conteúdo será direcionado para esse

repositório. Note que o nome “origin” não tem nada de especial. Você pode escolher qualquer outro nome e um repositório local pode ter diferentes referências para repositórios remotos diferentes, utilizando outros nomes. O importante é utilizar o nome escolhido na hora de fazer o envio, como faremos a seguir.

2.9 Para enviar o código para o servidor Git remoto (neste caso, o Github), use o seguinte comando

```
git push origin master
```

Faça login com seu usuário e senha do Github e seu código deverá ser enviado ao servidor remoto.

2.10 Note que você ainda não possui algo entregável. Você fez upload de algo que tornou permanente mas que não necessariamente está pronto para ir para produção (ou ser entregue ao professor). Possivelmente seu trabalho passará por muitas outras atualizações antes de poder ser entregue. Por enquanto, o envio ao servidor remoto foi feito apenas para que o trabalho feito até então não seja perdido. Consideremos que é necessário realizar mais uma atualização no conteúdo, conforme mostra a Listagem 2.10.1.

Listagem 2.10.1

```
import javax.swing.*;
public class EntregaGit {
public static void main (String [] args){
System.out.println ("Entregando tarefas com o Git na linha de comando...");
String nome = JOptionPane.showInputDialog("Qual seu nome?");
JOptionPane.showMessageDialog(null, "Oi, " + nome);
}
}
```

2.11 Suponha, também, que esta versão está pronta para ser entregue. Antes de fazer isso, precisamos tornar essa alteração permanente e enviá-la para o servidor remoto. Use os seguintes comandos para isso

```
git add  EntregaGit.java
git commit -m ‘Captura nome e exibe. Exercicio pronto.’
git push origin master
```

Agora desejamos marcar esse ponto em que o repositório está como “importante”. Associar a esse ponto um nome (por exemplo v1.0.0 ou EntregaExercicio1) que lembre a razão pela qual ele é importante. Isso é feito por meio da especificação de uma **tag**. Há dois tipos de tags: as “leves” e as “anotadas”. Vamos usar somente as anotadas. Uma tag anotada contém um nome, uma mensagem, a data, o commit a que se refere etc. Use o seguinte comando para adicionar uma tag ao repositório

```
git  tag -a EntregaExercicio1 -m ‘Minha primeira entrega para o professor’
```

Se quiser verificar a existência dela, use

```
git show EntregaExercicio1
```

Acabamos de adicionar uma tag ao projeto, porém ela existe apenas localmente. Para enviá-la ao servidor remoto, use

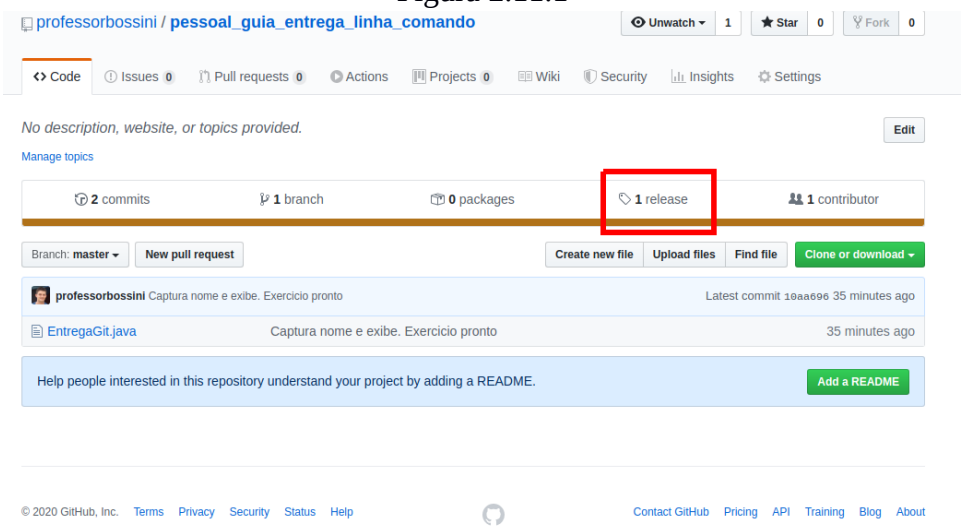
```
git  push origin EntregaExercicio1
```

Uma outra alternativa seria o comando a seguir, que envia todas as tags existentes no repositório local que ainda não foram enviadas para o servidor remoto.

```
git  push origin --tags
```

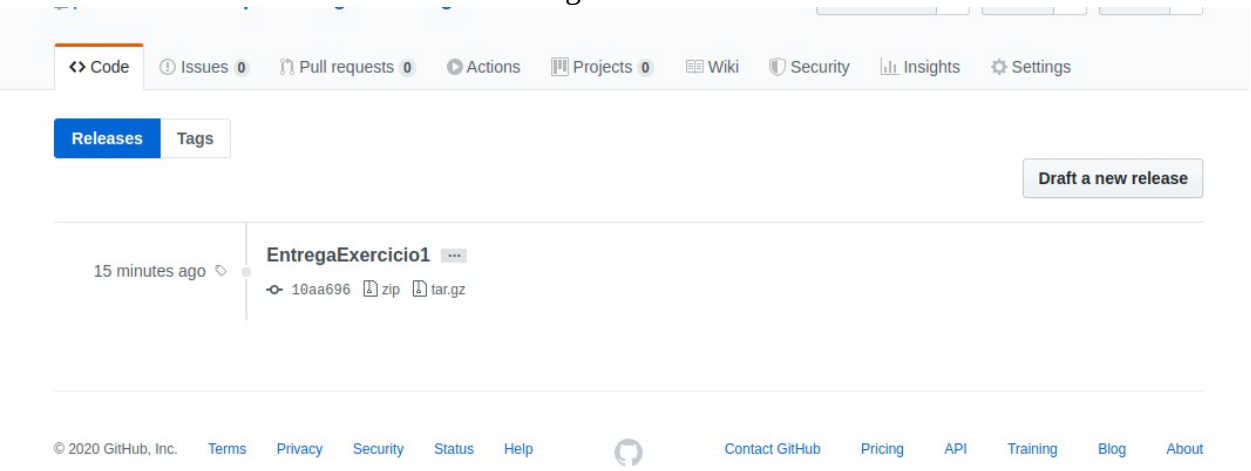
Agora visite o seu repositório no Github e encontre o Link para releases, como mostra a Figura 2.11.1.

Figura 2.11.1



Clique No link de releases e você verá a sua lista de releases, como na Figura 2.11.2.

Figura 2.11.2



Clique no link da release desejada para obter seu link completo. Neste exemplo, o link correto é

https://github.com/professorbossini/pessoal_guia_entrega_linha_comando/releases/tag/EntregaExercicio1

Esse é o padrão do link que você deverá entregar:

https://github.com/usuario/seurepositorio/releases/tag/suatag

Referências

Git – Book. Git, 2020. Disponível em: https://git-scm.com/book/en/v2. Acesso em: 17/02/2020.