



# BCC

# Engenharia de Software

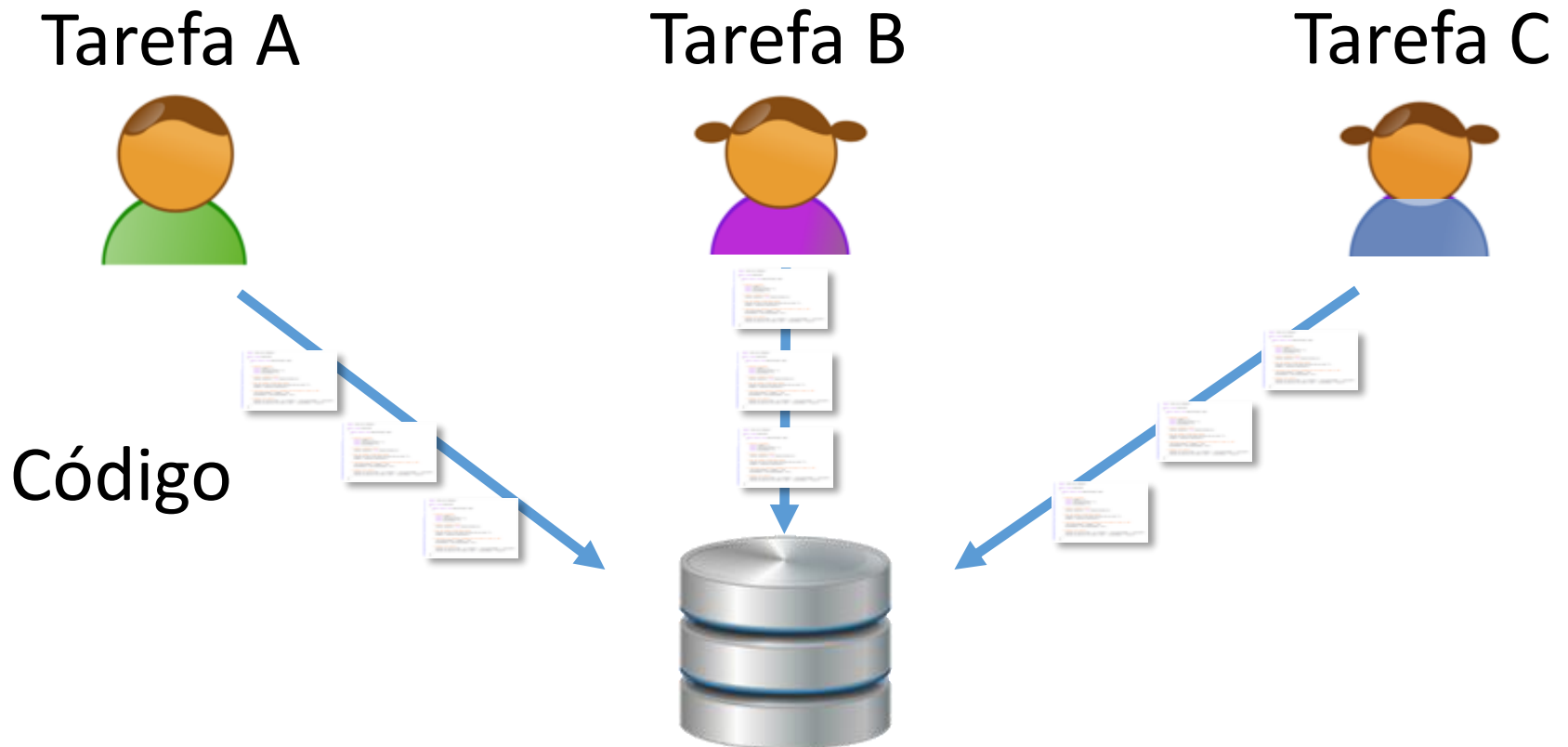
# 2018.2

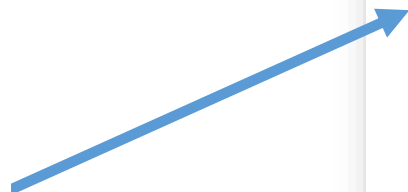
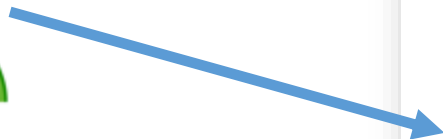
Professor Rodrigo Andrade

# Controle de Versão

-Git-

# Desenvolvimento colaborativo de software





```
class AlunoController {  
  
  static allowedMethods = [save: "POST", update: "PUT", delete: "DELETE"]  
  
  def index(Integer max) {  
    params.max = Math.min(max ?: 10, 100)  
    respond Aluno.list(params), model:[alunoInstanceCount: Aluno.count()]  
  }  
  
  def show(Aluno alunoInstance) {  
    respond alunoInstance  
  }  
  
  def create() {  
    respond new Aluno(params)  
  }  
  
  @Transactional  
  def save(Aluno alunoInstance) {  
    if (alunoInstance == null) {  
      notFound()  
      return  
    }  
  
    if (alunoInstance.hasErrors()) {  
      respond alunoInstance.errors, view:'create'  
      return  
    }  
  }  
}
```

# Qual a maneira de coordenar essas atividades?

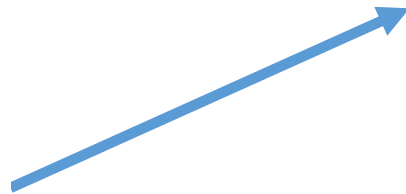
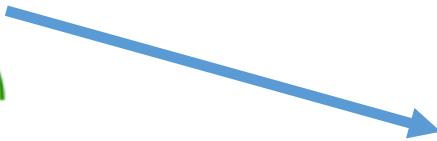


# Sistemas de Controle de Versão - SCV

# Problemas de conflitos

- Como resolver mudanças em um mesmo arquivo?
- Quais linhas foram alteradas por quais desenvolvedores?
- Quando as linhas foram modificadas?

# Desenvolver em um repositório compartilhado

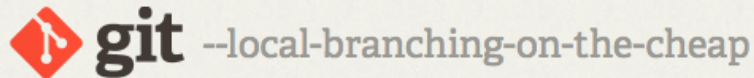


Computação na nuvem





# Git



🔍 Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.



**Learn Git in your browser for free with Try Git.**



# Github



Search GitHub

[Pull requests](#) [Issues](#) [Gist](#)

## Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

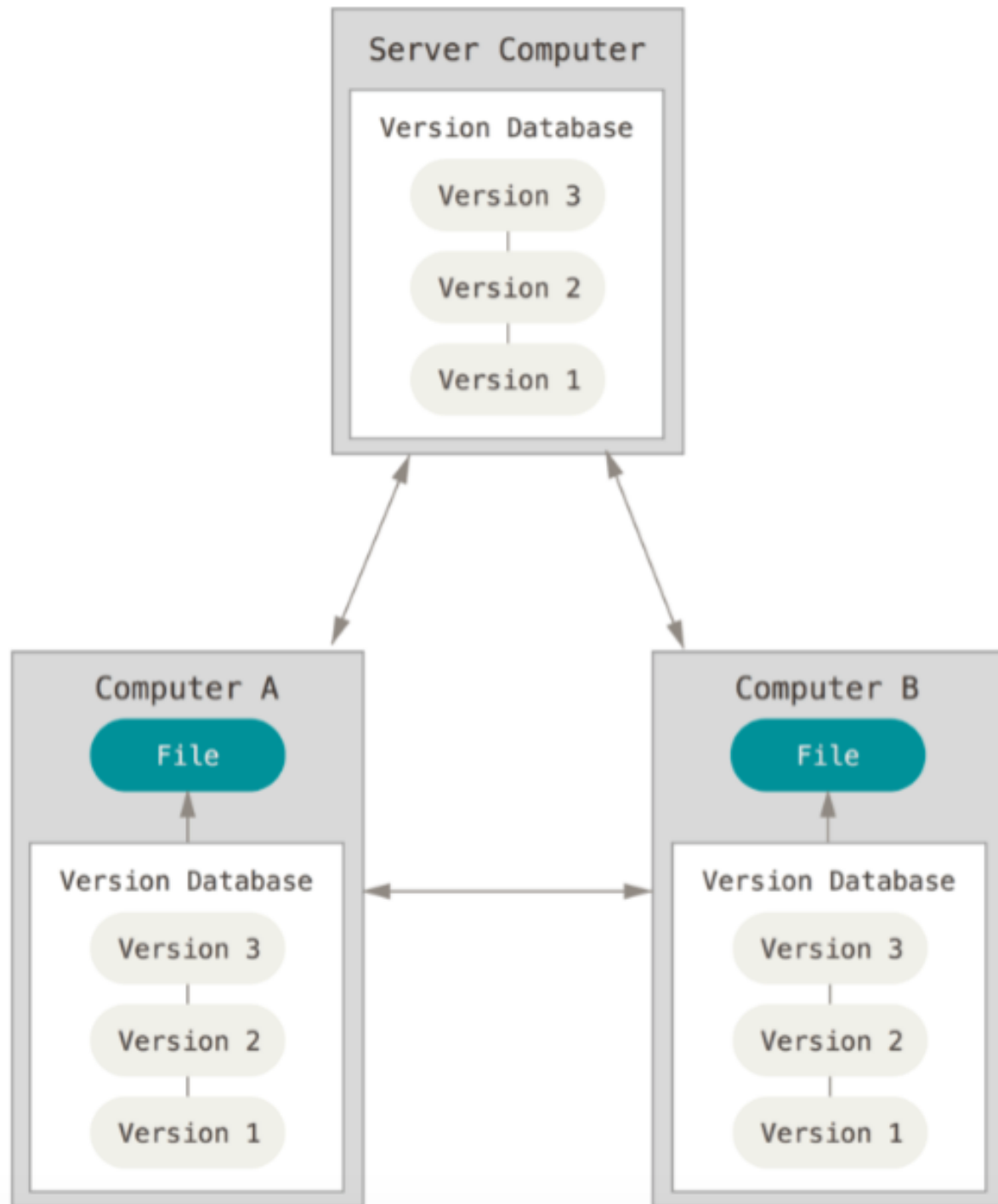
**Let's get started!**

Git

# Sistema de Controle de Versão - Git

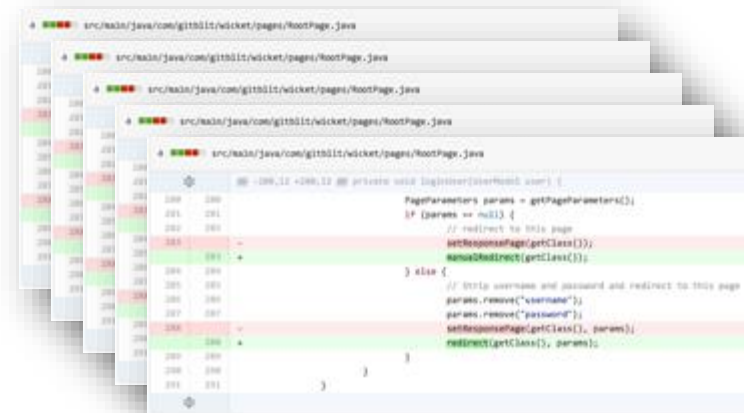
- Sistema que **registra alterações** em arquivos ao longo do tempo
- Nós podemos **recuperar versões** específicas desses arquivos
- Permite **voltar a uma versão antiga** dos arquivos
- Permite saber **quem** fez alterações nos arquivos

# SCV distribuído

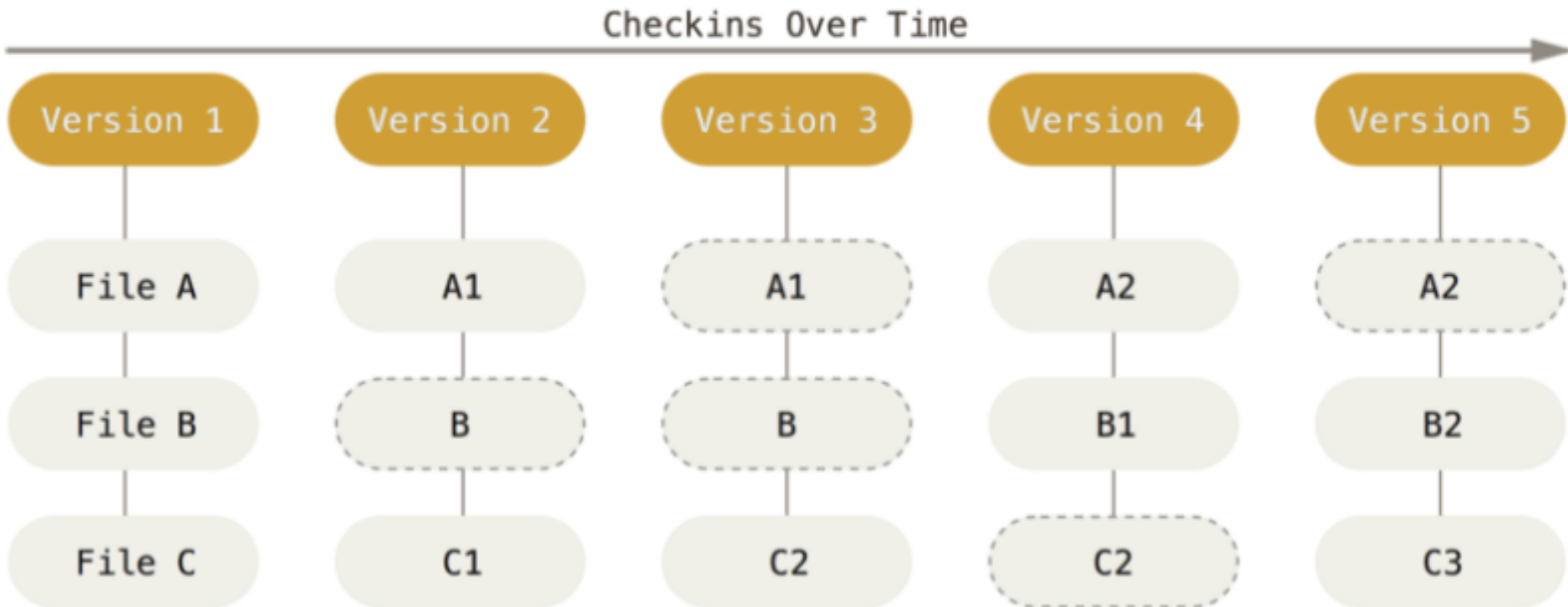


# Commits

- Registram um conjunto de alterações feitas pelo desenvolvedor
- Cada commit tem uma hash única associada



# Snapshots



# Estágios do Git

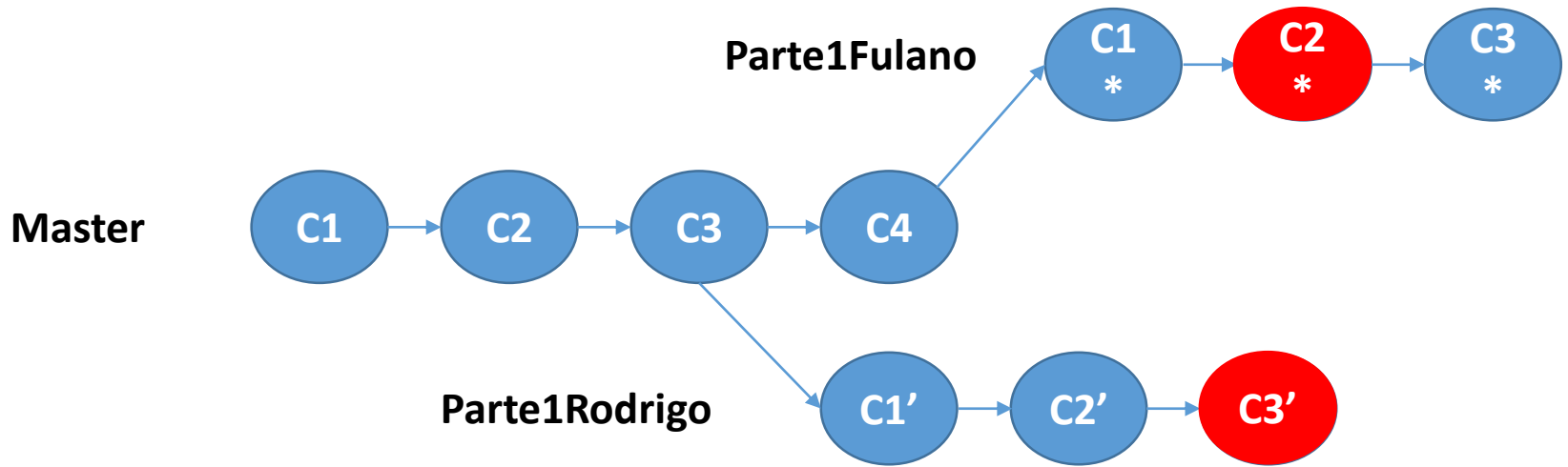
- Modified
  - Arquivos alterados, mas ainda não foram commitados
- Staged
  - Arquivos modificados foram assinalados para o próximo commit que você fizer
- Committed
  - Dados armazenados corretamente no repositório **local**

# Branches

- Representa uma linha independente de desenvolvimento
- Simula um novo diretório do projeto
  - Evita interferir na linha principal de desenvolvimento



# Exemplo de branches



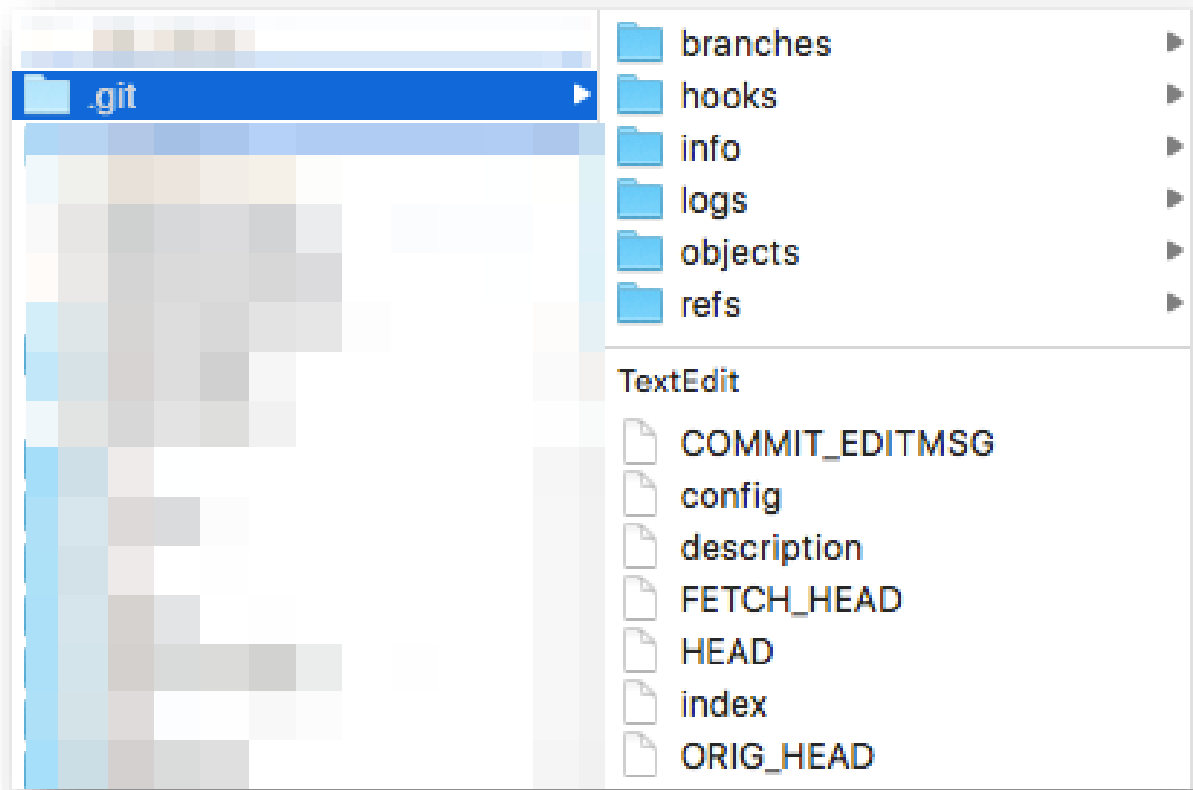
# Merge

- Mecanismo para resolver conflitos de código
- Integra uma contribuição em uma linha de desenvolvimento

```
23  
24 class String  
25 <<<<<< HEAD:lib/jekyll/core_ext.rb  
26   def cutoff(desired = 5)  
27     =====  
28     def cutoff(desired = 400)  
29 >>>>>> conflicts:lib/jekyll/core_ext.rb  
30     return self if self.length <= desired
```

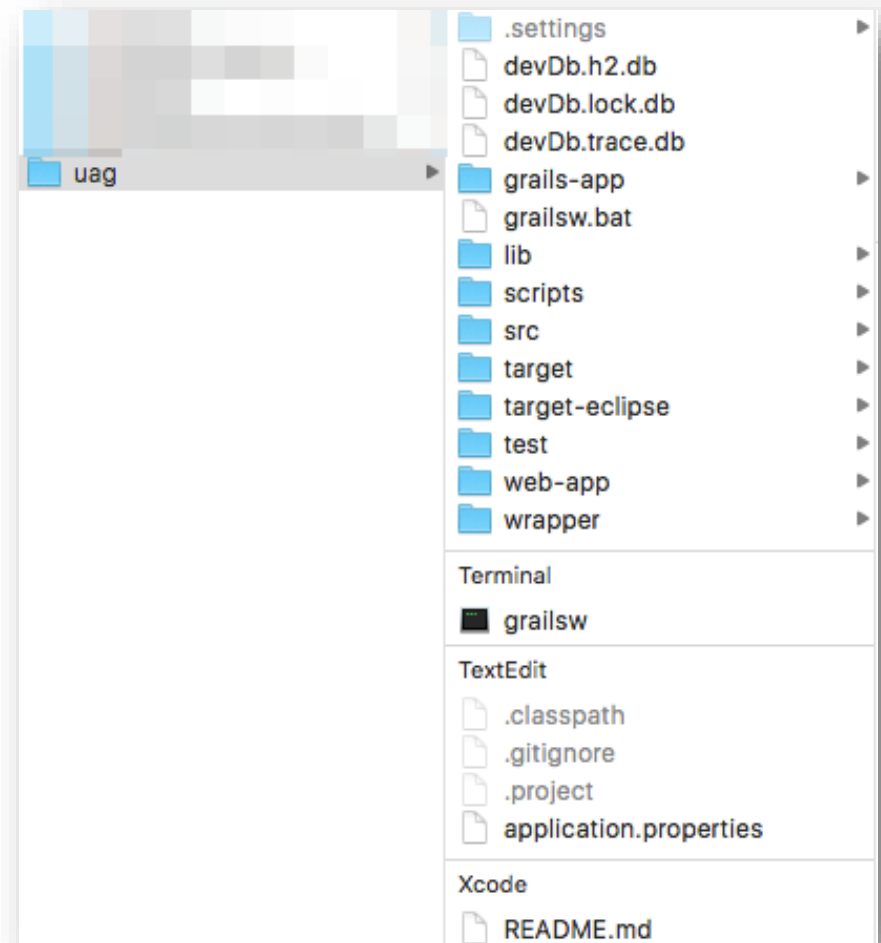
# Git directory

- Diretório que armazena **meta-data**
- Armazena a base de dados das alterações do projeto



# Working directory

- Representa uma versão específica de um conjunto de arquivos
- É o diretório que contém os arquivos que modificamos/trabalhamos



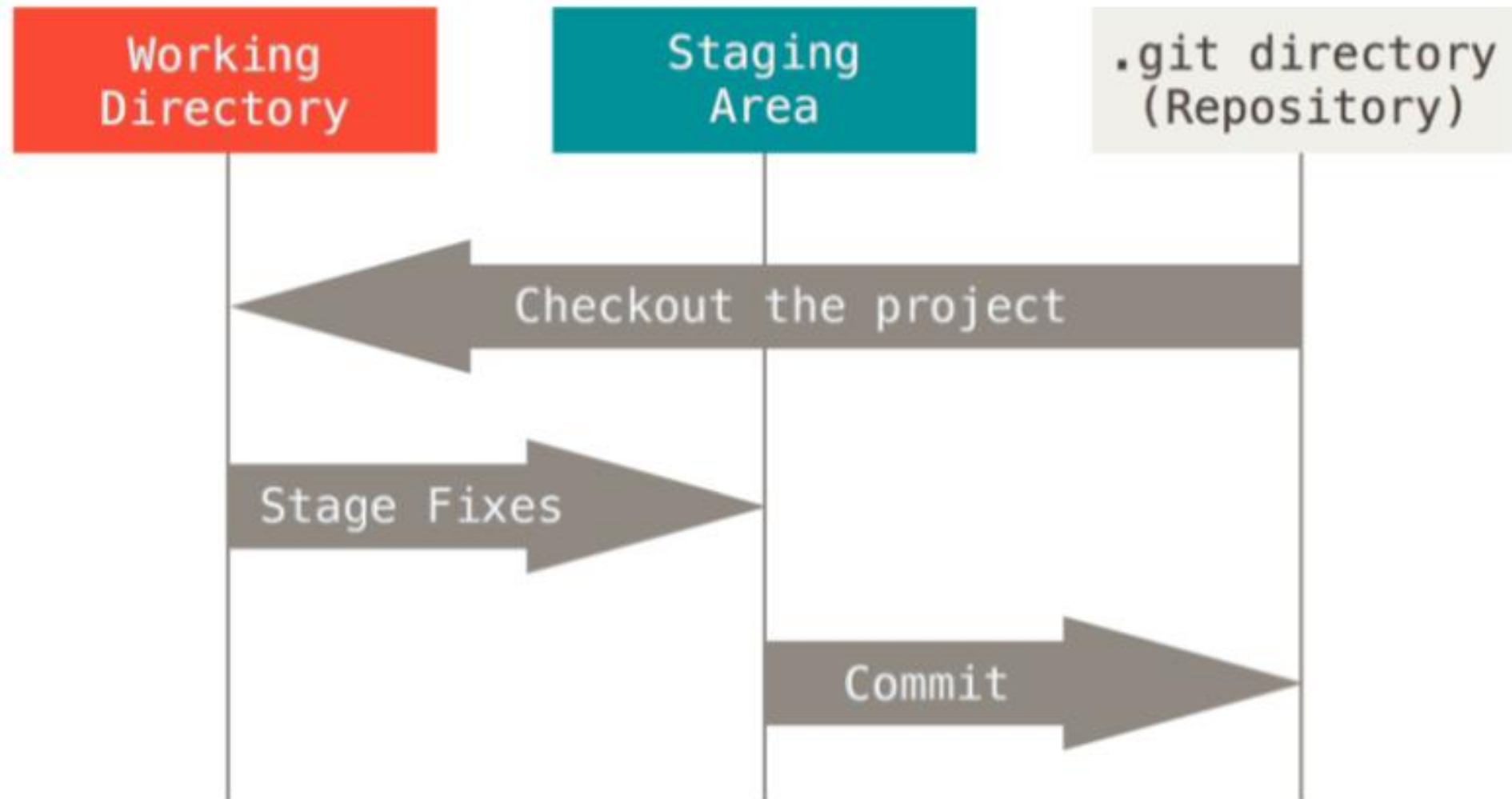
# Staging area

- É um arquivo que armazena informações sobre o que será incluído no próximo commit

```
rodrigo@DESKTOP-41LMC1Q:/mnt/c/ufrpe/workspace-ruby/es/blog$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   .idea/workspace.xml
```

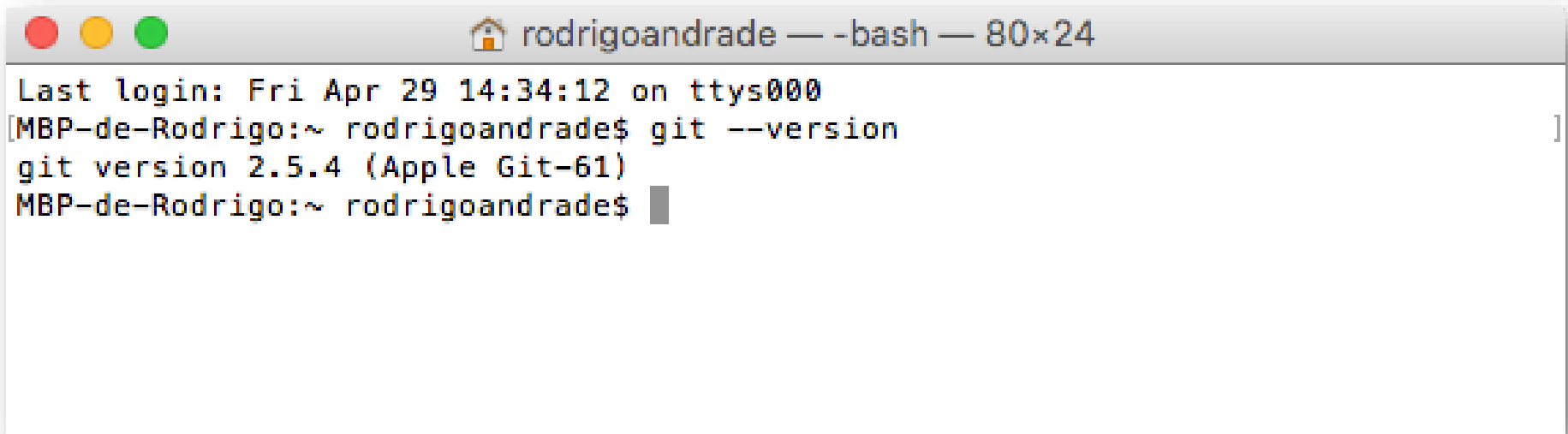
# Projeto Git



# Workflow do Git

- Você modifica arquivos no **Working directory**
- Você coloca os arquivos modificados no **Staging area** (snapshot)
- Você faz um commit para registrar o snapshot permanentemente no **.git directory**

# Linha de comando

A screenshot of a macOS terminal window. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left, a home icon followed by the text 'rodrigoandrade — -bash — 80x24' in the center, and a close button on the right. The terminal content shows a login message, a command to check git version, and the output of that command. The prompt is 'MBP-de-Rodrigo:~ rodrigoandrade\$' and there is a cursor at the end of the last line.

```
Last login: Fri Apr 29 14:34:12 on ttys000
[MBP-de-Rodrigo:~ rodrigoandrade$ git --version
git version 2.5.4 (Apple Git-61)
MBP-de-Rodrigo:~ rodrigoandrade$
```




# Instalando o Git no Windows




- Link para baixar o programa
  - <http://git-scm.com/download/win>
- Definindo sua identidade
  - `git config --global user.name "Fulano"`
  - `git config --global user.email fu@email.br`


Github

# Rede social


 Search GitHub


Pull requestsIssuesGist


  





**Rodrigo Andrade**  
rcaa


 [rodrigo\\_cardoso@hotmail.it](mailto:rodrigo_cardoso@hotmail.it)

 Joined on Jul 13, 2012






 Contributions

 Repositories




 Public activity

 Edit profile

**Popular repositories**

 <a href="#">ToyExample-LayeredAspects</a>	1 ★
 <a href="#">101Companies</a>	0 ★
 <a href="#">Freemind-Increment1</a>	0 ★
 <a href="#">pomodoro</a>	0 ★
 <a href="#">ToyExample-Increment2</a>	0 ★

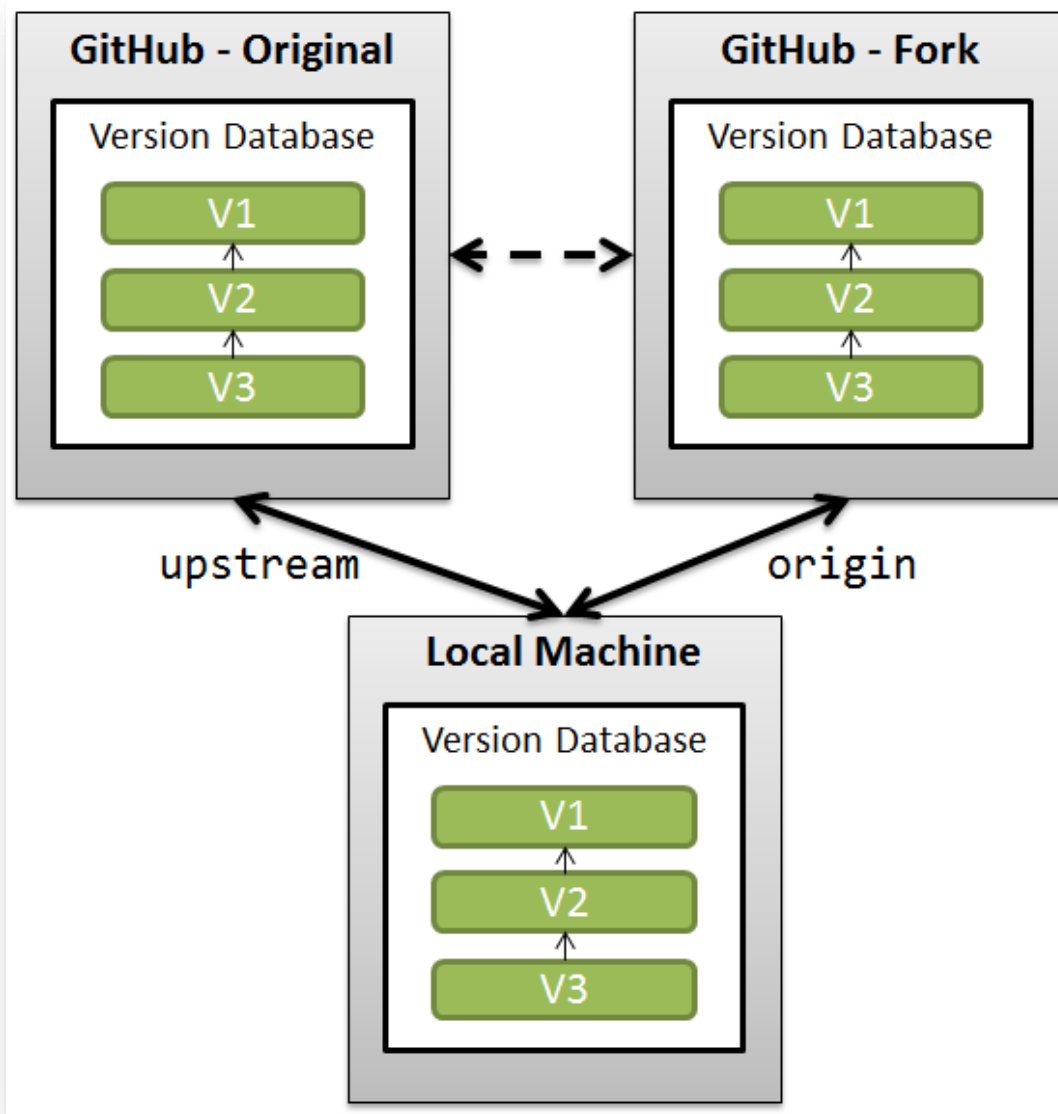
**Repositories contributed to**

 <a href="#">uag-es/EngenhariaSoftware</a>	0 ★
 <a href="#">gitblit/gitblit</a> pure java git solution	996 ★
 <a href="#">OpenRefine/OpenRefine</a> OpenRefine is a free, open source power tool ...	3,671 ★

# Características

- Serviço de hospedagem de repositório Git
- Fornece funcionalidades adicionais
  - Fork
  - Pull Request

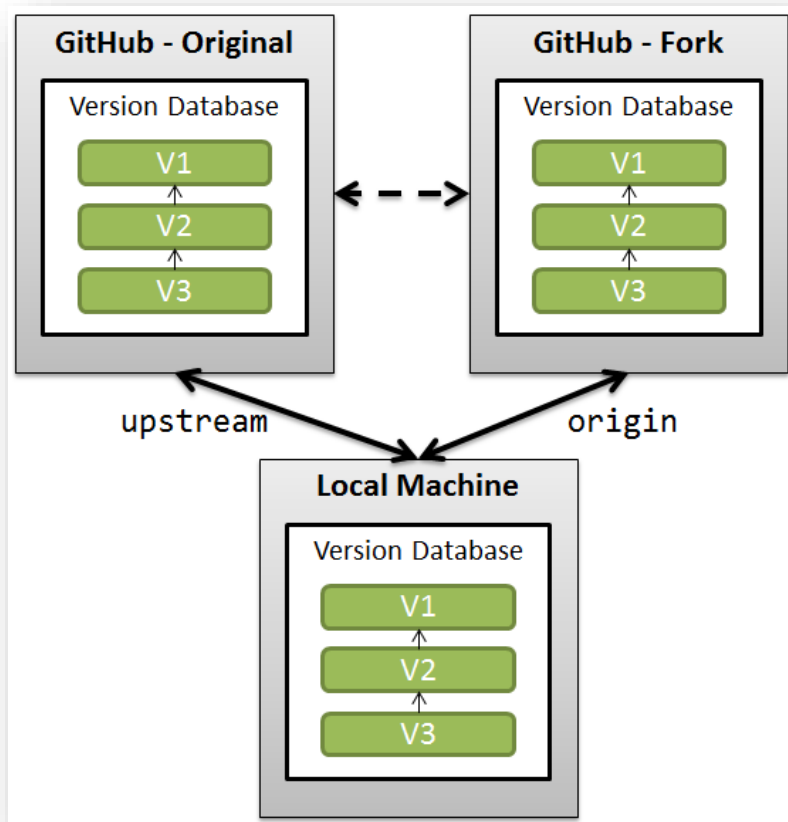
# Fork



 **Fork** 0

# Pull Request

- Notificação de que há uma contribuição pendente de aceitação



New pull request

# Git básico

# git clone

- Copia todos os arquivos e todas as suas versões de um projeto existente para o seu diretório local
- Mantém cópia completa
- Permite contornar problemas de servidor corrompido
- Inicializa o diretório Git

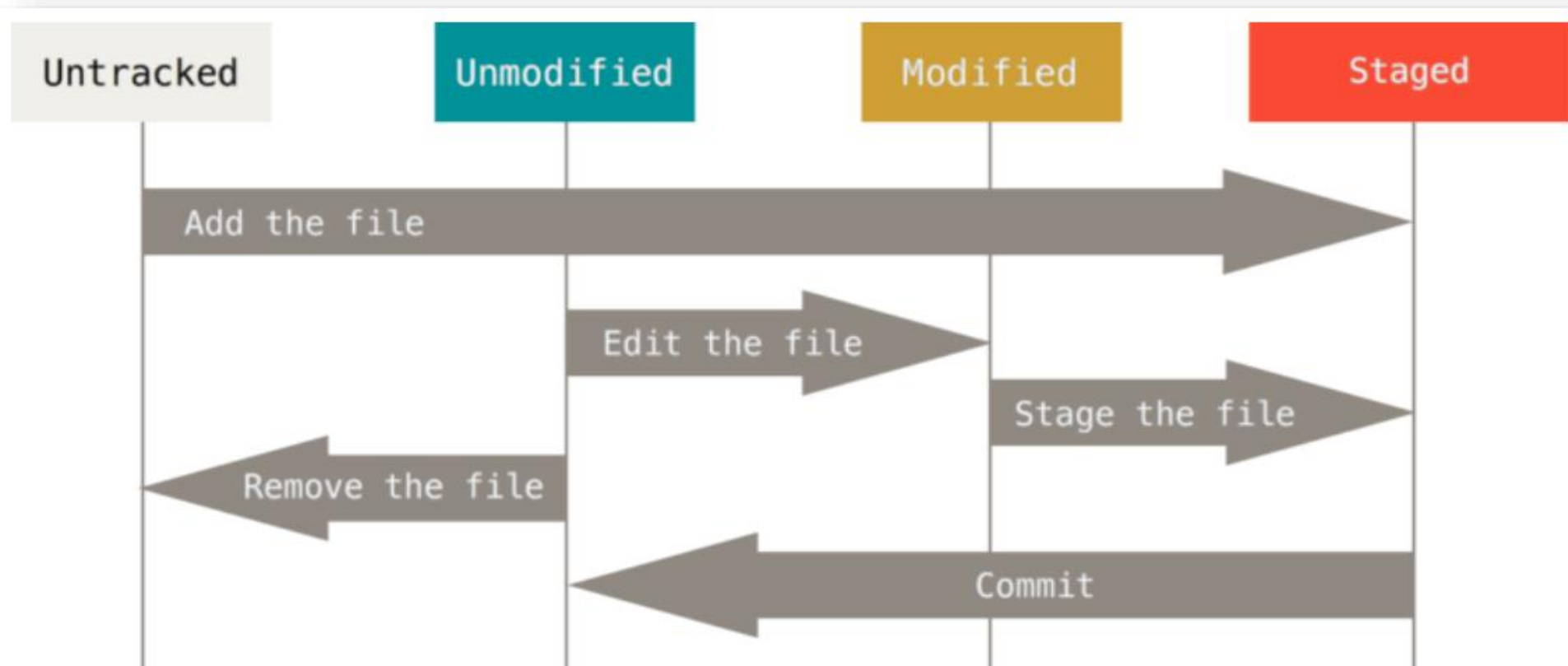
```
> git clone url
```

HTTPS ▼

<https://github.com/gitblit/git>



# Mudanças no repositório



# Verificar status dos arquivos

- Informa o estado dos arquivos
  - Untracked ou Staged

> git status

```
MacBook-Pro-de-Rodrigo:SimpleContributionExample rodrigoandrade$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```

```
MacBook-Pro-de-Rodrigo:SimpleContributionExample rodrigoandrade$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    temp/

nothing added to commit but untracked files present (use "git add" to track)
```

# Adicionando arquivos modificados

- Adiciona arquivos ao estado "staged"

```
> git add arquivo
```

```
> git add --all
```

```
[MacBook-Pro-de-Rodrigo:SimpleContributionExample rodrigoandrade$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   temp/temp.txt
```

# Ignorar determinados arquivos

- Criar um arquivo `.gitignore` no diretório raiz do repositório (working directory)
- Adicionar padrões usando um editor de texto:

```
*.class  
.settings/
```

# Verificando alterações nos arquivos

> git diff

> git diff --  
staged

```
diff --git a/src/text/Notepad.java b/src/text/Notepad.java
index 1a53450..b26ef11 100644
--- a/src/text/Notepad.java
+++ b/src/text/Notepad.java
@@ -8,8 +8,10 @@ public class Notepad extends JFrame {
    public Actions actions = new Actions(this);
    private String textPane;

+   // #if UndoAction
    UndoManager undo = new UndoManager();
    UndoAction undoAction = new UndoAction(this);
+   // #endif

    public Notepad () {
        textPane = new String();
@@ -19,4 +21,8 @@ public class Notepad extends JFrame {
    public String getTextComponent() {
        return textPane;
    }
+
+   public void setTextComponent(String textPane) {
+       this.textPane = textPane;
+   }
}
```

# Committed alterações

- Registra, de fato, as alterações no repositório

```
> git commit -m "mensagem"
```

```
MacBook-Pro-de-Rodrigo:shriram-example rodrigoandrade$ git commit -m "testing"  
[master 0ee9693] testing  
1 file changed, 6 insertions(+)
```

```
commit 0ee9693ae081efa06d83aeaad36655b7aed9536a  
Author: Rodrigo Andrade <rodrigo_cardoso@hotmail.it>  
Date: Fri May 6 16:36:24 2016 -0300  
  
testing
```

# Vendo a história de commits

> git log

```
commit 2625a31cc6097e60c14128e1fd0e8068dd0d6a9f
Author: Rodrigo Andrade <rodrigo_cardoso@hotmail.it>
Date: Thu May 5 14:15:11 2016 -0300

    adding syso example

commit 27e30181316d4f396c7cf1f12ea2c5f6a2426be4
Author: Rodrigo Andrade <rodrigo_cardoso@hotmail.it>
Date: Wed Feb 3 15:16:25 2016 -0300

    inheritance example 7

commit 9d0ba823b56b2aae978925c91e7bc4eec0f2939b
Author: Rodrigo Andrade <rodrigo_cardoso@hotmail.it>
Date: Wed Feb 3 15:12:28 2016 -0300

    inheritance example 6

commit 950298ca814f1639b5e70068a92ec9caa308ed63
Author: Rodrigo Andrade <rodrigo_cardoso@hotmail.it>
Date: Wed Feb 3 15:10:03 2016 -0300

    inheritance example 5
:...skipping...
commit 2625a31cc6097e60c14128e1fd0e8068dd0d6a9f
Author: Rodrigo Andrade <rodrigo_cardoso@hotmail.it>
Date: Thu May 5 14:15:11 2016 -0300

    adding syso example
```

Corrigindo erros



# Adicionando arquivos em commits já feitos

```
> git commit -m "commit inicial"
```

```
> git add arquivo_esquecido
```

```
> git commit --amend
```

# Tirando arquivos do estado staged

> git reset HEAD arquivo

```
[MacBook-Pro-de-Rodrigo:SimpleContributionExample rodrigoandrade$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    temp/

nothing added to commit but untracked files present (use "git add" to track)
```

# Revertendo alterações em arquivos

```
> git checkout -- arquivo
```

**CUIDADO**

**PERIGO**

# Trabalhando com remote

> git remote -v

```
MacBook-Pro-de-Rodrigo:gitblit rodrigoandrade$ git remote -v
origin  https://github.com/rcaa/gitblit.git (fetch)
origin  https://github.com/rcaa/gitblit.git (push)
```

> git remote add remote-name git-url

```
MacBook-Pro-de-Rodrigo:gitblit rodrigoandrade$ git remote -v
[origin  https://github.com/rcaa/gitblit.git (fetch)
origin  https://github.com/rcaa/gitblit.git (push)
[root    https://github.com/gitblit/gitblit.git (fetch)
root     https://github.com/gitblit/gitblit.git (push)
```

# Atualizando o repositório local com remotes

```
> git fetch remote-name
```

- Atualiza o repositório local
- Pode ser necessário fazer merge

# Enviando as suas alterações para um remote

```
> git push remote-name branch-name
```

# Comandos de branches

Criar novo branch

```
> git checkout -b nome-branch
```

Alterar para branch existente

```
> git checkout nome-branch
```

Verificar branches existentes

```
> git branch
```

# Fazer Merge

```
> git merge nome-branch
```



# Conflitos

```
<<<<<< HEAD:index.html
```

```
<div id="footer"> contato : </div>
```

```
=====
```

```
<div id="footer"> nos contate em </div>
```

```
>>>>>> iss53:index.html
```

# Resumo da aula

- Sistema de Controle de Versão
- Github
- Git

Fazer atividade de SCV



# BCC

# Engenharia de Software

# 2018.2

Professor Rodrigo Andrade