

# Gerenciamento de Configuração de Software

(Redmine, Git e Github)

Victor Hugo Santiago

victor.santiago@usp.br



# Agenda

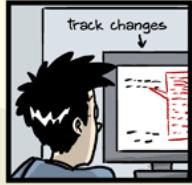
1. Introdução
2. Gerenciamento de Configuração de Software
3. Ferramentas de apoio
4. Controle de versões
5. Controle de mudanças
6. Integração contínua
7. **Redmine**
8. **Git e Github**
9. Práticas
10. Referências e outros materiais

# "FINAL".doc



FINAL\_rev.6.COMMENTS.doc

FINAL\_rev.8.comments55.  
CORRECTIONS.doc



FINAL\_rev.18.comments7.  
corrections9.MORE.30.doc



FINAL\_rev.22.comments49.  
corrections.10.#@%\$WHYDID  
ICOMETOGRADHSCHOOL?????.doc

# SIMPLY EXPLAINED

`budget_estimation_final_v1.1-ow.xlsx`

OR

`budget_estimation_last_version_2.xlsx`

OR

`budget_estimation_2012_10_25_ready_new.xlsx?`



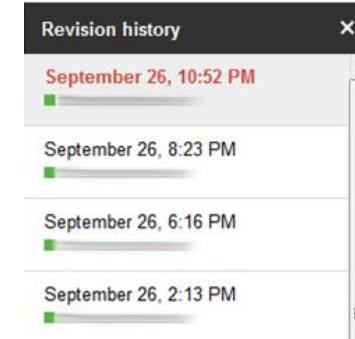
VERSION CONTROL

qualificacao-doutorado05.01.2016.zip  
qualificacao-doutorado06.01.2016.zip  
qualificacao-doutorado07.01.2016.zip  
qualificacao-doutorado08.01.2016.zip  
qualificacao-doutorado09.01.2016.zip  
qualificacao-doutorado10.01.2016.zip  
qualificacao-doutorado11.01.2016m.zip  
qualificacao-doutorado11.01.2016n.zip  
qualificacao-doutorado12.01.2016n.zip  
qualificacao-doutorado13.01.2016.zip  
qualificacao-doutorado14.01.2016.zip  
qualificacao-doutorado15.01.2016.zip  
qualificacao-doutorado15.01.2016noite.zip  
qualificacao-doutorado15.01.2016t.zip  
qualificacao-doutorado16.01.2016m.zip  
qualificacao-doutorado17.01.2016m.zip  
qualificacao-doutorado18.01.2016m.zip  
qualificacao-doutorado19m.zip  
qualificacao-doutorado20m.zip  
qualificacao-doutorado20n\_aplicadaRevSIMONE.zip  
qualificacao-doutorado22.01.2016n.zip



	Crash_Course_on_GIT_and_GitHub	29-Apr-15 23:55
	Crash_Course_on_GIT_and_GitHub	30-Apr-15 01:11
	Crash_Course_on_GIT_and_GitHub_rev01	30-Apr-15 09:11
	Crash_Course_on_GIT_and_GitHub_rev02	30-Apr-15 13:27
	Crash_Course_on_GIT_and_GitHub_rev03	30-Apr-15 20:29
	Crash_Course_on_GIT_and_GitHub_rev04	05-May-15 13:36
	Crash_Course_on_GIT_and_GitHub_rev05	07-May-15 19:21

Google docs



# ...e no desenvolvimento de software?

- Mudanças:

- não são **analisadas** antes de ser feitas
- não são **registradas** antes de ser implementadas
- não são **relatadas** àqueles que precisam saber
- ou não são **controladas** de forma que melhore a qualidade e reduza erros



# Gerenciamento de Configuração de Software (GCS)

- Durante o desenvolvimento do software queremos saber:
  - O que mudou e quando? (**controle de versão**)
  - Por que mudou? (**controle de mudanças**)
  - Quem fez a mudança? (**auditoria de configuração**)
  - Podemos reproduzir esta mudança? (**auditoria de configuração**)

# Gerenciamento de Configuração de Software

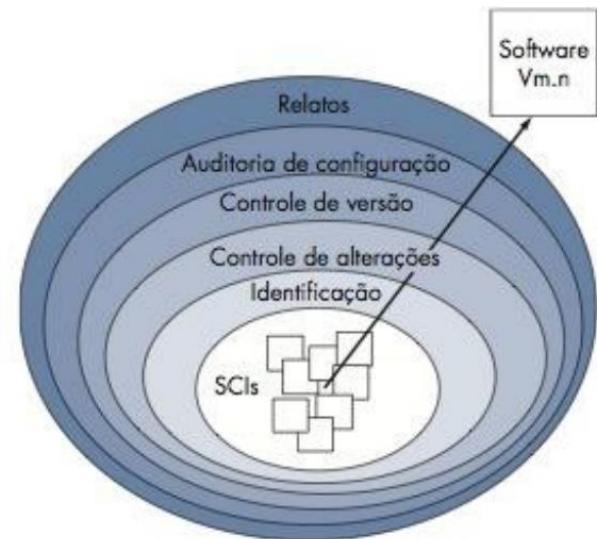
- Coordenar o desenvolvimento para minimizar a confusão!

*“Arte de **identificar, organizar e controlar modificações** no software que está sendo criado, maximizando a produtividade e reduzindo os erros” [Pressman, 2011]*

# Gerenciamento de Configuração de Software

As atividades são desenvolvidas para:

- Identificar a alteração
- Controlar a alteração
- Assegurar que a alteração seja implementada corretamente
- Relatar as alterações aos outros interessados



# O que é uma Configuração?

**Configuração de um sistema** é uma coleção de **versões específicas** de itens de configuração (hardware, firmware ou software) que são combinados de acordo com procedimentos específicos de construção para servir a uma finalidade particular.

# Outros conceitos

**Item de configuração:** Elemento unitário ou um grupo de elementos para efeito de controle de versão.

- Código
- Documentação
- Diagramas, planos, ferramentas, casos de teste e etc

Todos os documentos que podem ser úteis para a evolução futura do sistema

# Outros conceitos

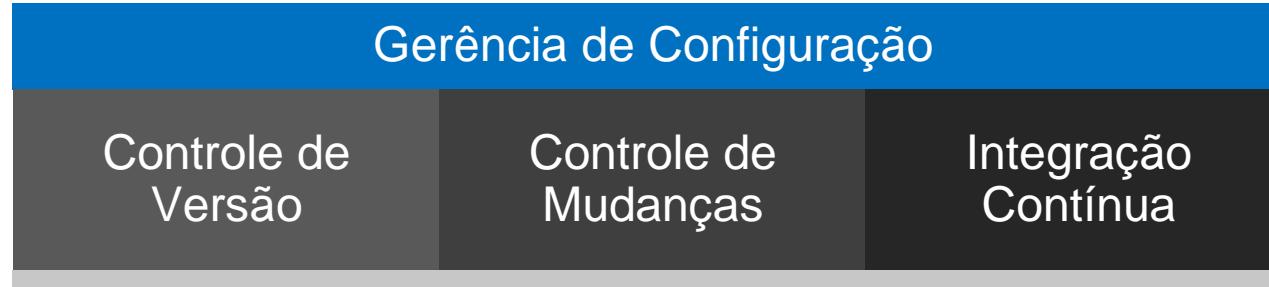
**Baseline:** configuração formalmente aprovada que servirá de referência para desenvolvimento posterior

A configuração do software em um ponto discreto no tempo

Quando um conjunto de artefatos de software se torna um item de configuração?



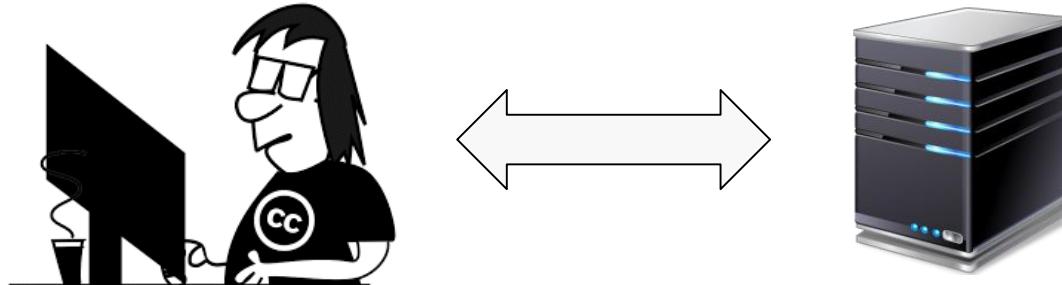
# Qual a importância do controle de versões?



# Problema exemplo

**Você precisa editar um site hospedado em um servidor:**

- 1) Você faz o download via FTP
- 2) Faz as alterações necessárias
- 3) Envia os arquivos alterados para o servidor via FTP



# Problema exemplo

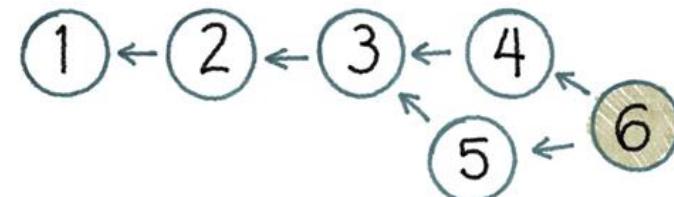
Agora outro desenvolvedor também deve fazer alterações no site...

- 1) Baixa o mesmo arquivo junto com você
- 2) Edita e manda para o servidor depois de você
- 3) Sobrescreve suas alterações!!!



# Controle de Versões

- Principais objetivos do Controle de Versões do Sistema [Sink, 2011]
  - Trabalhar simultaneamente
  - Evitar conflitos em alterações
  - Manter versões



# Controle de Mudanças

**Objetivo:** Garantir que todas as mudanças ocorram de maneira controlada e otimizada

- Manter histórico de mudança
- Justificar mudança



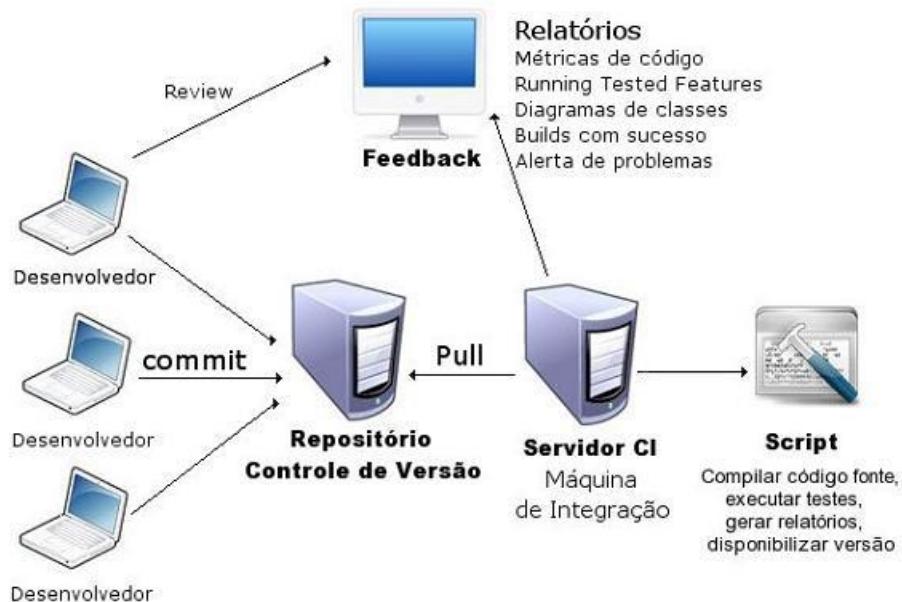
Exemplo:

- a) Mudanças da versão 2.2 para a versão 2.3:
  - correção do defeito D345
  - correção do defeito D346
  - adicionada a funcionalidade do RF44
- b) Pendências para uma versão posterior:
  - melhorar usabilidade da interface I43

# Integração contínua

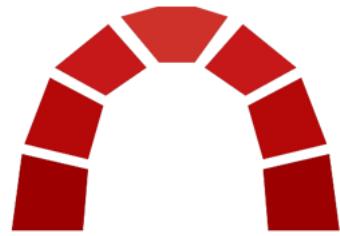
- Uma das práticas mais importantes do **desenvolvimento ágil!**
- Agilizar tarefas demoradas como a compilação de um projeto e a execução dos seus testes automatizados
- Tarefas são executadas a cada mudança no repositório de código e, em caso de erros de compilação ou falhas nos testes automatizados, todos os desenvolvedores são alertados rapidamente

# Integração contínua



# Ferramentas de apoio

Tipo de Ferramenta	Open Source	Comercial
Controle de Versão	Mercurial   <a href="#">Git</a>   Subversion   CVS	Team Foundation Server - Microsoft   Team Concert - IBM/Rational   ClearCase   StarTeam   Perforce   BitKeeper
Controle de Mudança	Trac   <a href="#">Redmine</a>   Mantis   Bugzilla	JIRA   FogBUGZ   CaliberRM   Perforce
Integração Contínua	<a href="#">Jenkins</a>   Bitten   Scons   Ant   Maven   CruiseControl   Gump   TinderBox	AntHill Pro   FinalBuilder   BuildForge



# REDMINE

flexible project management

# Visão geral do Redmine

- Um gerenciador de projetos com uma interface web
- Fornece várias ferramentas para gestão de tarefas, calendário e gráficos de *Gantt* para ajudar na representação visual dos prazos de entrega
- Desenvolvido com o framework **Ruby on Rails**

# Projeto disciplina Engenharia de Software Pos-2015

Busca:

Projeto disciplina Engenharia de Software Pos-2015

[Visão geral](#) [Atividade](#) [Tarefas](#) [Nova tarefa](#) [Gantt](#) [Calendário](#) [Notícias](#) [Documentos](#) [Wiki](#) [Arquivos](#) [Configurações](#)

## Visão geral

[Novo subprojeto](#)

### Tarefas

- Bug: 0 aberta / 0
- Feature: 5 abertas / 5
- Support: 0 aberta / 0

[Ver todas as tarefas](#) | [Calendário](#) | [Gantt](#)

### Membros

Manager: Diógenes Dias Simão, Dênis Leonardo Zaniro, Helder Jefferson Ferreira da Luz, Iohan Gonçalves Vargas, Nilton Mendes de Souza, Rosana Braga, Victor Hugo Santiago  
Reporter: Dênis Leonardo Zaniro, Iohan Gonçalves Vargas, Nilton Mendes de Souza, Victor Hugo Santiago

### Tempo gasto

20.00 horas

[Detalhes](#) | [Relatório](#)

Tarefas - Projeto disciplina X Victor Hugo

redmine.labes.icmc.usp.br/projects/proj\_es\_pos\_2015/issues

Apps RU IEEE Scopus ACM SBC Gmail Bolsa o/ Hot Twitter Facebook YouTube UFSCar DC EaD Translate 4shared C&A Outros favoritos

Página inicial Minha página Projetos Ajuda Acessando como: vctorsantiago Minha conta Sair

# Projeto disciplina Engenharia de Software Pos-2015

Busca: Projeto disciplina Engenharia de Software Pos-2015

Visão geral Atividade Tarefas Nova tarefa Gantt Calendário Notícias Documentos Wiki Arquivos Configurações

## Tarefas

↓ Filtros

Situação

Aberta ▾

Adicionar filtro ▾

↓ Opções

Aplicar

#	Tipo	Situação	Prioridade	Título	Atribuído para	Alterado em
121	Feature	New	Normal	Modelagem de Análise do RepRS		25/05/2015 09:30 hs
120	Feature	Resolved	High	Descrição dos Casos de Uso		24/05/2015 21:33 hs
119	Feature	Resolved	High	Diagrama de Casos de Uso		24/05/2015 13:53 hs
112	Feature	Resolved	High	Especificação do Requisitos		24/05/2015 21:42 hs
111	Feature	In Progress	High	Projeto ES-Pos - Repositório de Revisões Sistemáticas	Rosana Braga	25/05/2015 09:30 hs

(1-5/5) | Por página: 25, 50, 100

Exportar para Atom | CSV | PDF

## Tarefas

Ver todas as tarefas

Resumo

Calendário

Gantt

Projeto disciplina Engenharia Victor Hugo

redmine.labes.icmc.usp.br/projects/proj\_es\_pos\_2015/issues/new

Apps RU IEEE Scopus ACM SBC Gmail Bolsa o/ Hot Twitter Facebook YouTube UFSCar DC EaD Translate 4shared C&A Outros favoritos

## Nova tarefa

**Tipo \*** Bug

**Título \*** Bug

**Descrição**

**Situação \*** New

**Prioridade \*** Normal

**Atribuído para**

**Arquivos** Escolher arquivo Nenhum arquivo selecionado

Adicionar outro arquivo (Tamanho máximo: 5 MB)

**Observadores**

Diógenes Dias Simão Nilton Mendes de Souza Dênis Leonardo Zaniro Rosana Braga Helder Jefferson Ferreira da Luz Victor Hugo Santiago Iohan Gonçalves Vargas

Search for watchers to add

**Privado**

**Formatação do texto**

Criar Criar e continuar Pré-visualizar

Powered by Redmine © 2006-2012 Jean-Philippe Lang

Projeto disciplina Engenharia Victor Hugo

redmine.labes.icmc.usp.br/projects/proj\_es\_pos\_2015/issues/new

Apps RU IEEE Scopus ACM SBC Gmail Bolsa o/ Hot Twitter Facebook YouTube UFSCar DC EaD Translate 4shared C&A Outros favoritos

## Nova tarefa

**Tipo \*** Bug

**Título \***

**Descrição**

**Situação \*** New   
New  
In Progress  
Resolved  
Feedback  
Closed  
Rejected

**Prioridade \*** New   
In Progress  
Resolved  
Feedback  
Closed  
Rejected

**Atribuído para**

**Tarefa pai**   
**Inicio** 2016-03-05

**Data prevista**

**Tempo estimado**  Horas

**% Terminado** 0 %

**Arquivos**  Nenhum arquivo selecionado     Adicionar outro arquivo (Tamanho máximo: 5 MB)

**Observadores**  Diógenes Dias Simão  Dênis Leonardo Zaniro  Helder Jefferson Ferreira da Luz  Iohan Gonçalves Vargas  
 Nilton Mendes de Souza  Rosana Braga  Victor Hugo Santiago  
 Search for watchers to add

Powered by Redmine © 2006-2012 Jean-Philippe Lang

Projeto disciplina Engenharia X Victor Hugo

redmine.labes.icmc.usp.br/projects/proj\_es\_pos\_2015/issues/new

Apps RU IEEE Scopus ACM SBC Gmail Bolsa o/ Hot Twitter Facebook YouTube UFSCar DC EaD Translate 4shared C&A Outros favoritos

## Nova tarefa

**Tipo \*** Bug

**Título \***

**Descrição**

**Situação \*** New

**Prioridade \*** Normal

**Atribuído para**

- Low
- Normal
- High**
- Urgent
- Immediate

**Tarefa pai**

**Início** 2016-03-05

**Data prevista**

**Tempo estimado**  Horas

**% Terminado** 0 %

**Arquivos** Escolher arquivo Nenhum arquivo selecionado

Adicionar outro arquivo (Tamanho máximo: 5 MB)

**Observadores**

Diógenes Dias Simão  Dênis Leonardo Zaniro  Helder Jefferson Ferreira da Luz  Iohan Gonçalves Vargas

Nilton Mendes de Souza  Rosana Braga  Victor Hugo Santiago

Search for watchers to add

Powered by Redmine © 2006-2012 Jean-Philippe Lang

## Nova tarefa

Tipo \*  Privado

Título \*

Descrição

[Formatação do texto](#)Situação \* Tarefa pai Prioridade \* Início 

Atribuído para

&lt;&lt; mim &gt;&gt;

Diógenes Dias Simão

Dênis Leonardo Zaniro

Data prevista Tempo estimado  Horas% Terminado 

Arquivos

Helder Jefferson Ferreira da Luz

Iohan Gonçalves Vargas

Nilton Mendes de Souza

Rosana Braga

Victor Hugo Santiago

Descrição opcional 

Observadores

Dênis Leonardo Zaniro

Rosana Braga

 Helder Jefferson Ferreira da Luz Iohan Gonçalves Vargas Victor Hugo Santiago Search for watchers to add[Criar](#)[Criar e continuar](#)[Pré-visualizar](#)

# Projeto disciplina Engenharia de Software Pos-2015

Busca: Projeto disciplina Engenharia de Software Pos-2015

Visão geral Atividade Tarefas Nova tarefa Gantt Calendário Notícias Documentos Wiki Arquivos Configurações

## Gantt

▼ Filtros

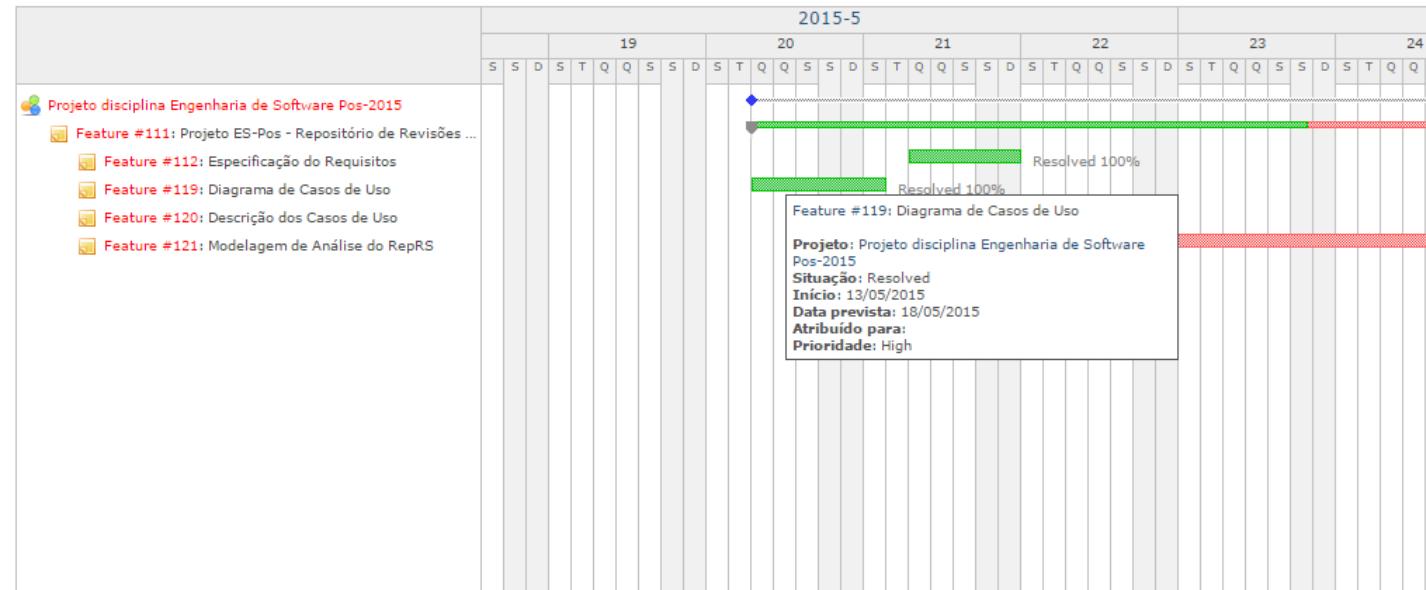
Situação

Aberta ▾

Adicionar filtro ▾

6 meses a partir de Maio 2015 ▾  

 Aproximar zoom  Afastar zoom



## Tarefas

Ver todas as tarefas

Resumo

Calendário

Gantt

## Feature #111

Atualizar Tempo de trabalho Observar Copiar Excluir

### Projeto ES-Pos - Repositório de Revisões Sistemáticas

« Anterior | 5/5 | Próximo »

Adicionado por Rosana Braga 10 meses atrás. Atualizado 10 meses atrás.

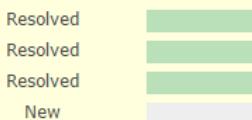
Situação:	In Progress	Início:	13/05/2015
Prioridade:	High	Data prevista:	14/06/2015
Atribuído para:	Rosana Braga	% Terminado:	<div style="width: 75%; background-color: #6aa84f;"></div> 75%
Categoria:	-	Tempo gasto:	20.00 horas
Versão:	-	Tempo estimado:	16.00 horas

Projeto-Parte1.pdf (260,7 KB) Rosana Braga, 25/05/2015 09:17 hs

### Subtarefas

[Adicionar](#)

- Feature #112: Especificação do Requisitos
- Feature #119: Diagrama de Casos de Uso
- Feature #120: Descrição dos Casos de Uso
- Feature #121: Modelagem de Análise do RepRS



### Tarefas relacionadas

[Adicionar](#)

### Histórico

Atualizado por Rosana Braga há 10 meses [#1](#)

- Projeto alterado de Disciplinas Rosana para Projeto disciplina Engenharia de Software Pos-2015

Atualizado por Rosana Braga há 10 meses [#2](#)

- Arquivo Projeto-Parte1.pdf adicionado

### Tarefas

[Ver todas as tarefas](#)

[Resumo](#)

[Calendário](#)

[Gantt](#)

### Observadores (0)

[Adicionar](#)

## Feature #112

Atualizar Tempo de trabalho Observar Copiar Excluir

Feature #111: Projeto ES-Pos - Repositório de Revisões Sistemáticas

« Anterior | 4/5 | Próximo »

### Especificação do Requisitos

Adicionado por Rosana Braga 10 meses atrás. Atualizado 10 meses atrás.

Situação: Resolved

Início: 20/05/2015

Prioridade: High

Data prevista: 24/05/2015

Atribuído para: -

% Terminado: 100%

Categoria: -

Tempo gasto: 7.00 horas

Versão: -

Tempo estimado: 7.00 horas

Descrição requisitos RS.xlsx (64,2 KB) Victor Hugo Santiago, 24/05/2015 21:36 hs

### Subtarefas

[Adicionar](#)

### Tarefas relacionadas

[Adicionar](#)

### Histórico

Atualizado por Victor Hugo Santiago há 10 meses

#1

- **Arquivo** Descrição requisitos RS.xlsx adicionado
- **Data prevista** ajustado para 24/05/2015
- **Situação** alterado de In Progress para Resolved
- **Início** ajustado para 20/05/2015
- **Tempo estimado** ajustado para 10.00

Link google docs: <https://docs.google.com/spreadsheets/d/188uqMdahZsN2xQoGvgY7AianM6VNi3wbWhMRCVYm28s/edit#gid=618936423>

Atualizado por Victor Hugo Santiago há 10 meses

#2

- **Tempo estimado** alterado de 10.00 para 7.00

### Tarefas

[Ver todas as tarefas](#)

[Resumo](#)

[Calendário](#)

[Gantt](#)

### Observadores (1)

Rosana Braga

[Adicionar](#)

# Projeto disciplina Engenharia de Software Pos-2015

Busca:  Projeto disciplina Engenharia de Software Pos-2015 ▾

Visão geral Atividade Tarefas Nova tarefa Gantt Calendário Notícias Documentos Wiki Arquivos Configurações

## Configurações

Informações Módulos Membros Versões Categorias das tarefas Wiki Repositórios Fóruns Atividades (registro de horas)

Nome \* Projeto disciplina Engenharia de Software Pos-2015

Sub-projeto de Disciplinas Rosana ▾

Descrição

Formatação do texto

Identificador \* proj\_es\_pos\_2015

Página do projeto

Público

### Tipos de tarefas

Bug

Feature

Support

# Projeto disciplina Engenharia de Software Pos-2015

Busca:  Projeto disciplina Engenharia de Software Pos-2015 ▾[Visão geral](#) [Atividade](#) [Tarefas](#) [Nova tarefa](#) [Gantt](#) [Calendário](#) [Notícias](#) [Documentos](#) [Wiki](#) [Arquivos](#) **Configurações**

## Configurações

[Informações](#) [Módulos](#) **Membros** [Versões](#) [Categorias das tarefas](#) [Wiki](#) [Repositórios](#) [Fóruns](#) [Atividades \(registro de horas\)](#)

Usuário / Grupo	Papéis	Editar	Excluir
Diógenes Dias Simão	Manager	Editar	Excluir
Dênis Leonardo Zaniro	Manager, Reporter	Editar	Excluir
Helder Jefferson Ferreira da Luz	Manager	Editar	Excluir
Iohan Gonçalves Vargas	Manager, Reporter	Editar	Excluir
Nilton Mendes de Souza	Manager, Reporter	Editar	Excluir
Rosana Braga	Manager	Editar	Excluir
Victor Hugo Santiago	Manager, Reporter	Editar	Excluir

### Novo membro

Pesquisar por usuários ou grupos:



- Cristiane Lana
- Daniel Feloni
- Domenico Schettini Filho
- Karen Rabelo
- Marcio Delamaro
- Marcos Pereira dos Santos
- Rafael Durelli
- Redmine Admin
- Silvana Morita
- Simone Souza
- Thiago Gottardi

**Papéis:**

- Manager
- Developer
- Reporter

**Adicionar**

Para praticar! acessem: <http://www.hostedredmine.com/projects/projeto-engsoft>

Página inicial Minha página Projetos Ajuda

Acessando como: **victorsantiago** Minha conta Sair

## Projeto Engenharia de Software

Busca:

Projeto Engenharia de Software ▾

[Visão geral](#) Atividade Tarefas Nova tarefa Gráficos Gantt Calendário Notícias Documentos Wiki Arquivos Configurações Delete Project

### Visão geral

Projeto de teste para fins didáticos com a turma SSC620 ENGENHARIA DE SOFTWARE

- Project Type: Client Collaboration

#### Tarefas

- Task: 0 aberta / 0
- Feature: 0 aberta / 0
- Bug: 0 aberta / 0
- Support: 0 aberta / 0

[Ver todas as tarefas](#) | [Calendário](#) | [Gantt](#)

Novo subprojeto Fechar

#### Membros

Manager: [Victor Hugo Santiago](#)

#### TEMPO GASTO

0.00 hora

[Tempo de trabalho](#) | [Detalhes](#) | [Relatório](#)

#### ADVERTISEMENTS



**Trouble  
Ticketing  
System**

Convert Emails to  
tickets; automate

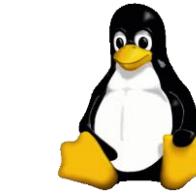


git

# Visão geral do Git e Github

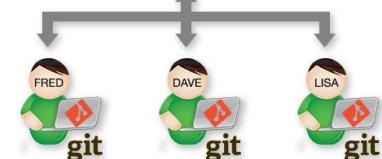
O que é Git?

- Sistema de Controle de Versão Distribuído
- Criado por Linus Torvalds (2005)
- Auxiliar no Desenvolvimento do Linux



O que é Github?

- Serviço de Web Hosting compartilhado para projetos que usam o controle de versionamento Git



## Companies & Projects Using Git

Google

facebook

Microsoft

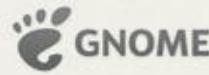
twitter

LinkedIn.

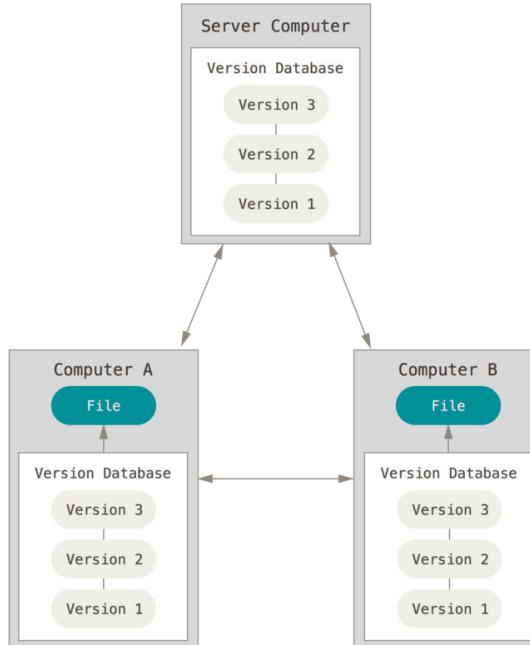
NETFLIX



PostgreSQL



# Controle de Versões Distribuído

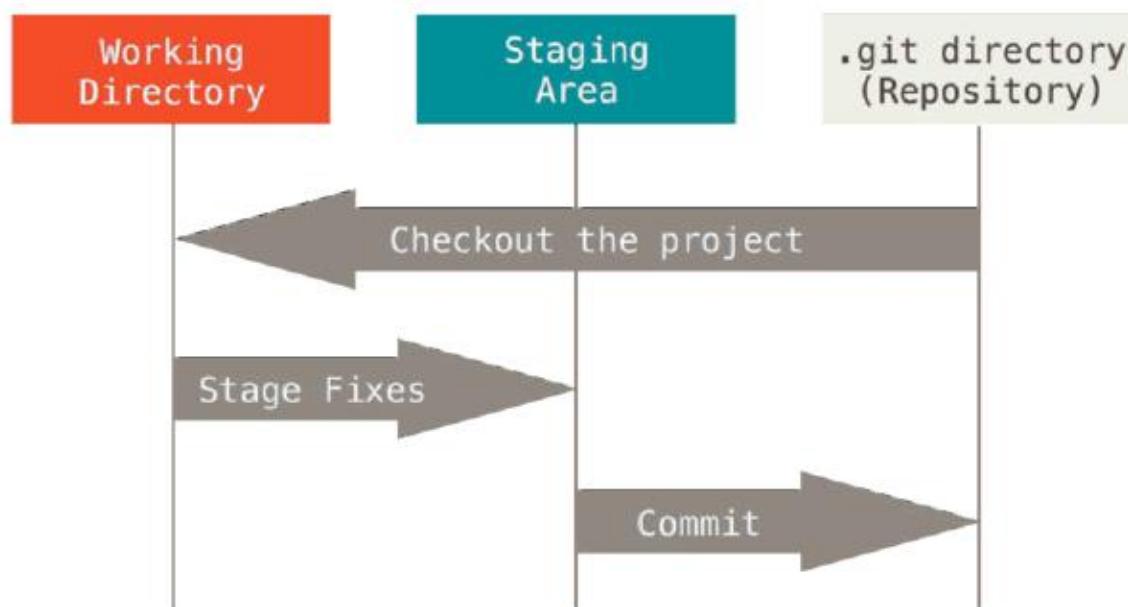


# Instalando Git

- Windows - <https://git-for-windows.github.io/>
- Linux - Package management tool (Ubuntu)  
*apt-get install git*
- Mac <http://sourceforge.net/projects/git-osx-installer/>



# Três estados do Git



# Configurações básicas

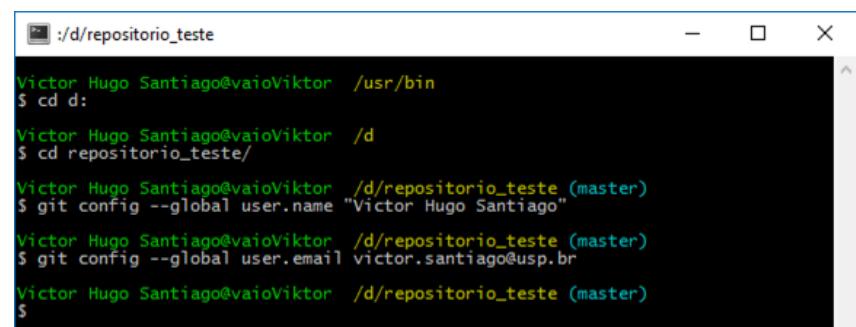
É importante nos identificarmos para o Git, execute os comandos a seguir no terminal com os seus dados:

```
git config -l
```

```
git config --global <variavel> <parâmetro>
```

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

```
git config --global user.email meu@email.com
```



A screenshot of a terminal window titled ':/d/repositorio\_teste'. The window shows a series of commands being run:

```
Victor Hugo Santiago@vaioViktor /usr/bin
$ cd d:
Victor Hugo Santiago@vaioViktor /d/repositorio_teste/
$ git config --global user.name "Victor Hugo Santiago"
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git config --global user.email victor.santiago@usp.br
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$
```

Precisou de ajuda?

```
git help
```

```
git help <parâmetro>
```

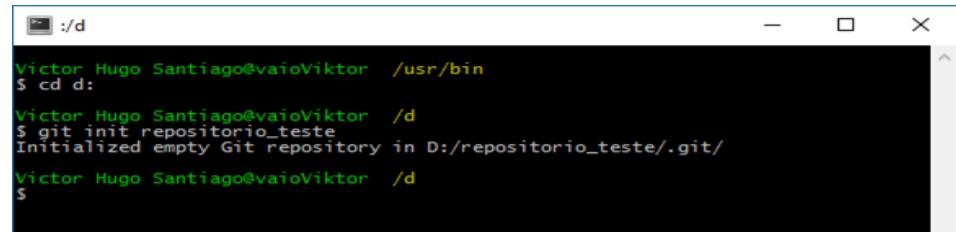
# Iniciando seu repositório com o Git

Para transformar um diretório local em um repositório Git, basta abrir o diretório via terminal e executar o comando:

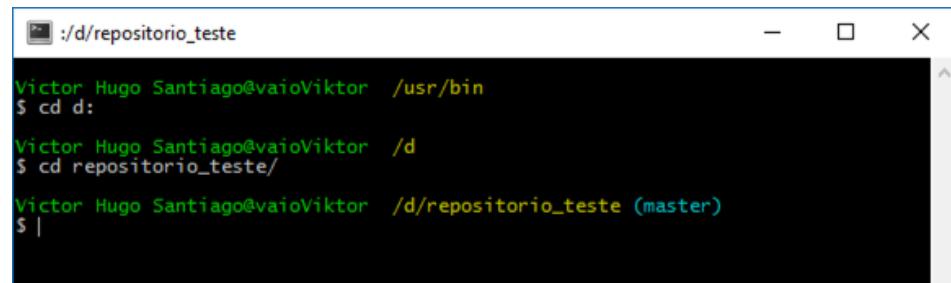
```
git init
```

ou

```
git init repositorio_teste
```



```
:/d
Victor Hugo Santiago@vaioViktor /usr/bin
$ cd d:
Victor Hugo Santiago@vaioViktor /d
$ git init repositorio_teste
Initialized empty Git repository in D:/repositorio_teste/.git/
Victor Hugo Santiago@vaioViktor /d
$
```

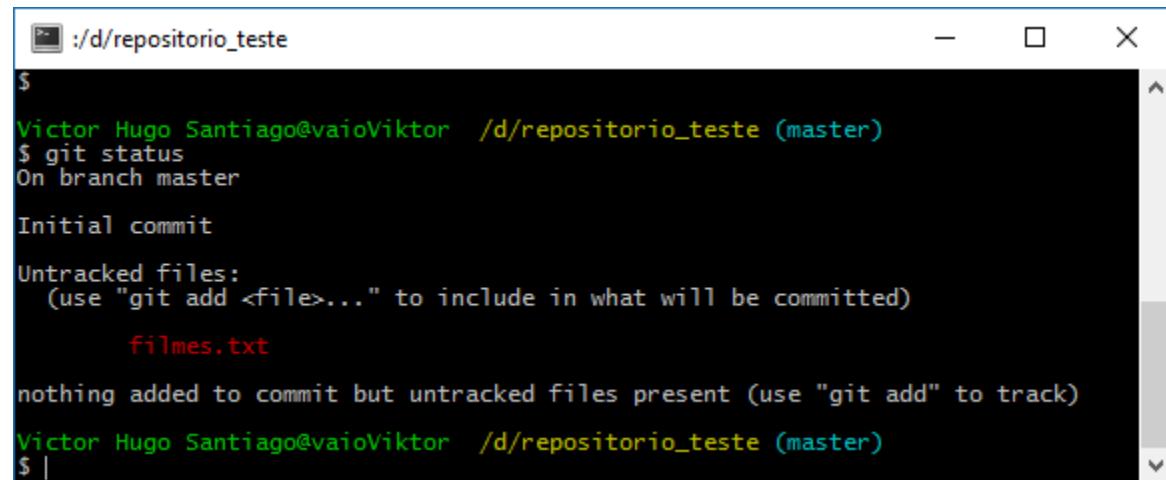


```
:/d/repositorio_teste
Victor Hugo Santiago@vaioViktor /usr/bin
$ cd d:
Victor Hugo Santiago@vaioViktor /d
$ cd repositorio_teste/
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ |
```

# Rastreando com Git

Supondo que adicionamos o arquivo filmes.txt no diretório “repositorio\_teste”

**git status**



```
./d/repositorio_teste
$ 
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    filmes.txt

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

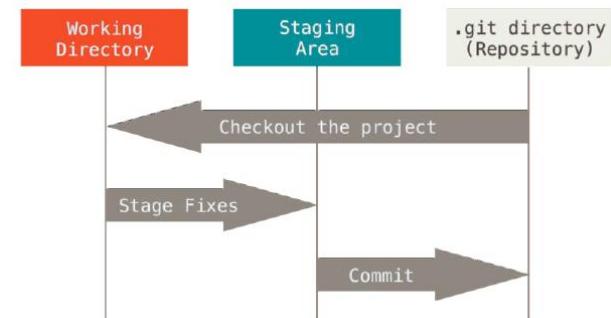
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ |
```

# Rastreando com Git

Para que um arquivo seja rastreado, devemos executar o seguinte comando:

```
git add filmes.txt ou  
git add . (para incluir todos os arquivos)
```

```
./d/repositorio_teste  
  
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)  
$ git add filmes.txt  
  
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)  
$ git status  
On branch master  
  
Initial commit  
  
Changes to be committed:  
(use "git rm --cached <file>..." to unstage)  
    new file:   filmes.txt  
  
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)  
$
```



# Gravando um arquivo no repositório (*commit*)

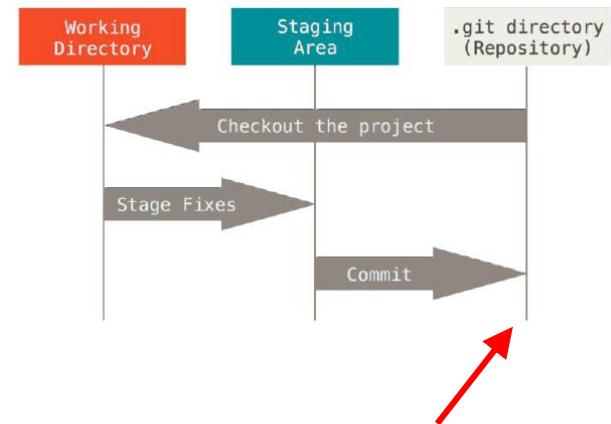
Para gravar as mudanças no repositório, execute o comando:

```
git      commit -m "Arquivo inicial de filmes para download"
```

```
./d/repositorio_teste
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git commit -m "Arquivo inicial de filmes para download"
[master (root-commit) 67aa6a7] Arquivo inicial de filmes para download
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 filmes.txt

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git status
On branch master
nothing to commit, working directory clean

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$
```



# Gravando um arquivo no repositório (*commit*)

Para visualizar o histórico de commits

**git log**

Para mostrar as alterações de um commit

**git show [commit]**

Commit deve ser especificado pela chave

```
./d/repositorio_teste
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git log
commit 67aa6a743b3159f3dac92b5fb6fd98185e702013
Author: Victor Hugo Santiago <victor.santiago@usp.br>
Date:   Sun Mar 6 18:55:45 2016 -0300

    Arquivo inicial de filmes para download

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$
```

```
commit 67aa6a743b3159f3dac92b5fb6fd98185e702013
Author: Victor Hugo Santiago <victor.santiago@usp.br>
Date:   Sun Mar 6 18:55:45 2016 -0300

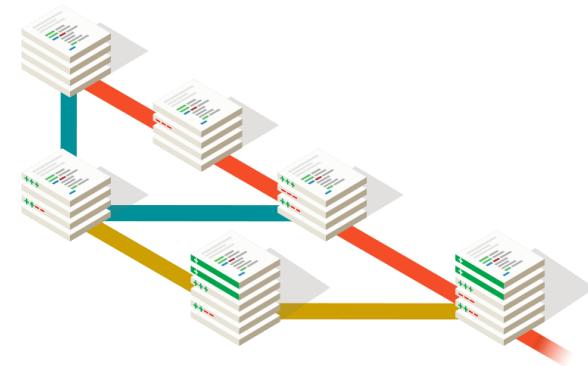
    Arquivo inicial de filmes para download

diff --git a/filmes.txt b/filmes.txt
new file mode 100644
index 0000000..e69de29
```

**Curiosidade:** o identificador do *commit* é um número de verificação de integridade de dados (*checksum*) criado a partir do conteúdo do arquivo usando a função de *hash* SHA-1

# Branching *(criando ramificações no repositório)*

- Branch é uma ramificação do repositório
- Alterações (*commits*) ocorrem na *branch*
- Muito útil para trabalhos colaborativos
- Branches de desenvolvimento facilitam o controle



Para listar as branches do nosso repositório, devemos executar o comando:

**git branch**

Por que \***master** ?

```
git branch
```



```
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git branch
* master

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ |
```

# Branching (*criando ramificações no repositório*)

Como listar todas as *branches*?

**git branch -v** (-r para branch remota, ver depois)

Como criar uma *branch*?

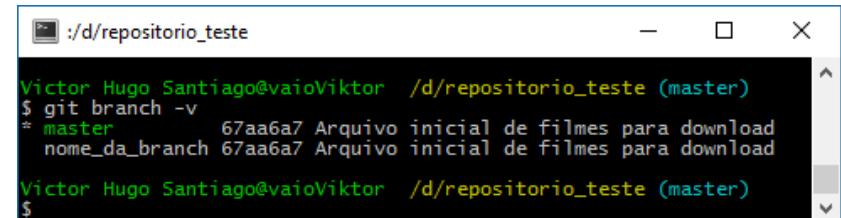
**git branch nome\_da\_branch**

Modo mais rápido de criar e deixá-la como corrente:

**git checkout -b nome\_da\_branch**

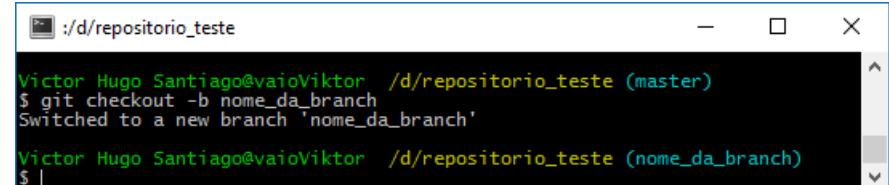
E se fosse para criar um *branch* remota? ([ver depois](#))

**git push origin nome\_da\_branch**



```
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git branch -v
* master 67aa6a7 Arquivo inicial de filmes para download
  nome_da_branch 67aa6a7 Arquivo inicial de filmes para download

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$
```



```
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git checkout -b nome_da_branch
Switched to a new branch 'nome_da_branch'

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (nome_da_branch)
$ |
```

# Apagando uma *branch*

Apagar uma *branch* em seu repositório local

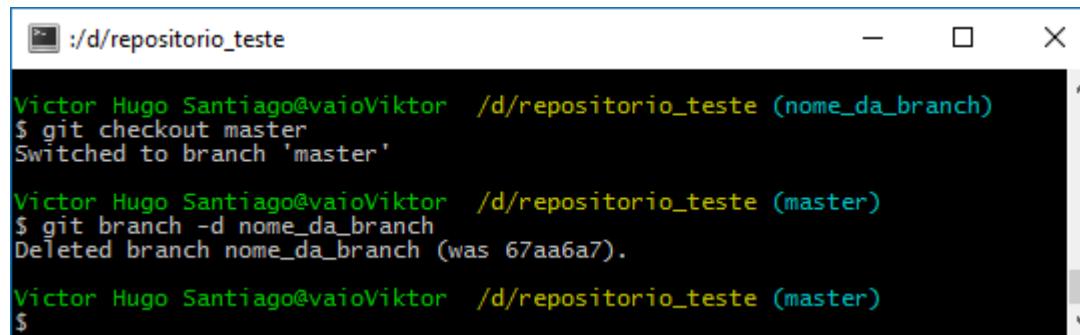
```
git branch -d nome_da_branch
```

Apagar uma *branch* remota ([ver depois](#))

```
git push origin --delete nome_da_branch
```

Como mudo de *branch*?

```
git checkout nome_da_branch
```



The screenshot shows a terminal window with the following session:

```
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (nome_da_branch)
$ git checkout master
Switched to branch 'master'

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git branch -d nome_da_branch
Deleted branch nome_da_branch (was 67aa6a7).

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$
```

# Mesclando *branches*

Supondo que estamos na branch *master*, podemos verificar as *branches* ainda não mescladas da seguinte forma:

**git branch --no-merged**

Temos uma branch com modificações em relação a *branch master*!

```
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (nome_da_branch)
$ git status
On branch nome_da_branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   filmes.txt

no changes added to commit (use "git add" and/or "git commit -a")

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (nome_da_branch)
$ git add filmes.txt

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (nome_da_branch)
$ git commit -m "Arquivo com lista de filmes modificado"
[nome_da_branch bc1f6b5] Arquivo com lista de filmes modificado
 1 file changed, 6 insertions(+)

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (nome_da_branch)
$ git checkout master
Switched to branch 'master'

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git branch --no-merged
  nome_da_branch

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ |
```

# Mesclando *branches*

Para esclarecer:

```
Terminator Genisys (2015)
Momentum (2015)
Inside Out (2015)
```

[filmes.txt \(master\)](#)

```
Bridge of Spies (2015)
Creed (2015)
Ex Machina (2015)
Secret in Their Eyes (2015)
Spotlight (2015)
The Martian (2015)
```

[filmes.txt \(nome\\_da\\_branch\)](#)

Queremos mesclar a branch “**nome\_da\_branch**” com “**master**”!

**Será que isso resultará em um conflito?**

# Mesclando *branches*

Para mesclar podemos usar um comando como esse:

```
git merge nome_da_branch -m "Mesclando com a branch master"
```

```
:/d/repositorio_teste

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (nome_da_branch)
$ git checkout master
Switched to branch 'master'

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git merge nome_da_branch -m "Mesclando com a branch master"
Auto-merging filmes.txt
CONFLICT (content): Merge conflict in filmes.txt
Automatic merge failed; fix conflicts and then commit the result.

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master|MERGING)
$ |
```



# Mesclando *branches*

Resultado:

```
filmes.txt - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
<<<<< HEAD
Terminator Genisys (2015)
Momentum (2015)
Inside Out (2015)
=====
Bridge of Spies (2015)
Creed (2015)
Ex Machina (2015)
Secret in Their Eyes (2015)
Spotlight (2015)
The Martian (2015)
>>>> nome_da_branch
```

Após alterações:

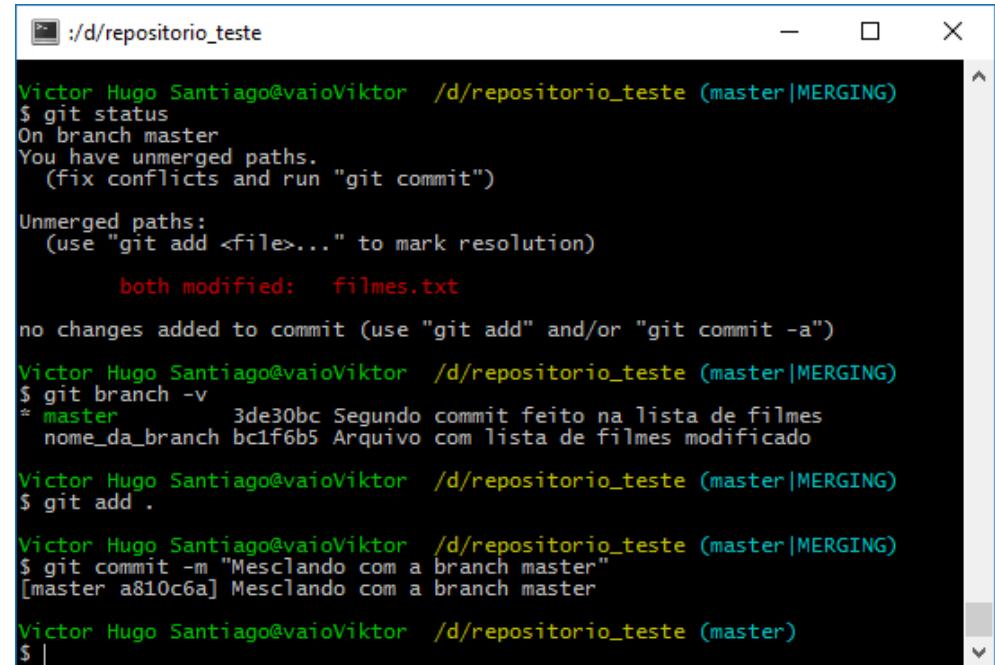
```
filmes.txt - Bloco de n...
Arquivo Editar Formatar Exibir Ajuda
Terminator Genisys (2015)
Momentum (2015)
Inside Out (2015)
Bridge of Spies (2015)
Creed (2015)
Ex Machina (2015)
Secret in Their Eyes (2015)
Spotlight (2015)
The Martian (2015)
```

# Resolvendo conflitos

Faça as alterações no arquivo mesclado e depois use os comando:

```
git add .
```

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



The screenshot shows a terminal window with the following session:

```
./d/repositorio_teste
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master|MERGING)
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:  filmes.txt

no changes added to commit (use "git add" and/or "git commit -a")

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master|MERGING)
$ git branch -v
* master 3de30bc Segundo commit feito na lista de filmes
  nome_da_branch bc1f6b5 Arquivo com lista de filmes modificado

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master|MERGING)
$ git add .

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master|MERGING)
$ git commit -m "Mesclando com a branch master"
[master a810c6a] Mesclando com a branch master

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ |
```

# Ignore arquivos com `.gitignore`

Basta criar um arquivo chamado `.gitignore` no diretório principal do nosso projeto com os nomes dos arquivos ou extensões que queremos ignorar.

Ex.:

```
*.log  
imagens/*.bmp
```

Em projetos Java, arquivos `.class`, `.jar` e `.war` são exemplos de arquivos que devem ser ignorados.

**Detalhe importante:** o `.gitignore` também envoluirá com o projeto.

**Assim, o que precisamos fazer?**

# Visualizar logs dos *commits*

git log

```
:/d/repositorio_teste
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git log
commit a810c6ac6961da052cfcc31a3c4a7a6f60298154
Merge: 3de30bc bc1f6b5
Author: Victor Hugo Santiago <victor.santiago@usp.br>
Date:   Sun Mar 6 19:44:57 2016 -0300

    Mesclando com a branch master

commit 3de30bc338e4403fad91e933f73ad58041b7c471
Author: Victor Hugo Santiago <victor.santiago@usp.br>
Date:   Sun Mar 6 19:32:34 2016 -0300

    Segundo commit feito na lista de filmes

commit bc1f6b5e1b80633cd6ac3b4b50393538b674897b
Author: Victor Hugo Santiago <victor.santiago@usp.br>
Date:   Sun Mar 6 19:27:37 2016 -0300

    Arquivo com lista de filmes modificado

commit 67aa6a743b3159f3dac92b5fb6fd98185e702013
Author: Victor Hugo Santiago <victor.santiago@usp.br>
Date:   Sun Mar 6 18:55:45 2016 -0300

    Arquivo inicial de filmes para download

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ |
```

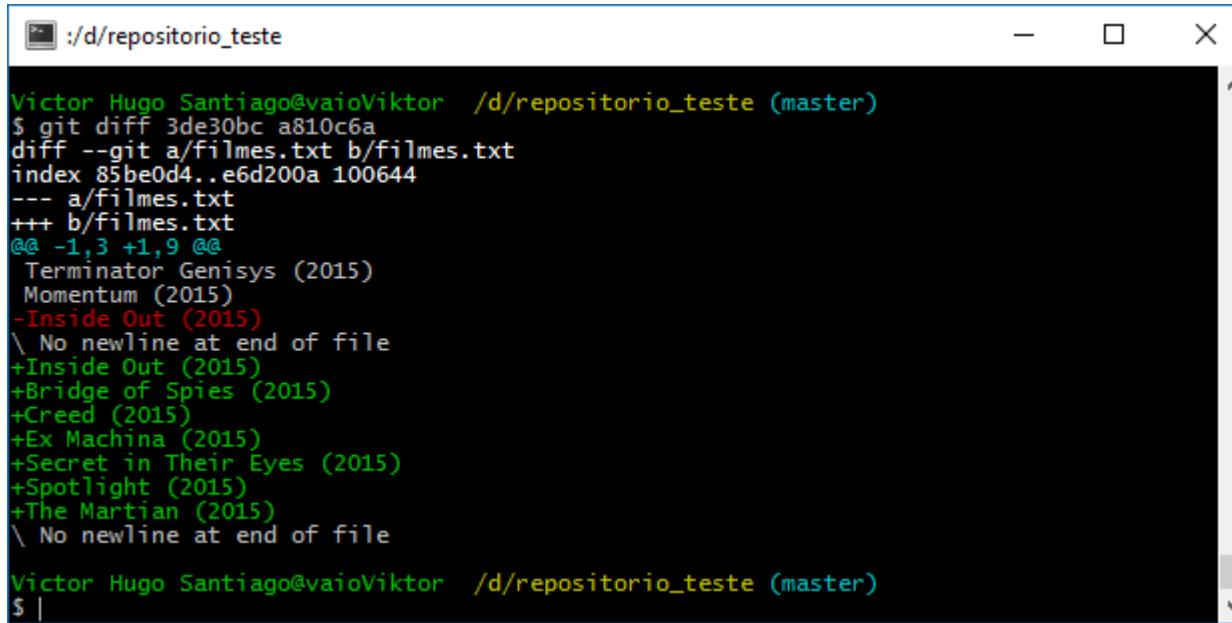
git log --graph --oneline

```
:/d/repositorio_teste
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git log --graph --oneline
* a810c6a Mesclando com a branch master
| \
| * bc1f6b5 Arquivo com lista de filmes modificado
| * 3de30bc Segundo commit feito na lista de filmes
| /
* 67aa6a7 Arquivo inicial de filmes para download

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ |
```

# Verificar diferenças entre commits

```
git diff 3de30bc a810c6a
```



The screenshot shows a terminal window with the following content:

```
:/d/repositorio_teste
Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ git diff 3de30bc a810c6a
diff --git a/filmes.txt b/filmes.txt
index 85be0d4..e6d200a 100644
--- a/filmes.txt
+++ b/filmes.txt
@@ -1,3 +1,9 @@
 Terminator Genisys (2015)
 Momentum (2015)
-Inside Out (2015)
 \ No newline at end of file
+Inside Out (2015)
+Bridge of Spies (2015)
+Creed (2015)
+Ex Machina (2015)
+Secret in Their Eyes (2015)
+Spotlight (2015)
+The Martian (2015)
 \ No newline at end of file

Victor Hugo Santiago@vaioViktor /d/repositorio_teste (master)
$ |
```

# Revendo conceitos



**github**  
SOCIAL CODING



# Remote

- Reppositório remoto, hospedado em um servidor
- São referenciados por uma URL
- Podem receber vários commits
- Sincronizar o trabalho colaborativo
- Exige chave SSH

```
git remote add origin <URL>
```

# Push

- Enviar alterações (*commits*) de uma *branch* para o repositório remoto
- A primeira vez:  
**git push -u origin master**
- O envio é rejeitado se o repositório local não estiver sincronizado

```
git push <remote> <branch>  
git push
```

# Pull

- Baixa as alterações do repositório remoto e realiza o **Merge** automático com o repositório local
- Mantém o repositório sincronizado com os últimos *commits* de uma *branch*
- Permite baixar novas *branches* que não foram criadas localmente

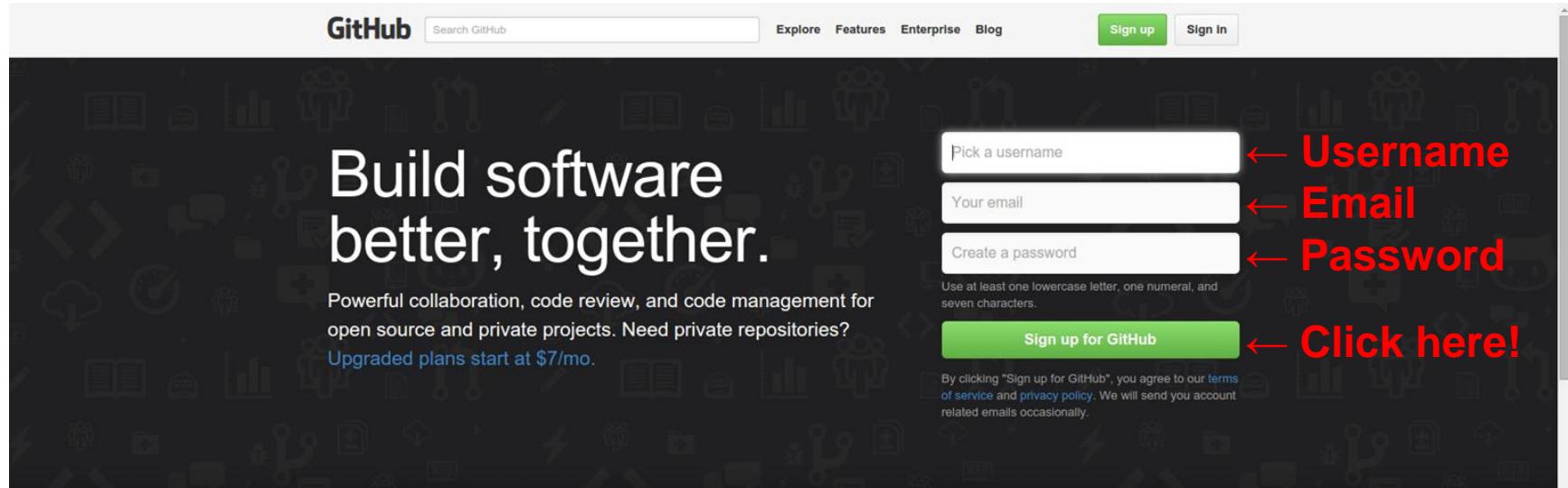
```
git pull  
git pull <remote> <nova>
```

# Git Clone

- Baixa o repositório remoto
- Outra forma de criar um repositório local
- Já vem com o remote configurado

```
git clone <URL>
```

# Inscrevendo-se no GitHub



<https://github.com>

# Inscrevendo-se no GitHub

## Welcome to GitHub

You've taken your first step into a larger world, @ [REDACTED]



### Choose your personal plan

Plan	Cost (view in BRL)	Private repos	Action
Large	\$50/month	50	<button>Choose</button>
Medium	\$22/month	20	<button>Choose</button>
Small	\$12/month	10	<button>Choose</button>
Micro	\$7/month	5	<button>Choose</button>
Free	\$0/month	0	<button>Chosen</button>

### Each plan includes:

- Unlimited collaborators
- Unlimited public repositories
- ✓ Free setup
- ✓ SSL Protection
- ✓ Email support
- ✓ Wikis, Issues, Pages, & more

Charges to your account will be made in US Dollars. Converted prices are provided as a convenience and are only an estimate based on current exchange rates. Local prices will change as the exchange rate fluctuates.

Don't worry, you can cancel or upgrade at any time.

Help me set up an organization next

Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.  
Learn more about organizations.

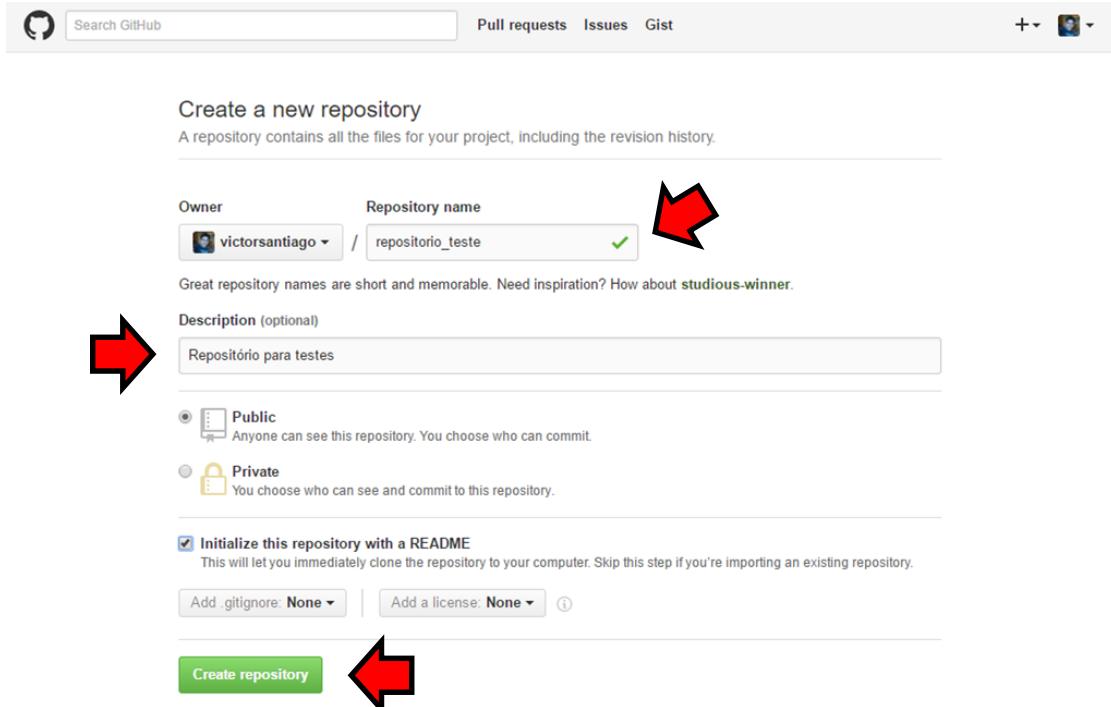
Finish sign up



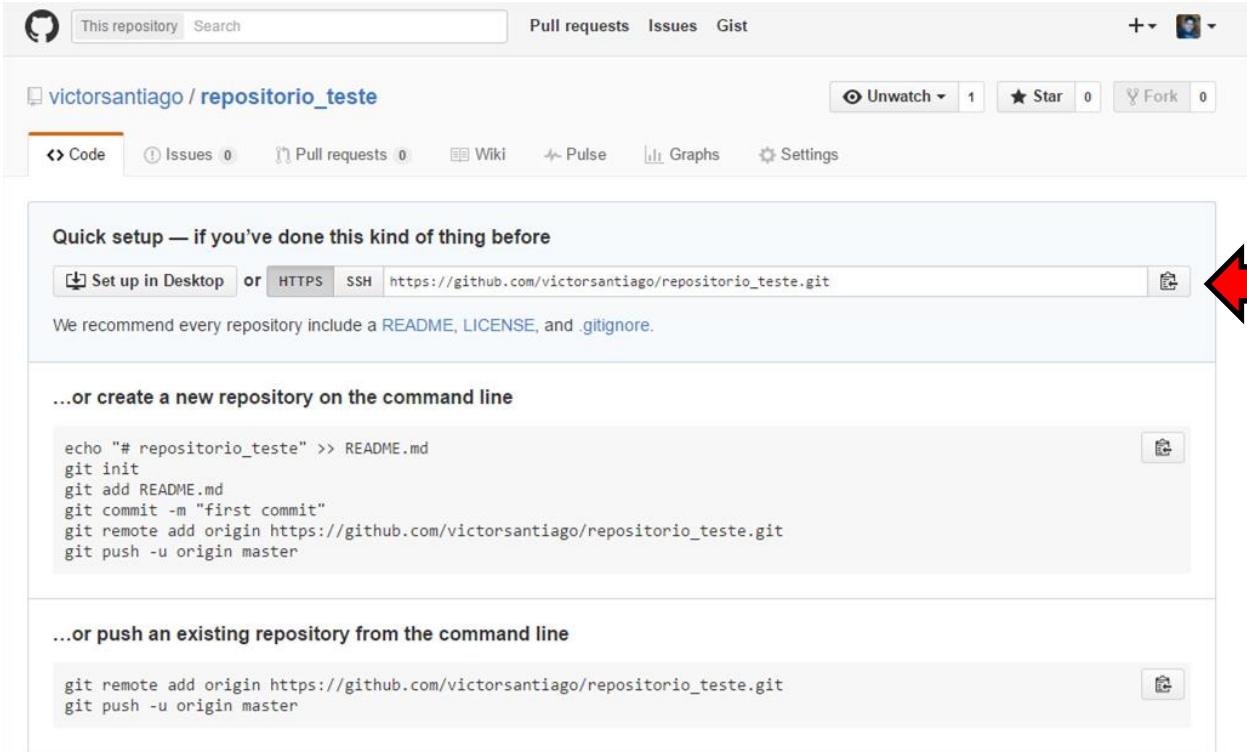
# Criando um repositório

The screenshot shows the GitHub homepage with a dark blue header. In the top right corner, there is a user profile icon for 'damascenodiego'. A red arrow points to this icon. Below the header, a message box displays: "Your repository 'damascenodiego/myFriendsSample' was successfully deleted." Another red arrow points to the close button of this message box. The main content area features the "GitHub Bootcamp" section with four cards: "Set up Git", "Create repositories", "Fork repositories", and "Work together". Below this, there are several notifications and a "Welcome to GitHub! What's next?" section. On the right side, there is a "Announcing Git Large File Storage (LFS)" announcement and a "View 140 new broadcasts" link. At the bottom right, there is a green button labeled "+ New repository" with a red arrow pointing to it. The footer contains a "ProTip!" link and navigation tabs for "All", "Public", "Private", "Sources", and "Forks".

# Criando um repositório



# Usando seu repositório do GitHub



The screenshot shows a GitHub repository page for 'victorsantiago / repositorio\_teste'. The page includes navigation links for 'Pull requests', 'Issues', and 'Gist'. Below the header, there are buttons for 'Unwatch' (with 1 follower), 'Star' (0 stars), and 'Fork' (0 forks). The main content area has tabs for 'Code', 'Issues (0)', 'Pull requests (0)', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. A 'Quick setup — if you've done this kind of thing before' section contains a 'Set up in Desktop' button, an 'HTTPS' button, an 'SSH' button, and a URL field containing 'https://github.com/victorsantiago/repositorio\_teste.git'. A red arrow labeled 'COPY' points to the URL field. Below this, instructions for creating a new repository on the command line are provided, along with a code block:

```
echo "# repositorio_teste" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/victorsantiago/repositorio_teste.git
git push -u origin master
```

Further down, instructions for pushing an existing repository from the command line are shown:

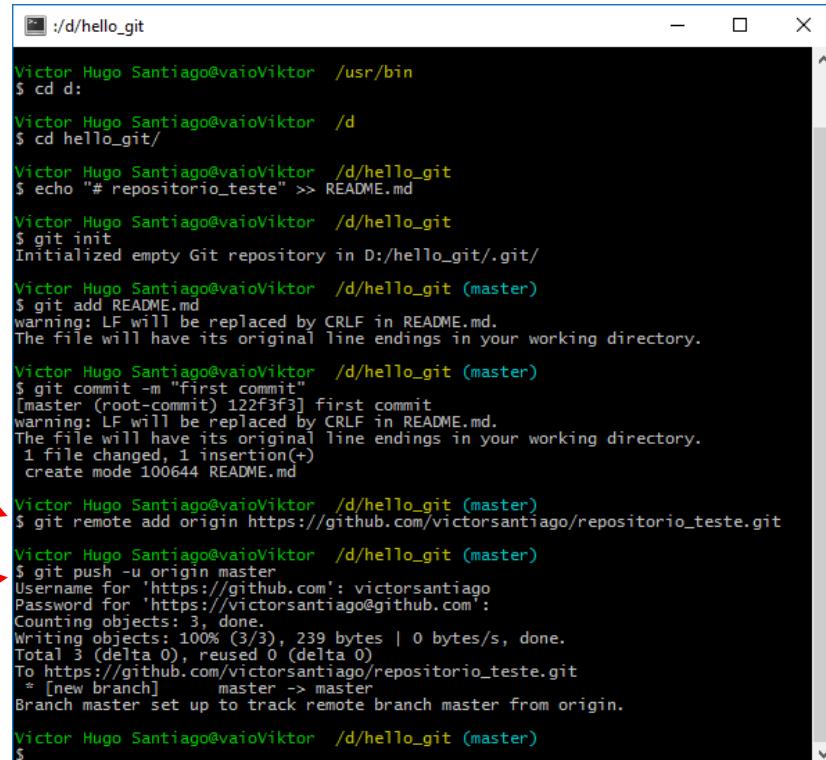
```
git remote add origin https://github.com/victorsantiago/repositorio_teste.git
git push -u origin master
```

# Usando seu repositório do GitHub

<https://github.com/<YOUR USERNAME>/helloGit.git>

**git remote add <repository name> <url>**

**git push -u <repository name> <branch name>**



```
:/d/hello_git
Victor Hugo Santiago@vaioViktor /usr/bin
$ cd d:
Victor Hugo Santiago@vaioViktor /d
$ cd hello_git/
Victor Hugo Santiago@vaioViktor /d/hello_git
$ echo "# repositorio teste" >> README.md
Victor Hugo Santiago@vaioViktor /d/hello_git
$ git init
Initialized empty Git repository in D:/hello_git/.git/
Victor Hugo Santiago@vaioViktor /d/hello_git (master)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.
Victor Hugo Santiago@vaioViktor /d/hello_git (master)
$ git commit -m "first commit"
[master (root-commit) 122f3f3] first commit
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
Victor Hugo Santiago@vaioViktor /d/hello_git (master)
$ git remote add origin https://github.com/victorsantiago/repositorio_teste.git
Victor Hugo Santiago@vaioViktor /d/hello_git (master)
$ git push -u origin master
Username for 'https://github.com': victorsantiago
Password for 'https://victorsantiago@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 239 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/victorsantiago/repositorio_teste.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
Victor Hugo Santiago@vaioViktor /d/hello_git (master)
$
```

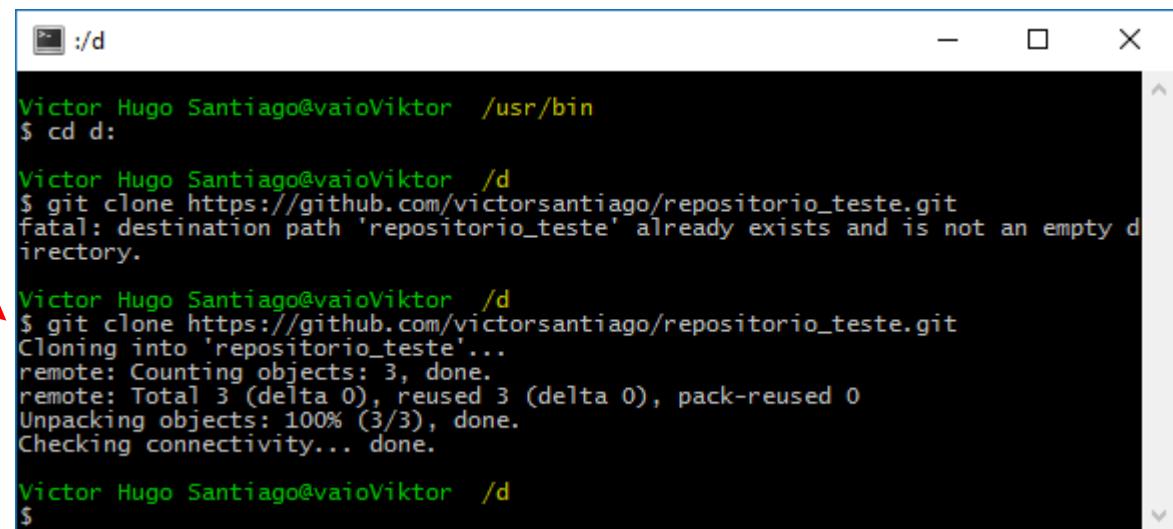
# Clonando seu repositório do GitHub (com Git Clone)

`https://github.com/<YOUR USERNAME>/repositorio_teste.git`

`git clone <url>`

Exibir o remote configurado:

`git remote -v`



The screenshot shows a terminal window with the following session:

```
Victor Hugo Santiago@vaioViktor /usr/bin
$ cd d:

Victor Hugo Santiago@vaioViktor /d
$ git clone https://github.com/victorsantiago/repositorio_teste.git
fatal: destination path 'repositorio_teste' already exists and is not an empty directory.

Victor Hugo Santiago@vaioViktor /d
$ git remote -v
origin  https://github.com/victorsantiago/repositorio_teste.git (fetch)
origin  https://github.com/victorsantiago/repositorio_teste.git (push)

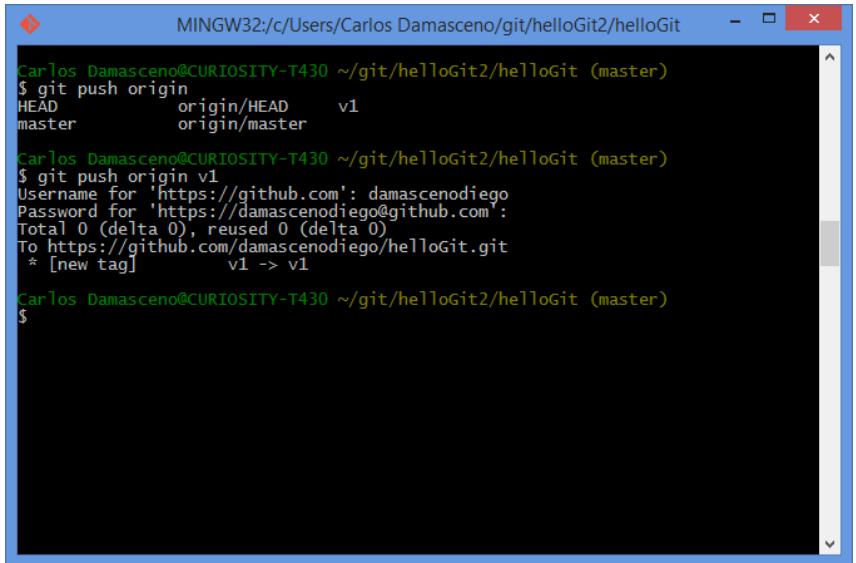
Victor Hugo Santiago@vaioViktor /d
$
```

# Usando seu repositório do GitHub

A screenshot of a GitHub repository page for 'victorsantiago / repositorio\_teste'. The page shows basic repository statistics: 1 commit, 1 branch, and 0 releases. Below the stats, there's a navigation bar with 'Branch: master' (with a dropdown), a green 'New pull request' button, and other options like 'New file', 'Upload files', 'Find file', and 'HTTPS'. To the right of the navigation bar is a URL 'https://github.com/victor...' followed by two small icons: a clipboard and a copy symbol, with a red arrow pointing to the copy icon. A large red arrow also points from the word 'COPY' to the same icon. The main content area displays a single commit from 'victorsantiago' with the message 'first commit'. At the bottom, there's a preview of the 'README.md' file containing the text 'repositorio\_teste'.

# Tagging

*git tag <tagname>*



```
MINGW32:/c/Users/Carlos Damasceno/git/helloGit2/helloGit - x
Carlos Damasceno@CURIOSITY-T430 ~ /git/helloGit2/helloGit (master)
$ git push origin v1
HEAD      origin/HEAD    v1
master    origin/master

Carlos Damasceno@CURIOSITY-T430 ~ /git/helloGit2/helloGit (master)
$ git push origin v1
Username for 'https://github.com': damascenodiego
Password for 'https://damascenodiego@github.com':
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/damascenodiego/helloGit.git
 * [new tag]      v1 -> v1

Carlos Damasceno@CURIOSITY-T430 ~ /git/helloGit2/helloGit (master)
$
```

# Tagging

damascenodiego / helloGit

Unwatch 1 Star 0 Fork 0

Releases Tags Draft a new release

an hour ago v1 ...  
9ef4699 zip tar.gz

The screenshot shows a GitHub repository page for 'damascenodiego / helloGit'. At the top, there's a dark blue header with the word 'Tagging' in white. Below the header, the repository name 'damascenodiego / helloGit' is displayed, along with 'Unwatch 1', 'Star 0', and 'Fork 0'. There are two tabs: 'Releases' (which is selected) and 'Tags'. A button for 'Draft a new release' is also visible. The main content area shows a single release named 'v1' which was created 'an hour ago'. It includes a commit hash '9ef4699' and download links for 'zip' and 'tar.gz'. To the right of the main content, there's a vertical sidebar with various icons: a copy icon, an info icon, a file icon, a chart icon, and a close icon.

Fork me on GitHub

# Forking

This repository Search

Explore Gist Blog Help

glcrashcourse + - ⌂ ⚙ ⚙

damascenodiego / reflections

Watch 1 Star 0 Fork 0

Code Issues Pull requests Wiki Pulse Graphs

3 commits 1 branch 0 releases 1 contributor

branch: master reflections / +

lesson 3 reflections file

damascenodiego authored on Mar 1 latest commit b7a6c11ce2

lesson01_reflections.txt	lesson reflection 01 and 02	2 months ago
lesson02_reflections.txt	lesson reflection 01 and 02	2 months ago
lesson03_reflections.txt	lesson 3 reflections file	2 months ago
reflections.txt	reflections text file	2 months ago

HTTPS clone URL <https://github.com>

You can clone with HTTPS, SSH, or Subversion.

Download ZIP

© 2015 GitHub, Inc. Terms Privacy Security Contact

Status API Training Shop Blog About

Fork me on GitHub

# Forking

Screenshot of a GitHub repository page for "gitcrashcourse / reflections". A red arrow points to the "Fork" button in the top right corner.

The repository details are as follows:

- Description: Short description of this repository
- Website: Website for this repository (optional)
- Save or Cancel
- 3 commits
- 1 branch
- 0 releases
- 1 contributor
- branch: master → reflections / +
- This branch is even with damascenodiego:master
- Pull Request Compare
- lesson 3 reflections file
- damascenodiego authored on Mar 1 latest commit b7a6c1ice2
- lesson01\_reflections.txt lesson reflection 01 and 02 2 months ago
- lesson02\_reflections.txt lesson reflection 01 and 02 2 months ago
- lesson03\_reflections.txt lesson 3 reflections file 2 months ago
- reflections.txt reflections text file 2 months ago

Help people interested in this repository understand your project by adding a README!

Add a README Download ZIP

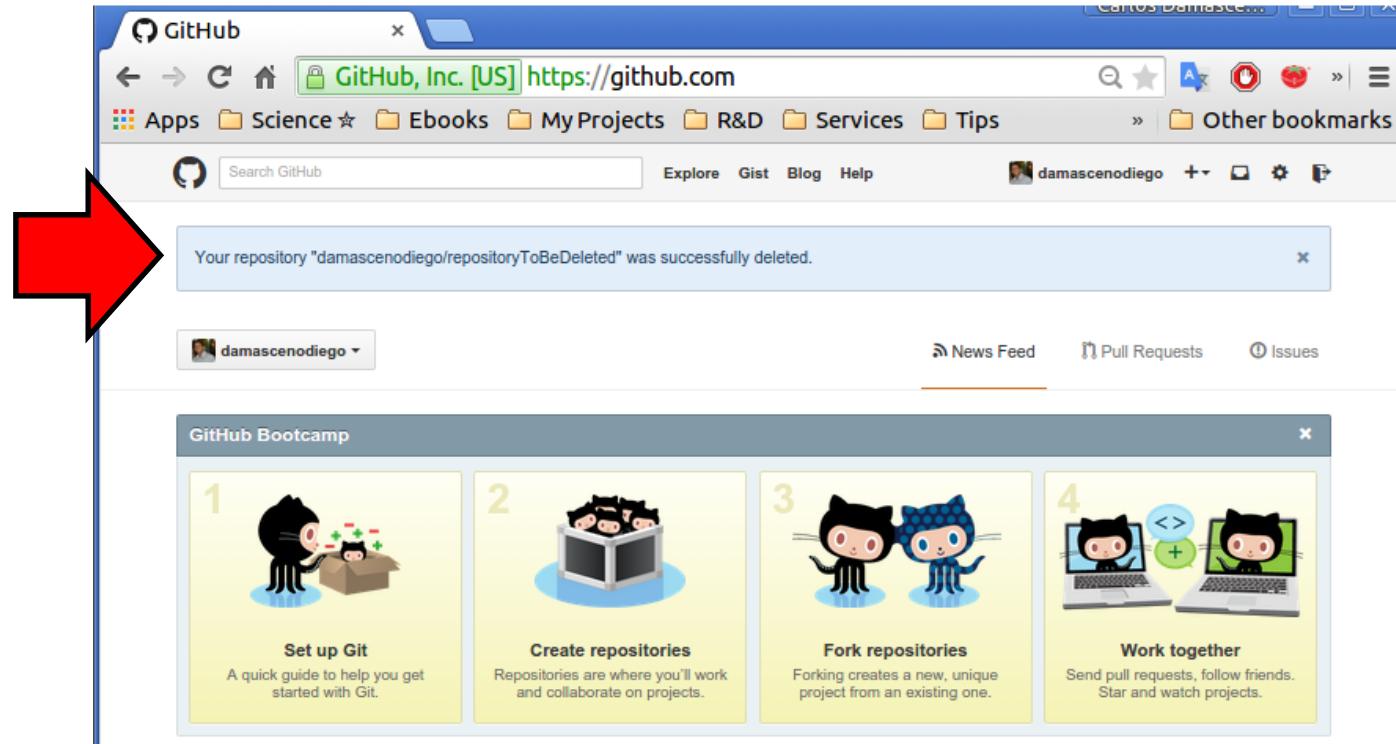
# Deletando um repositório

The screenshot shows a GitHub repository page for 'damascenodiego / repositoryToBeDeleted'. The page displays basic repository statistics: 1 commit, 1 branch, 0 releases, and 1 contributor. It lists an 'Initial commit' made by 'damascenodiego' 2 minutes ago. A prominent button at the bottom encourages adding a README. On the right side, there's a sidebar with links for 'Code', 'Issues', 'Pull requests', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The 'Settings' link is highlighted with a large, semi-transparent black box and a white cursor arrow pointing to it. Below the 'Settings' link, there's a section for cloning the repository via HTTPS, SSH, or Subversion, with a 'Download ZIP' button.

# Deletando um repositório



# Deletando um repositório



# Outras funcionalidade do GitHub

- Readme.md
- Wiki
- Web diff (código/imagens)
- Pull Request
- Comentários & Discussões
- Issue Tracking

# Readme.md

This screenshot shows a GitHub repository page for a user named damascenodiego. The repository is titled "egitTest". The main content is a README.md file containing the following text:

```
testing eclipse egit plugin — Edit
branches: master + egitTest / +
```

readme created  
damascenodiego authored 7 minutes ago latest commit b758ba34eb

HelloWorld big smile! :D a month ago

README.md readme created 7 minutes ago

## The largest heading (an h1 tag)

## The second largest heading (an h2 tag)

This is a blockquote

*This text will be italic*

**This text will be bold**

**This is bold and *this is italic*.**

- Item
- Item
- Item

1. Item 1
2. Item 2
3. Item 3

Check out this neat program I wrote:

```
x = 0
x = 2 + 2
what is x
```

Code Issues Pull requests Wiki Pulse Graphs Settings

HTTPS clone URL <https://github.com/damascenodiego/egitTest> Download ZIP

# Wiki

This screenshot shows a GitHub Wiki page for the repository `damascenodiego / egitTest`. The page title is `Home`. A message at the top states "Your Wiki was created." The main content area contains the text "Welcome to the egitTest wiki!". On the right side, there is a sidebar with sections for "Pages" (containing a link to "Home") and "Clone this wiki locally" (with a URL: <https://github.com/damascenodiego/egitTest>). The GitHub header includes links for Explore, Gist, Blog, Help, and the user profile damascenodiego.

This repository was created.

damascenodiego / `egitTest`

Unwatch 1 ⭐ Star 0 ⌂ Fork 0

## Home

Carlos Damasceno edited this page 2 minutes ago · 1 revision

Welcome to the `egitTest` wiki!

+ Add a custom footer

+ Add a custom sidebar

Pages 1

Home

Clone this wiki locally

<https://github.com/damascenodiego/egitTest>

# Web Diff (code)

The screenshot shows a GitHub commit page for a file named `myFriends.txt`. The commit was authored by `damascenodiego` 14 hours ago. It has 1 parent commit (`8398424`) and a commit hash of `62fd229f5c6e4ee33495750c87e9d4634cc9d23e`. The diff shows 1 addition and 1 deletion:

Line	Change Type	Text
1	-	Brian
2	+	Barbara
3	-	+Brian
4	-	Carlos
5	-	Diego
6	-	Jane
6	-	Rafael

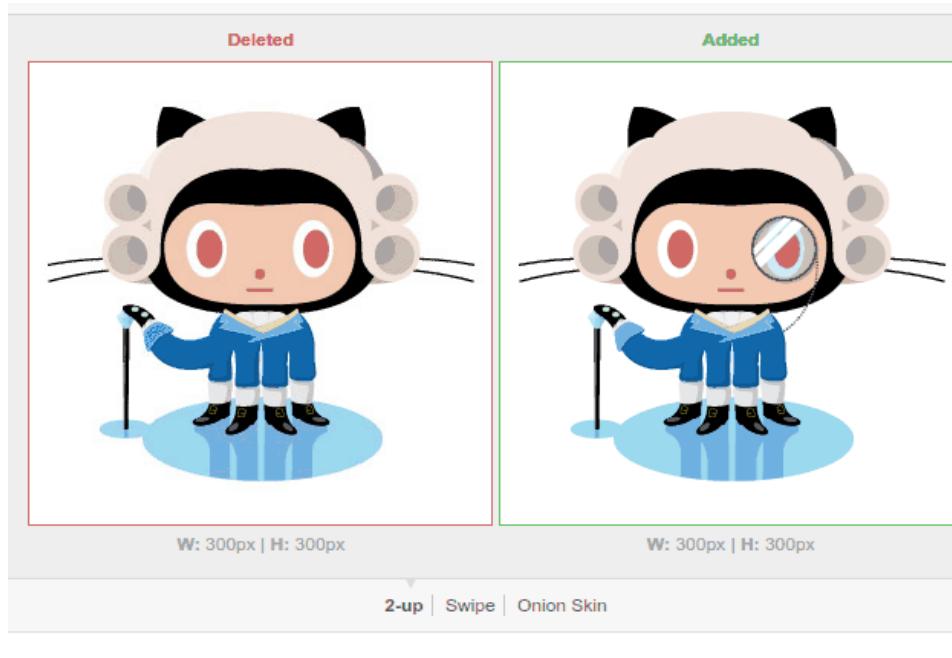
A comment form is displayed, allowing users to "Write" or "Preview" their comment. It includes a rich text editor with Markdown support and an "Edit in fullscreen" option. The text area contains the placeholder "Leave a comment". Below the text area, there is a note about attaching images and a "Comment on this commit" button.



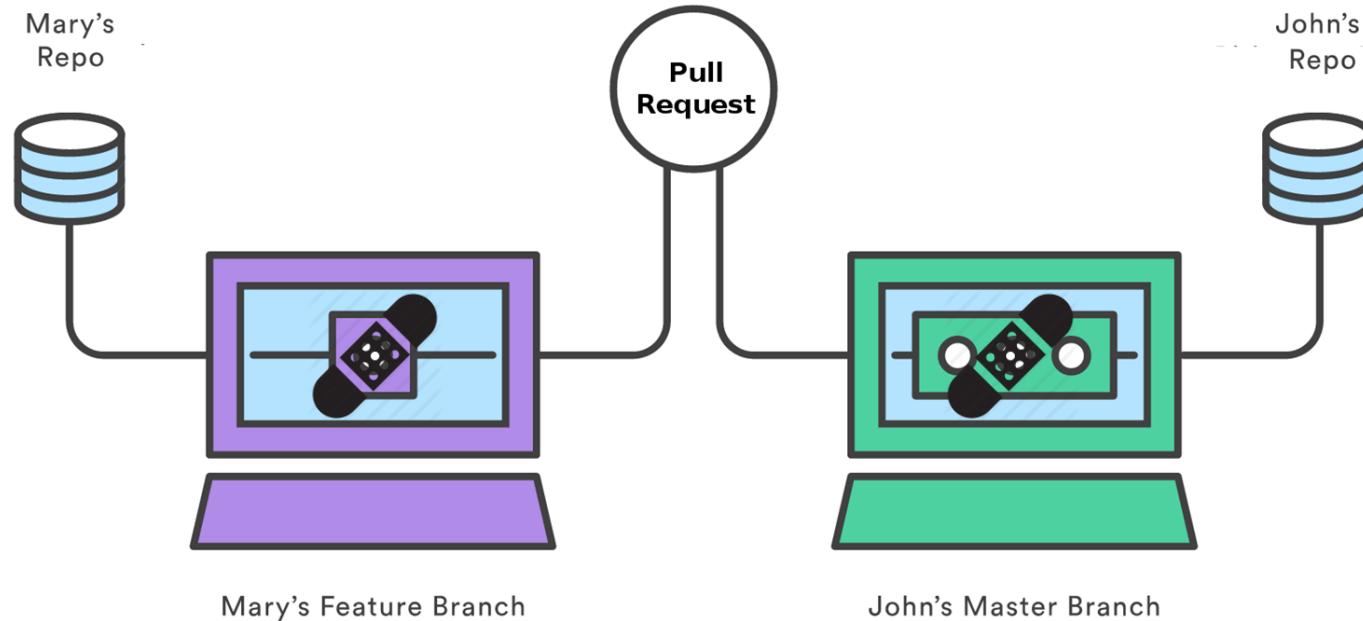
Unsubscribe

You're receiving notifications because you're subscribed to this repository.

# Web Diff (Image)



# Pull Request



# Comentários e Discussões

The screenshot shows a GitHub pull request page for the 'rails/rails' repository. The pull request is titled 'Fix grammar/style in attribute methods docs #19888'. It has 4 commits and was opened by 'imtayadeway' 15 minutes ago. The commit messages are:

- Fix grammar/style: pluralize 'each of its method'
- Fix grammar/style: break up long sentence
- Fix grammar/style: use (v) fall back (on)
- Fix grammar/style: assigns/declares -> assignments/declarations

The pull request can be automatically merged by project collaborators. A comment section is visible at the bottom.

GitHub interface elements include: Search bar, Explore, Gist, Blog, Help, User profile (damascenoediego), Watch (1,871), Star (25,822), Fork (10,225), Conversation (8), Commits (4), Files changed (8), Labels (None yet), Milestone (No milestone), Assignee (No one assigned), Notifications (Subscribe), and a comment input field with a 'Comment' button.

# Issue Tracking

The screenshot shows the GitHub Issues page for the `rails/rails` repository. The page has a dark theme. At the top, there are navigation links for Explore, Gist, Blog, and Help. On the right, there are icons for watching the repository, starring it (1.871 stars), and forks (10,225 forks). The main area shows a list of open issues. The first few issues listed are:

- ① Regression - scope of a relation is applied to unrelated object [#11982](#) opened 18 hours ago by phatt guy addressed
- ② Rails 4.2 Regression: before\_configuration hook is run too late [#11980](#) opened 17 hours ago by laserf0rm settled
- ③ Please do check if group value is not empty [#11973](#) opened 4 days ago by paul-peter addressed
- ④ Weird handling of nested association error messages when assigned to :base attribute [#11963](#) opened 2 days ago by ramredenct settled
- ⑤ ActiveJob: Improve Error Message for Unpersisted Models [#11961](#) opened 2 days ago by fmg addressed
- ⑥ [PERF] Regression in 4.2+ when creating/updating records [#11958](#) opened 2 days ago by igorwod settled
- ⑦ All Helpers Will Not Load in Rails 4.2.x (Virtual Box Development Environment) [#11950](#) opened 3 days ago by rokrents1000 needs feedback
- ⑧ place holder previews for NullMail [#11949](#) opened 3 days ago by manuallt addressed
- ⑨ Asset pipeline docs refer to no-longer-working config.assets.cache\_store [#11935](#) opened 4 days ago by jonsweat fixed
- ⑩ Slash in file name for render does not search in the application folder [#11934](#) opened 4 days ago by BerndJorgensO addressed
- ⑪ Check enum values for name 'parents' [#11916](#) opened 5 days ago by byteg addressed
- ⑫ Fixture should support 'yaml' or give a warning when you use foo.yaml. [#11905](#) opened 6 days ago by change addressed
- ⑬ MemCacheStore#increment bug [#11903](#) opened 7 days ago by AlexParamonov needs feedback
- ⑭ Create table fails on db:migrate task in ActiveRecord 4.2.1 if database adapter not support foreign\_keys. [#11898](#) opened 7 days ago by jonsweat addressed

# Vamos praticar (parte 1 - local)

#1 Instale o git na sua máquina

#2 Crie um diretório chamado **aprendendo\_git** e entre nele via terminal

#3 Use **git config** e informe seu nome e email

#4 **git init**

#5 **git status**

#6 Crie um arquivo **anotacoes.txt** nesse diretório, inclua seu nome nesse arquivo

#7 **git add** anotacoes.txt

#8 **git commit** -m “meu primeiro commit”

#9 **git status**

# Vamos praticar (parte 2 - local)

#1 Crie uma **branch** e chame de **feedback** (`git checkout -b feedback`)

#2 Abra o arquivo **anotacoes.txt** e veja se contém seu nome

#3 Altere o **anotacoes.txt** (nessa *branch*), apague seu nome e digite o que achou da aula ;D

#4 `git status`

#5 `git add anotacoes.txt`

#6 `git commit -m` “meu commit na branch feedback”

#7 Mude para a **branch master** e faça o **merge** com a *branch feedback*

#8 **Resolva o conflito** (edite o arquivo *anotacoes.txt*)

#9 `git add anotacoes.txt`

#10 `git commit -m` “resolvendo conflitos”

#11 delete a *branch feedback*

# Vamos praticar (parte 1 - remoto)

#1 Crie uma conta no **gitHub** e um repositório chamado **disciplina\_es**

#2 Configure um diretório na sua máquina com o remote, considerando os dados do seu repositório (pode usar o **git clone**).

#3 No diretório local: disciplina\_es (na branch master) crie um arquivo chamado **projeto.txt**

#4 Escreva: <seu nome> está **aprendendo gitHub para o projeto** (feche o editor)

#5 **git status**

#6 Use **git add .** e **git commit -m** “meu primeiro commit”

#7 **git status**

#8 Faça um **push** para seu repositório remoto: **git push <remote> <branch>**.

#9 Entre com seus dados: usuário e senha

#10 Abra o repositório no **gitHub** e veja as alterações

# Vamos praticar (parte 2 - remoto)

#1 Abra o repositório no gitHub -> Settings -> Collaborators

#2 Adicione o nome de um colega (conta github) e passe o seu para que ele faça o mesmo

Repita as atividades (parte 1 - remoto), configurando um **novo diretório** com os dados do remote do seu colega. Pode usar **git clone**, novamente.

#3 Faça um **pull**

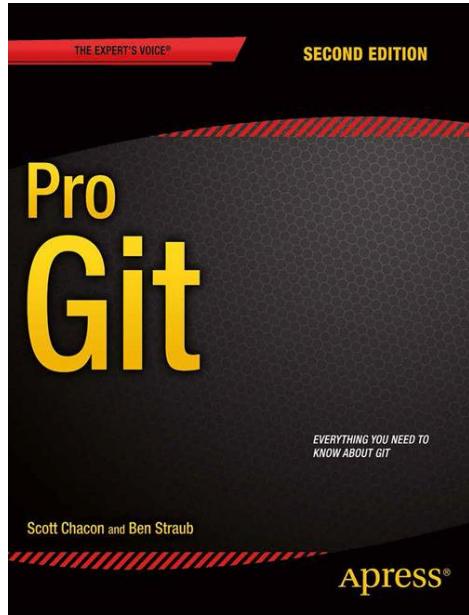
#4 Abra o arquivo projeto.txt do seu colega

#4 Adicione uma linha abaixo do texto dele: Também estou aprendendo gitHub, <seu nome>

#5 **git add .** e depois **git commit -m** “meu commit no repo remoto”

#5 Faça um **push** para o repositório do seu colega

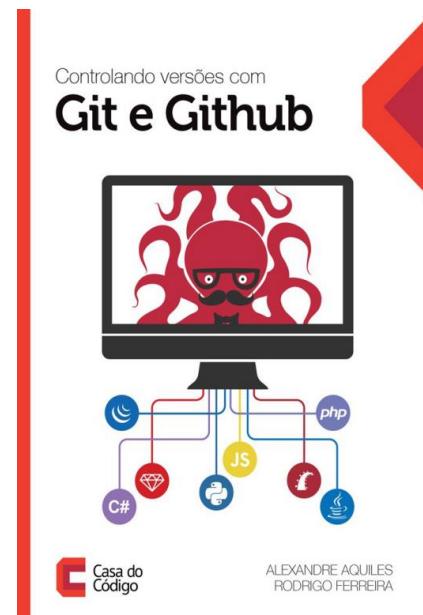
# Leituras



Scott Chacon e Ben Straub. 2014. *Pro Git* (2<sup>a</sup> ed.). Apress, Berkely, CA, USA.



Eric Sink. 2011. *Version Control by Example* (1<sup>a</sup> ed.). PYOW Sports Marketing.



Alexandre Aquile e Rodrigo Ferreira. 2016. (1<sup>a</sup> ed.). Casa do Código, São Paulo, SP

# Para dominar o Git!

## tryGit

<https://try.github.io>

## Guia prático

[http://rogerdudler.github.io/git-guide/index.pt\\_BR.html](http://rogerdudler.github.io/git-guide/index.pt_BR.html)

## Hostedredmine

<https://www.hostedredmine.com/>

The screenshot shows a web browser window titled "Code School - Try Git". The URL in the address bar is <https://try.github.io/levels/1/challenges/1>. The page content is a guide titled "1.1 · Got 15 minutes and want to learn Git?". It explains that Git is a distributed version control system allowing multiple people to work on the same documents simultaneously. Below this text is a terminal window with a blue header bar containing the command `git init`. The terminal itself has a dark background and displays the instruction "Press enter to submit commands" followed by a single greater-than sign (>). To the right of the terminal, there is a "tryGit" logo featuring a black cat head with a smiling mouth and whiskers, positioned next to the text "tryGit". The overall theme of the screenshot is a user-friendly introduction to Git through an interactive web-based environment.

# Referências

- Apresentações base
  - <https://github.com/damascenodiego/donationGame>, Diego Damasceno, LABES, ICMC-USP São Carlos
  - Minicurso: Git e contribuição para projetos Open Source, Bruno Orlandi - PET Computação, ICMC-USP São Carlos
- Livros
  - Eric Sink. 2011. **Version Control by Example** (1st ed.). PYOW Sports Marketing. ([FREE EBOOK](#))
  - Scott Chacon e Ben Straub. 2014. **Pro Git** (2nd ed.). Apress, Berkely, CA, USA. ([FREE EBOOK](#))
  - Pressman, R. S. **Software engineering: A practitioner's approach**. 5th ed. McGraw-Hill Higher Education, 2001.
  - Kemper C., Oxley, I. 2012. **Foundation Version Control for Web Developers**. Appress.
- Videos/Courses
  - **Learn Git in 20 Minutes:** [https://www.youtube.com/watch?v=Y9XZQO1n\\_7c](https://www.youtube.com/watch?v=Y9XZQO1n_7c)
  - **The Basics of Git and GitHub:** <https://www.youtube.com/watch?v=U8GBXvdmHT4>
  - **How to Use Git and GitHub:** <https://www.udacity.com/course/how-to-use-git-and-github--ud775>
- Web pages
  - <https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+101>
  - <https://sourceforge.net/p/forge/documentation/Git/>
  - <https://training.github.com/>
  - <https://www.atlassian.com/git/tutorials/>
  - <http://www.vogella.com/tutorials/EclipseGit/article.html>