



Tópicos Fundamentais

Prof.^a Catarina Costa

Agenda

01 GC

02 Git

"FINAL".doc



FINAL.doc!



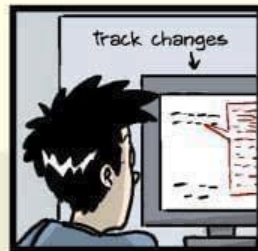
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRADSCHOOL????.doc

GC

Já perdeu uma versão anterior do arquivo do projeto e precisou dela?

Alguém já modificou indevidamente um arquivo e o original não poderia ter sido perdido?

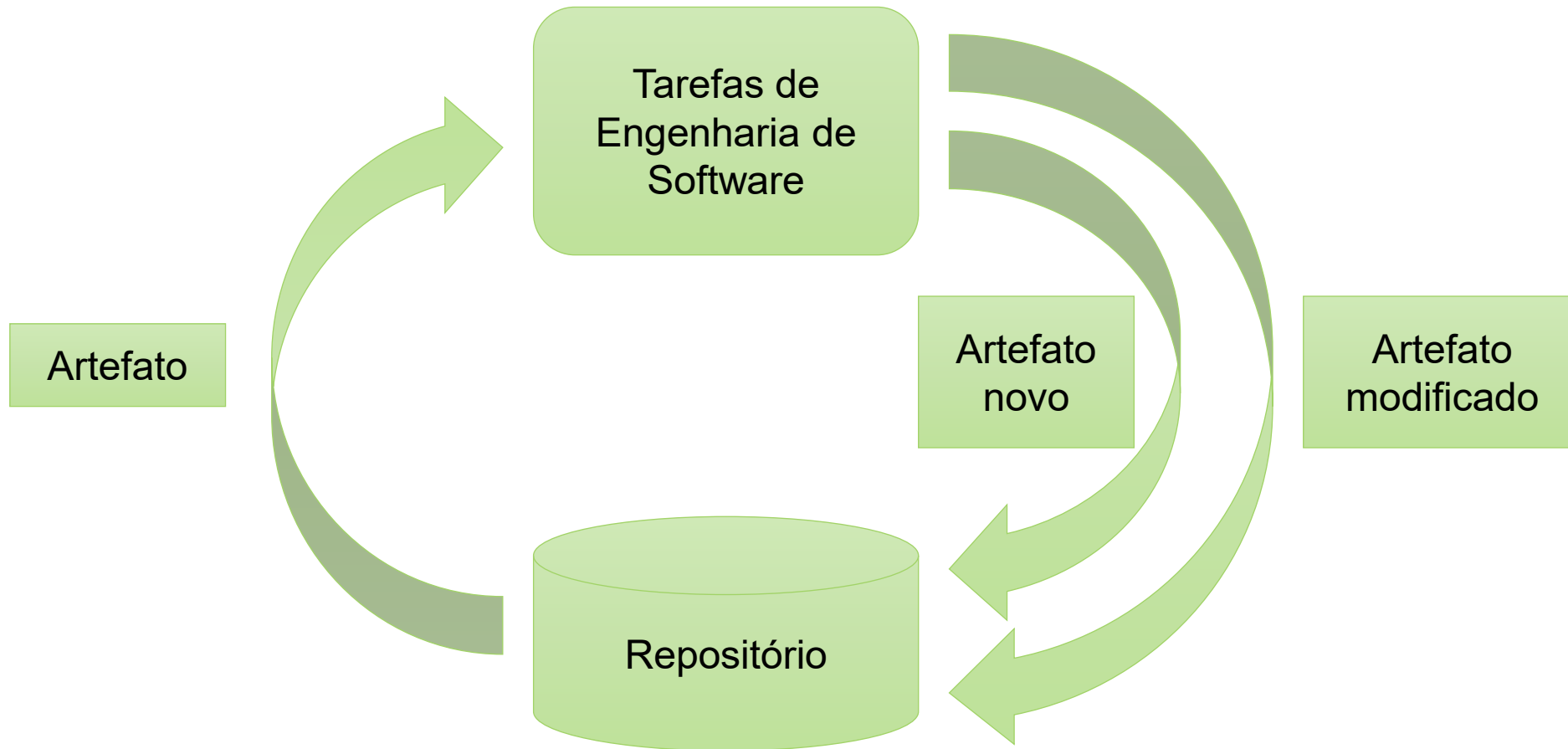
Tem dificuldade em saber quem modificou o que em um projeto?

É sobre isso...

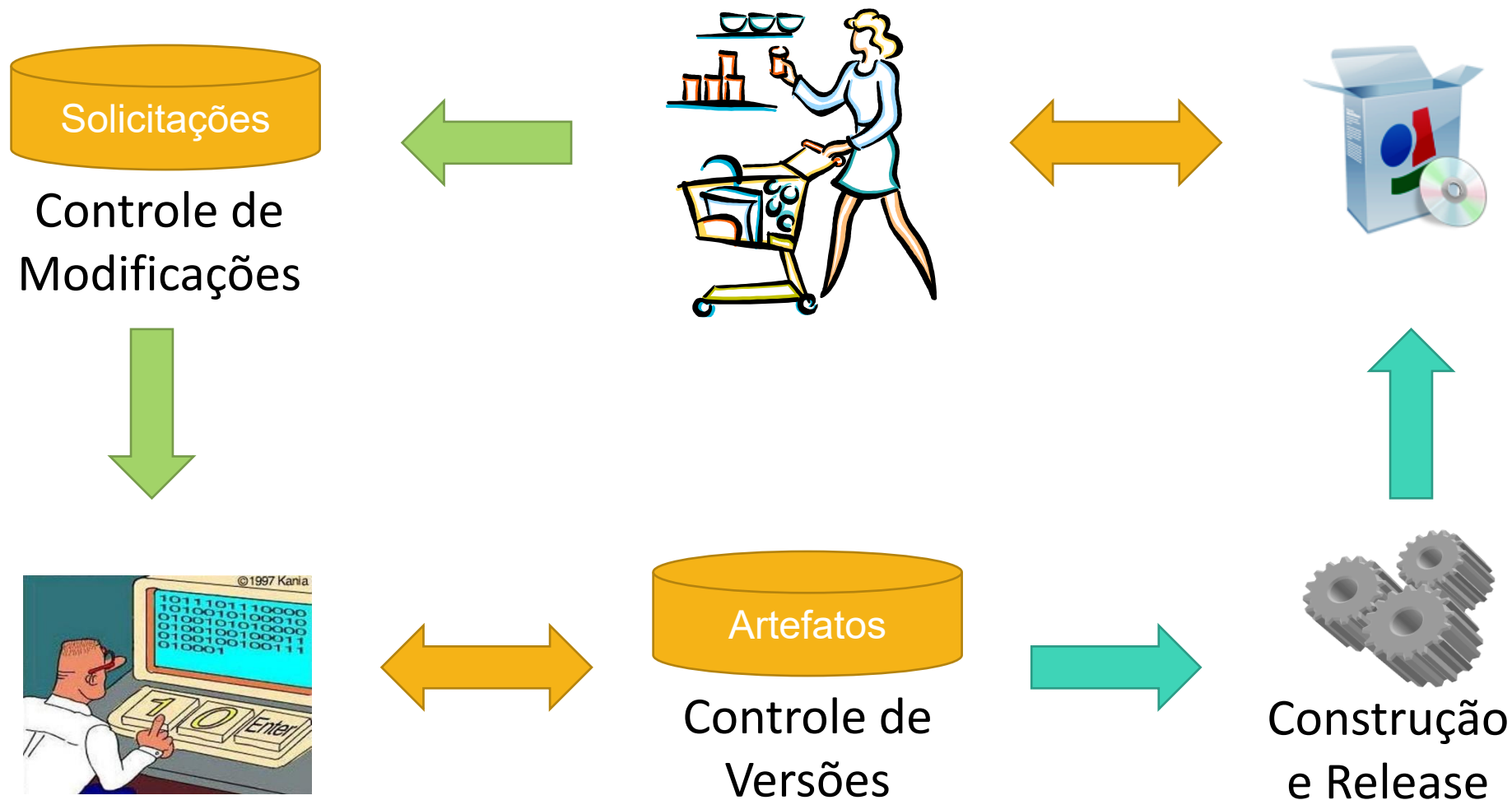
- A Engenharia de Software...
 - Abordagem disciplinada para o desenvolvimento de software
 - Grande diversidade de metodologias
- Ponto em comum nas metodologias:
 - refinamentos sucessivos de **artefatos**



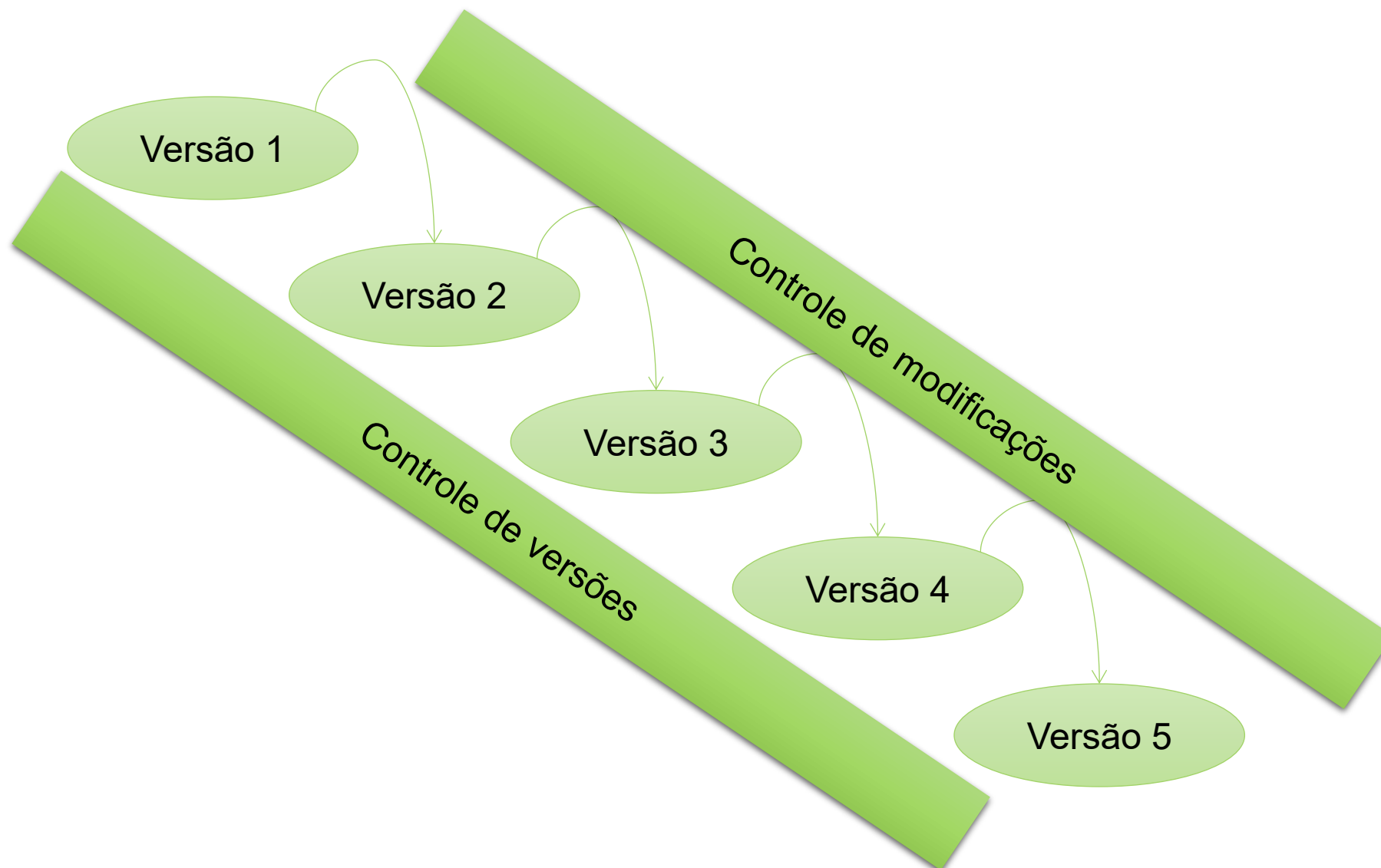
Mas onde ficam esses artefatos?



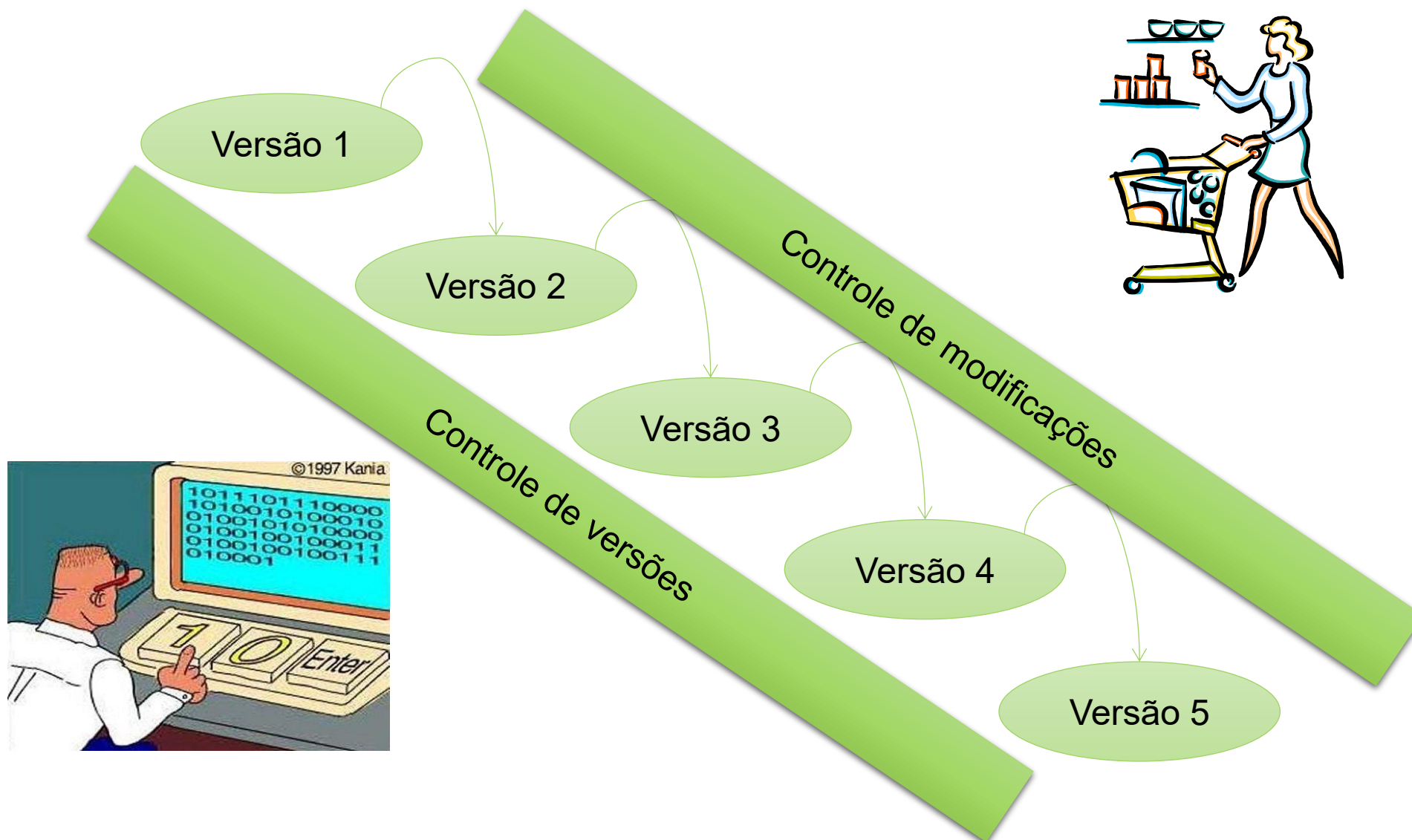
Sistema de Gerência de Configuração



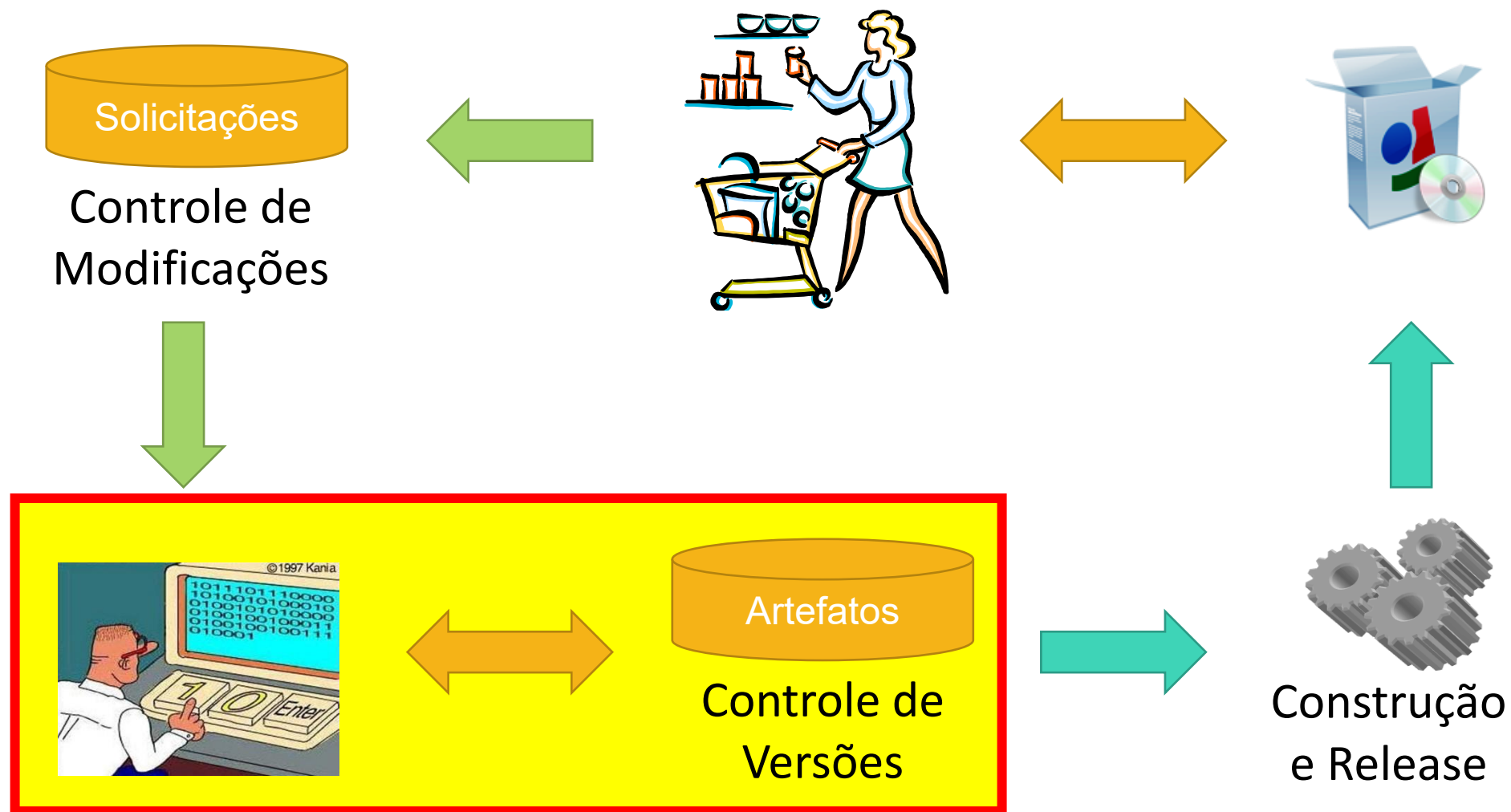
Sistema de Gerência de Configuração



Sistema de Gerência de Configuração



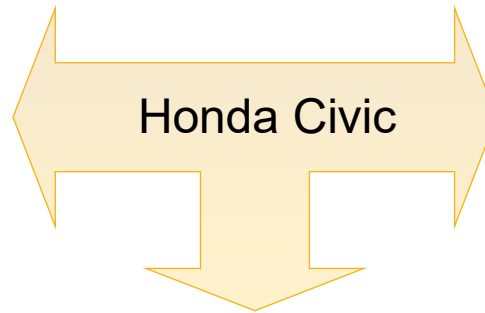
Sistema de Gerência de Configuração



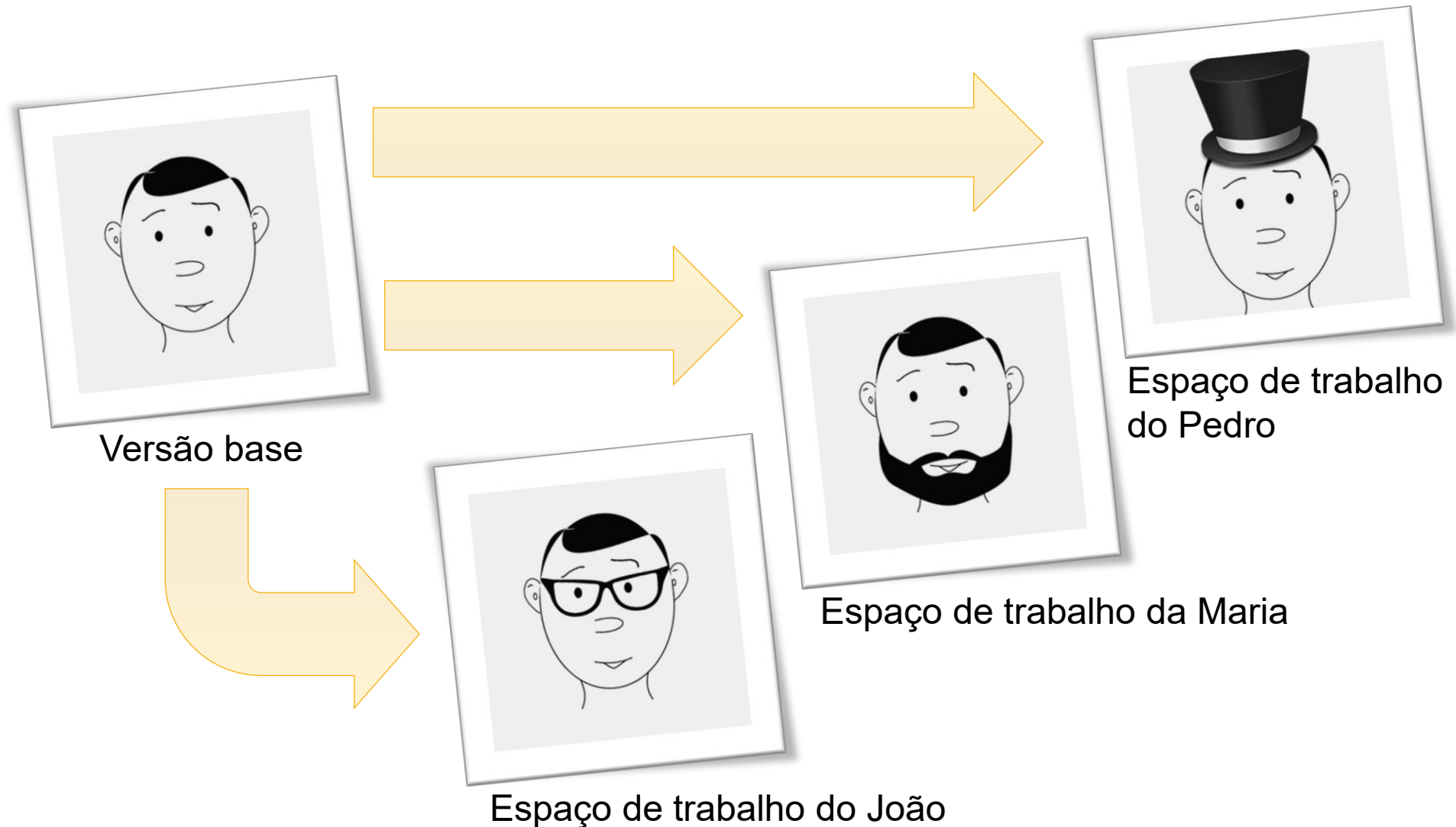
Revisões



Variantes



Cooperação (versões rascunho)



Versões de rascunho podem ser combinadas (operação de *merge*)



João



Maria

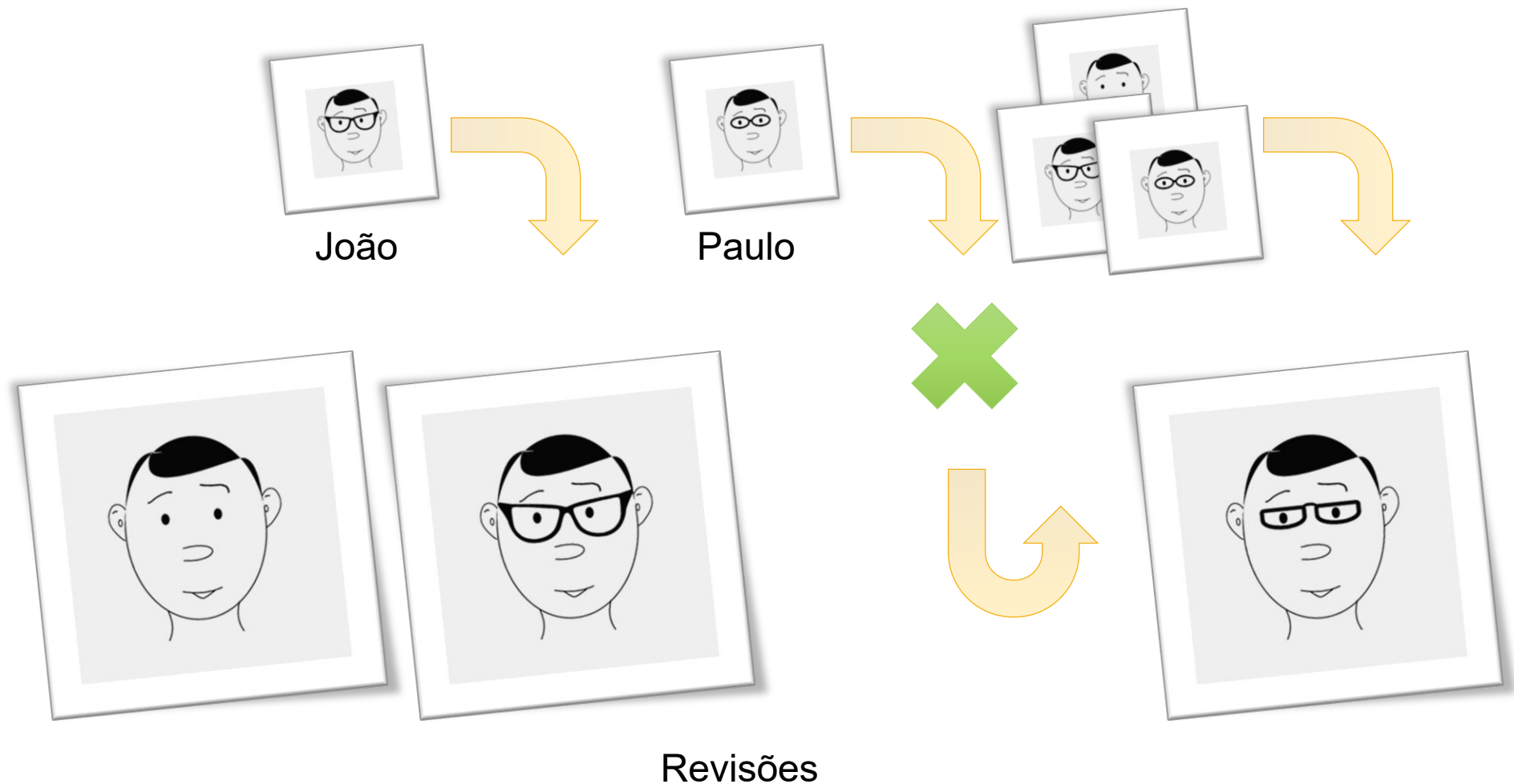


Pedro



Revisões

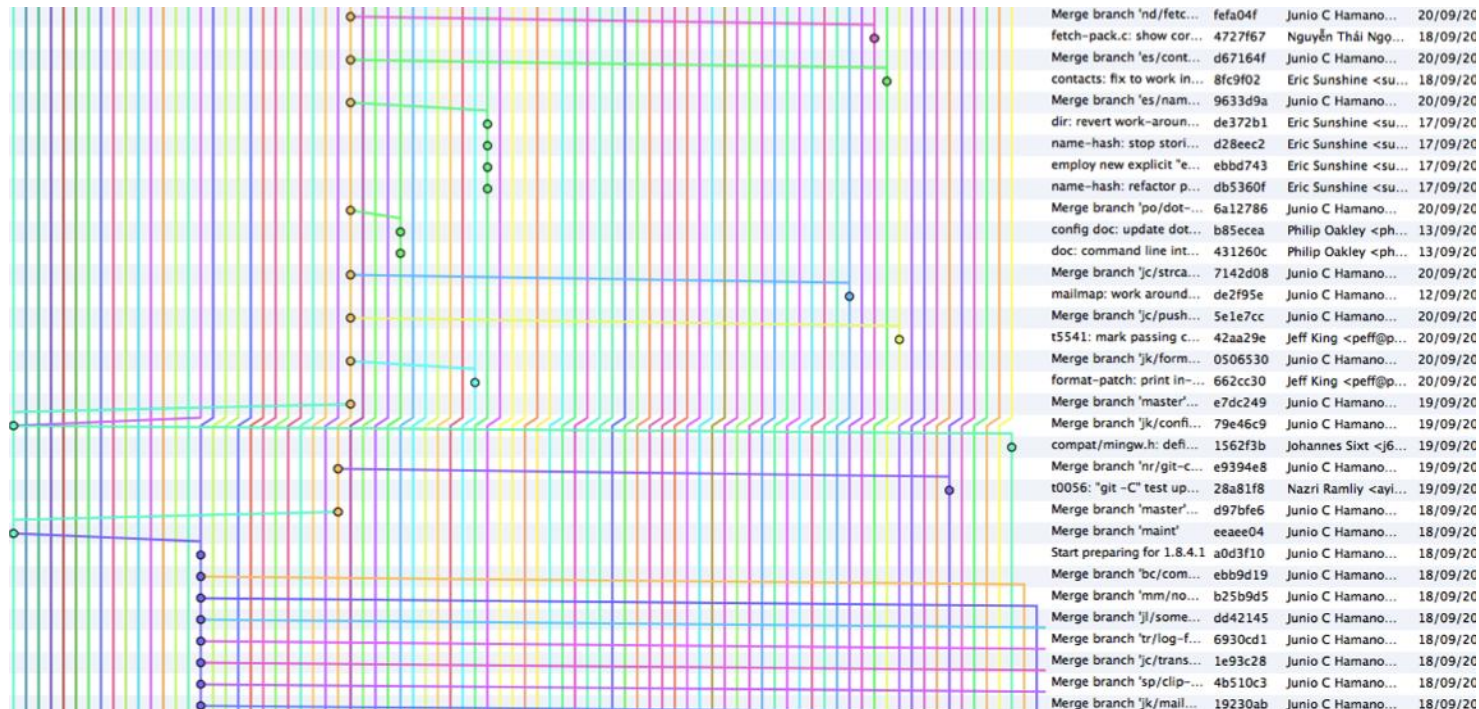
Conflitos podem ocorrer durante o *merge*





Versões no mundo real

- Infinitude de revisões e variantes juntas (sem contar versões rascunho)





Mas afinal, para que servem versões?

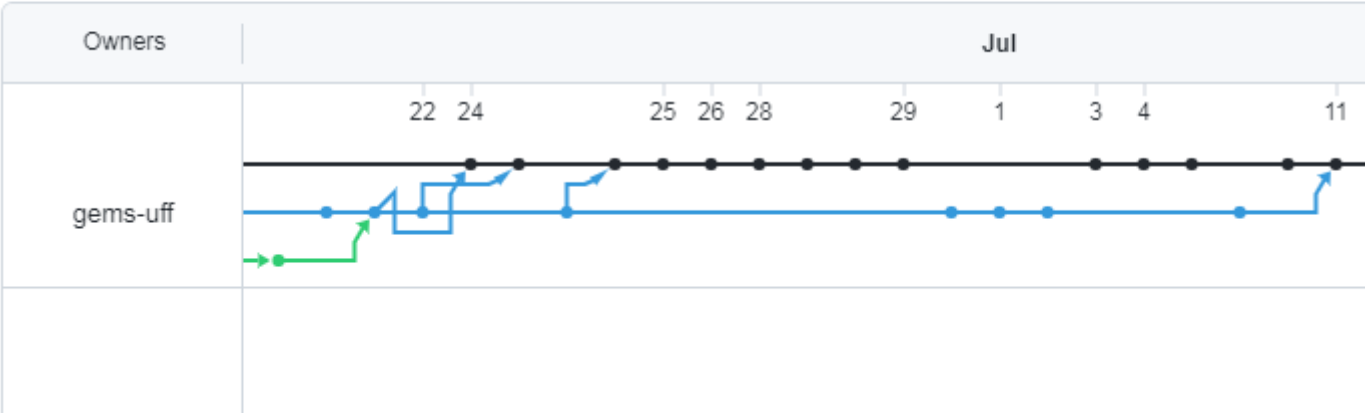
- Sincronizar equipes
- Reproduzir configurações passadas
- Explorar possibilidades
- Isolar desenvolvedores
- Customizar produtos
- Rastrear a introdução de bugs
- Entender a evolução de software (MSR)
- Auditar mudanças
- Etc.



Ramos + Merges

Network graph

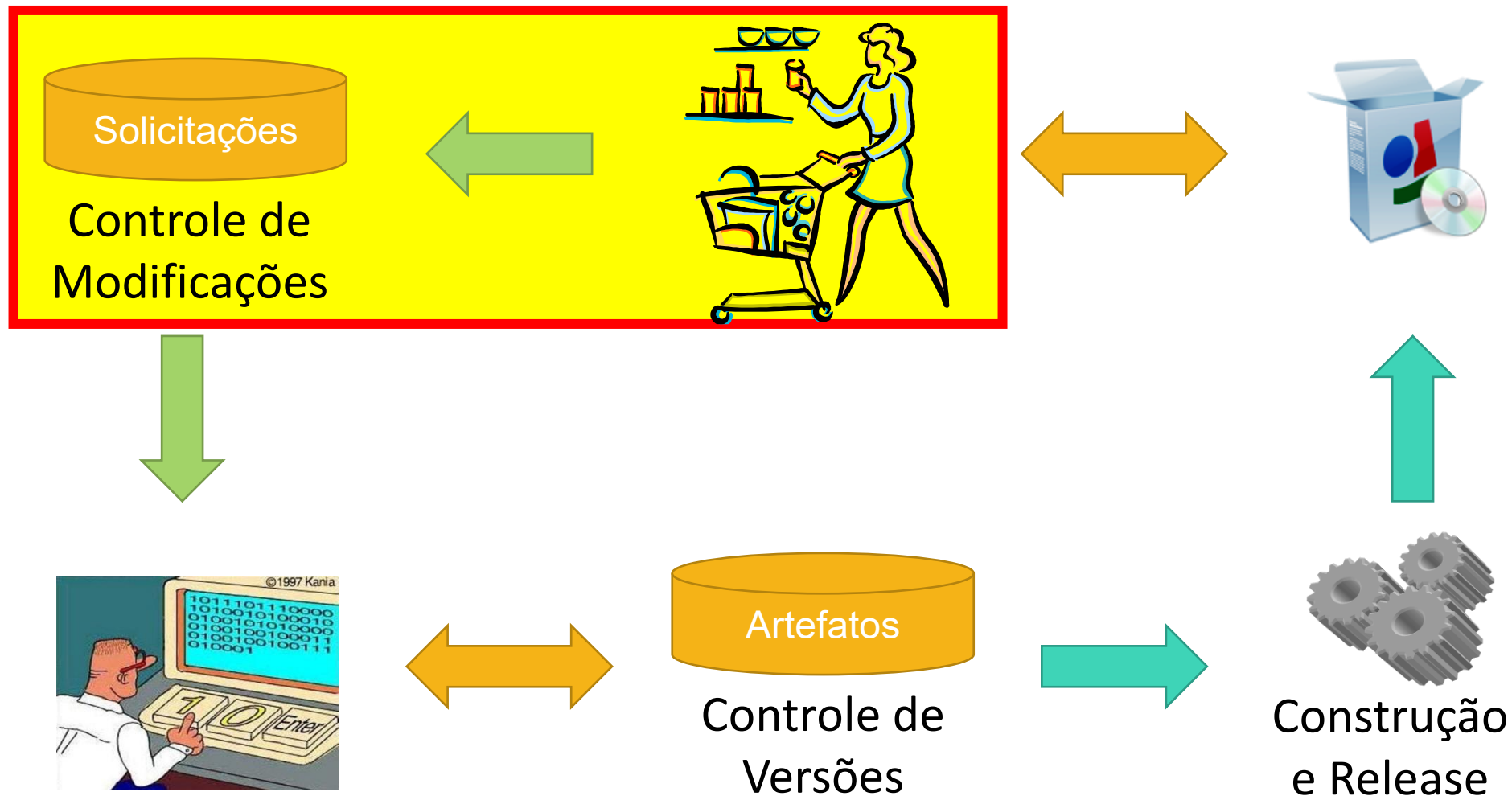
Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



Principal sistema de controle de versão open-source



Sistema de Gerência de Configuração





Baseline

- Configuração revisada e aprovada que serve como base para uma próxima etapa de desenvolvimento e que somente pode ser modificada via processo formal de GCS
- São estabelecidas ao final de cada fase de desenvolvimento



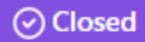
Controle de modificações

- Tarefas
 - Solicitação de modificação
 - Classificação da modificação
 - Análise da modificação
 - Avaliação da modificação
 - Implementação da modificação
 - Verificação da modificação
 - Geração de baseline



Controle de modificações

Inability to restore some items from trash #2369



Closed

dstillman opened this issue 15 days ago · 2 comments



dstillman commented 15 days ago · edited ▾

Member



<https://forums.zotero.org/discussion/94771/failed-to-restore-to-the-library>

```
[JavaScript Error: "Error: getNotes() cannot be called on items of type 'note'" {file:
"chrome://zotero/content/xpcom/data/item.js" line: 2250}]
```

(Not sure if this error is from this. We can get a Debug ID if we're not sure.)

Regression from [48a3235](#) ?



dstillman added the **Bug** label 15 days ago



dstillman assigned **AbeJellinek** 15 days ago



Controle de modificações

✓ Fix error restoring top-level notes & attachments

[Browse files](#)

Fixes #2369

master

AbeJellinek committed 14 days ago

1 parent 003b3db commit b54466f089b8933025f2f0fbf6ea56f24eea602e

Showing 1 changed file with 3 additions and 2 deletions.

Split

Unified

chrome/content/zotero/zoteroPane.js

↑	@@ -1946,14 +1946,16 @@ var ZoteroPane = new function()	
1946	1946	}
1947	1947	
1948	1948	let parent = this.itemsView.getRow(row).ref;
1949	+	let children = [];
1950	+	if (!parent.isNote()) children.push(...parent.getNotes(true));
1951	+	if (!parent.isAttachment()) children.push(...parent.getAttachments(true));
1949	1952	
1950	1953	if (isSelected(parent)) {
1951	1954	if (parent.deleted) {
1952	1955	parent.deleted = false;
1953	1956	await parent.save();
1954	1957	}
1955	1958	
1956	-	let children = [...parent.getNotes(true), ...parent.getAttachments(true)];
1957	1959	let noneSelected = !children.some(isSelected);
1958	1960	for (let child of Zotero.Items.get(children)) {
1959	1961	if ((noneSelected isSelected(child)) && child.deleted) {
↑	@@ -1963,7 +1965,6 @@ var ZoteroPane = new function()	

Exemplo de ferramentas de controle de modificações

- Livre
 - Github
 - Bugzilla
 - Mantis
 - Redmine
 - Trac
- Comercial
 - ClearQuest (IBM Rational)
 - JIRA (Atlassian)
 - StarTeam (Borland)
 - Synergy/Change (Telelogic)
 - TeamTrack (Serena)
 - Team Foundation Server (Microsoft)



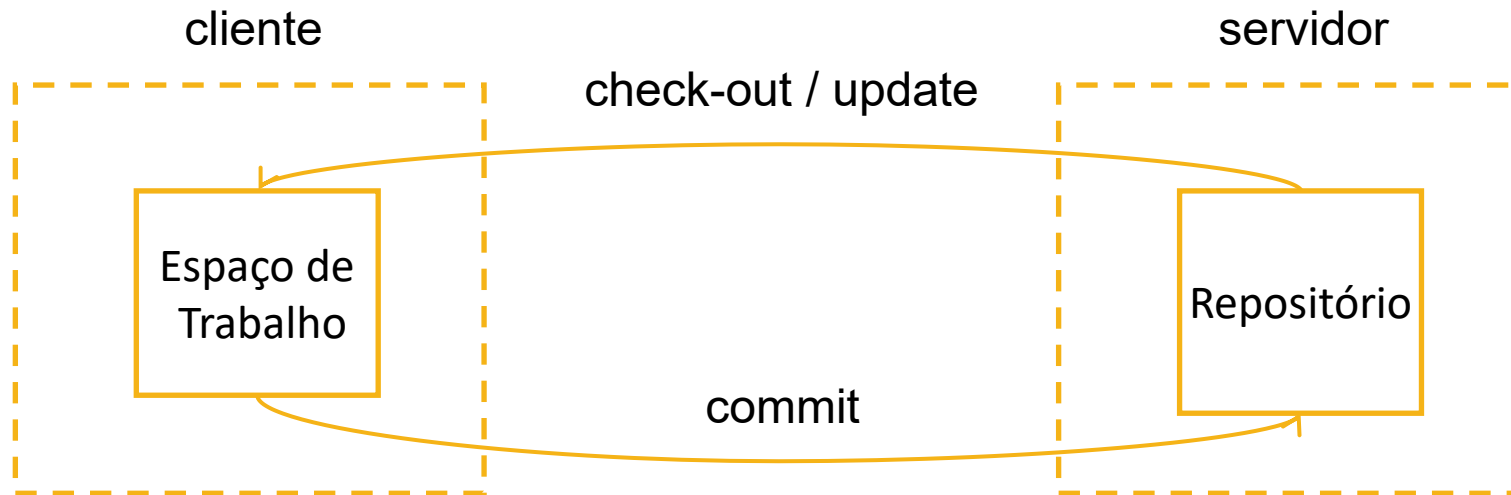
Agenda

01 GC

02 Git

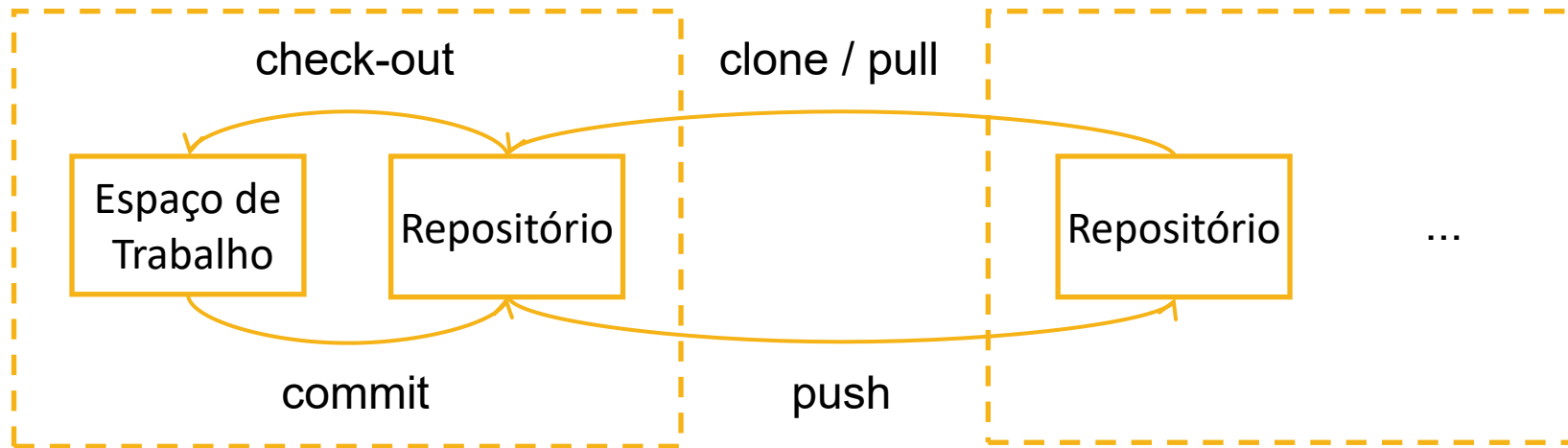
Centralizado

- Anos 80/90 – Sistemas cliente-servidor
 - CVS (1986)
 - Subversion (2000)



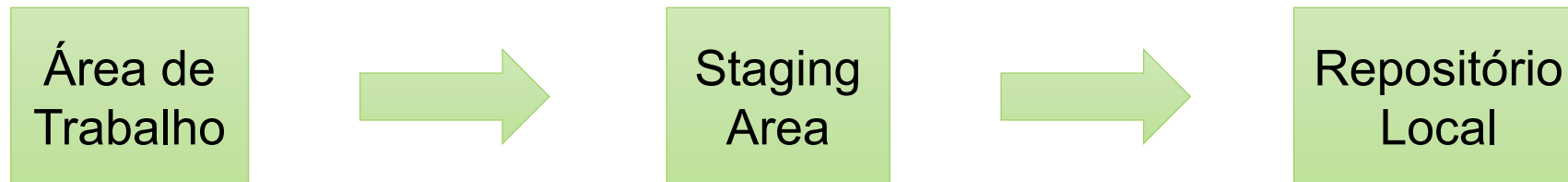
Distribuído

- Anos 2000 – Sistemas peer-to-peer
 - Git (2005)
 - Mercurial (2005)

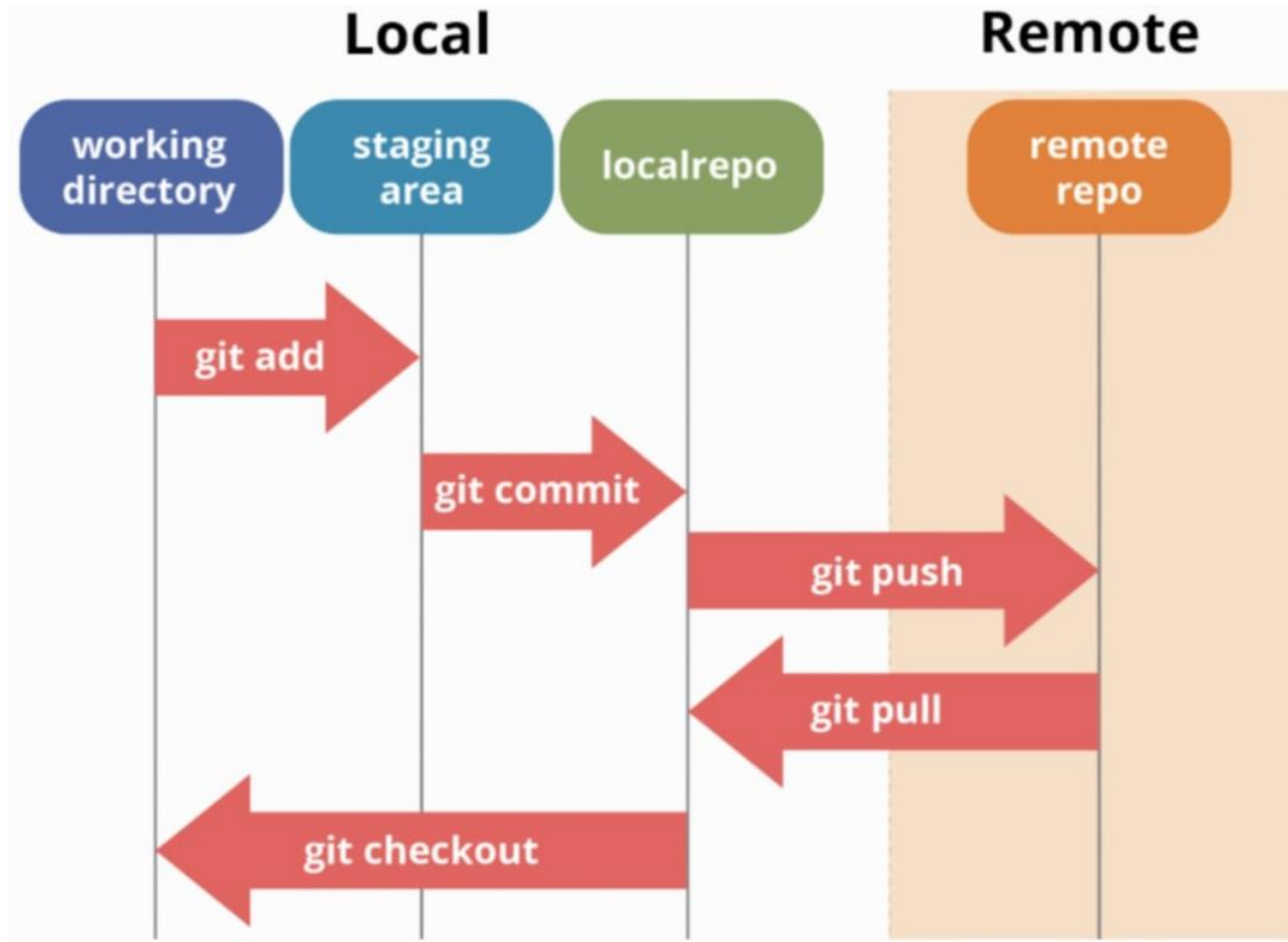


Conceitos básicos: *staging area*

- Área onde são colocados os arquivos que pretendemos enviar para o repositório



*Área de preparação





Conceitos básicos: *commit* id

- Cada sistema de controle de versão usa uma estratégia diferente para identificar *commits*
 - Número sequencial por arquivo (CVS)
 - Número sequencial por repositório (Subversion)
 - Hash (Git e Mercurial)



Conceitos básicos: *apelidos*

- A versão base do seu espaço de trabalho
 - *HEAD*
- O ramo principal do seu repositório
 - *main*
- O repositório do qual seu repositório foi clonado
 - *origin*



Conceitos básicos: *help*!

- git help
 - Oferece ajuda geral sobre o git
- git help <comando>
 - Oferece ajuda sobre um comando específico do git
- Demais comandos dão dicas do que pode ser feito (leia com atenção as saídas dos comandos!)



Conceitos básicos: *quem sou eu?*

- `git config --global user.name <seu nome>`
 - Configura o nome do usuário
- `git config --global user.email <seu email>`
 - Configura o email do usuário



Repositório local

- `git init <nome>`
 - Cria um repositório Git no diretório
- `git add --all`
 - Adiciona um arquivo (ou todos) na *staging area* para ser enviado ao repositório no próximo *commit*
- `git commit -m "mensagem"`
 - Envia os arquivos que estão na *staging area* para o repositório



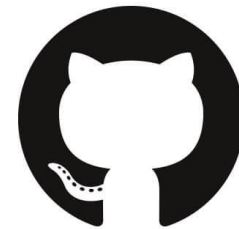
Repositório remoto

- `git clone <url>`
 - Cria um repositório local copiando o histórico de um repositório remoto
- `git pull`
 - Atualiza o repositório local e o espaço de trabalho em relação a um repositório remoto
- `git push`
 - Atualiza o repositório remoto em relação ao repositório local



Interface gráfica

- É possível fazer todos esses passos de forma visual
- Dentre várias ferramentas, temos...



GitHub
Desktop



Inspecionando mudanças

- git status
 - Inspeciona o espaço de trabalho
- git show
 - Inspeciona um *commit*
- git diff
 - Compara o espaço de trabalho com a staging area ou com alguma versão do repositório



Repositório local com ramos

- `git branch --all -v`
 - Lista os ramos existentes no repositório
- `git branch <nome do ramo>`
 - Cria um ramo à partir da versão indicada no HEAD
- `git branch -d <nome do ramo>`
 - Remove um ramo
- `git checkout <commit id ou nome do ramo>`
 - Troca a versão base do espaço de trabalho
- `git merge <nome do ramo>`
 - Combina um ramo com o ramo corrente



Múltiplos repositórios remotos

- `git remote -v`
 - Listar os repositórios remotos cadastrados
- `git remote add <nome> <url>`
 - Adiciona um novo repositório remoto
- `git remote remove <nome>`
 - Remove um repositório remoto existente

Serviço de hospedagem

- Há diversos serviços de hospedagem de repositórios Git na Internet
- Dentre eles, vamos praticar com...



GitHub



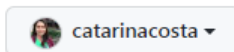
Bora criar nosso repositório

Create a new repository

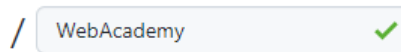
A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner *



Repository name *



Great repository names are short and memorable. Need inspiration? How about [fantastic-goggles?](#)

Description (optional)

Projeto teste para a disciplina Fundamentos



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.



Add a README file

This is where you can write a long description for your project. [Learn more.](#)



Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

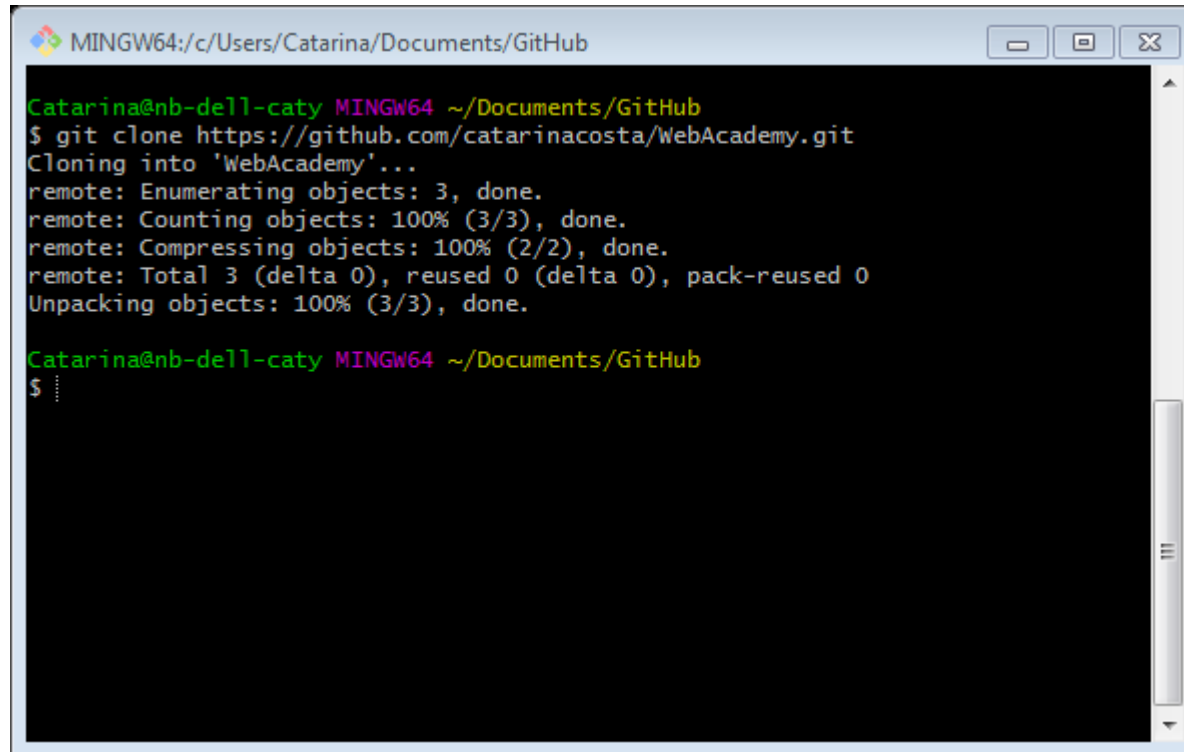


Choose a license

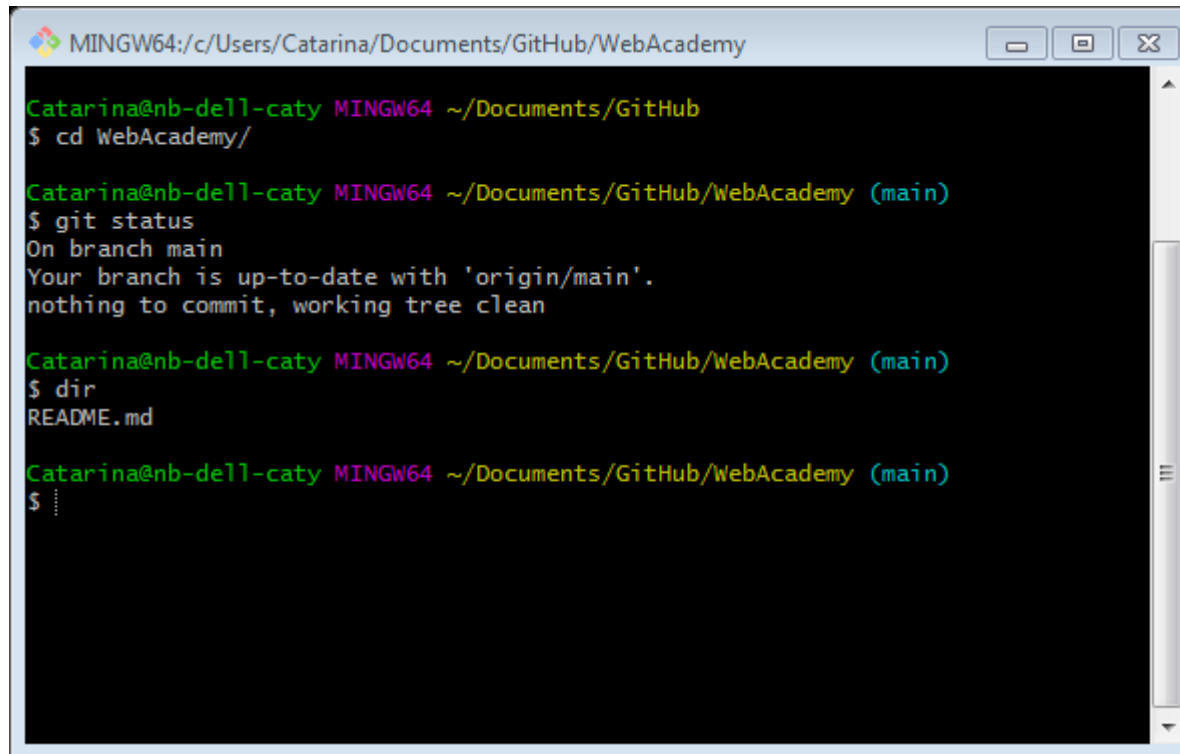
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Bora criar nosso repositório

A screenshot of a Windows terminal window titled 'MINGW64:/c/Users/Catarina/Documents/GitHub'. The window shows the execution of a git clone command. The prompt is 'Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub'. The command entered is '\$ git clone https://github.com/catarinacosta/WebAcademy.git'. The output shows the cloning process: 'Cloning into 'WebAcademy'...', 'remote: Enumerating objects: 3, done.', 'remote: Counting objects: 100% (3/3), done.', 'remote: Compressing objects: 100% (2/2), done.', 'remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0', and 'Unpacking objects: 100% (3/3), done.'. The prompt returns to '\$ '.

Verificando o que tem no repo

A screenshot of a Windows terminal window titled "MINGW64:/c/Users/Catarina/Documents/GitHub/WebAcademy". The terminal shows a series of commands and their outputs. The user is on the 'main' branch, and the working tree is clean. The directory listing shows a file named 'README.md'.

```
MINGW64:/c/Users/Catarina/Documents/GitHub/WebAcademy

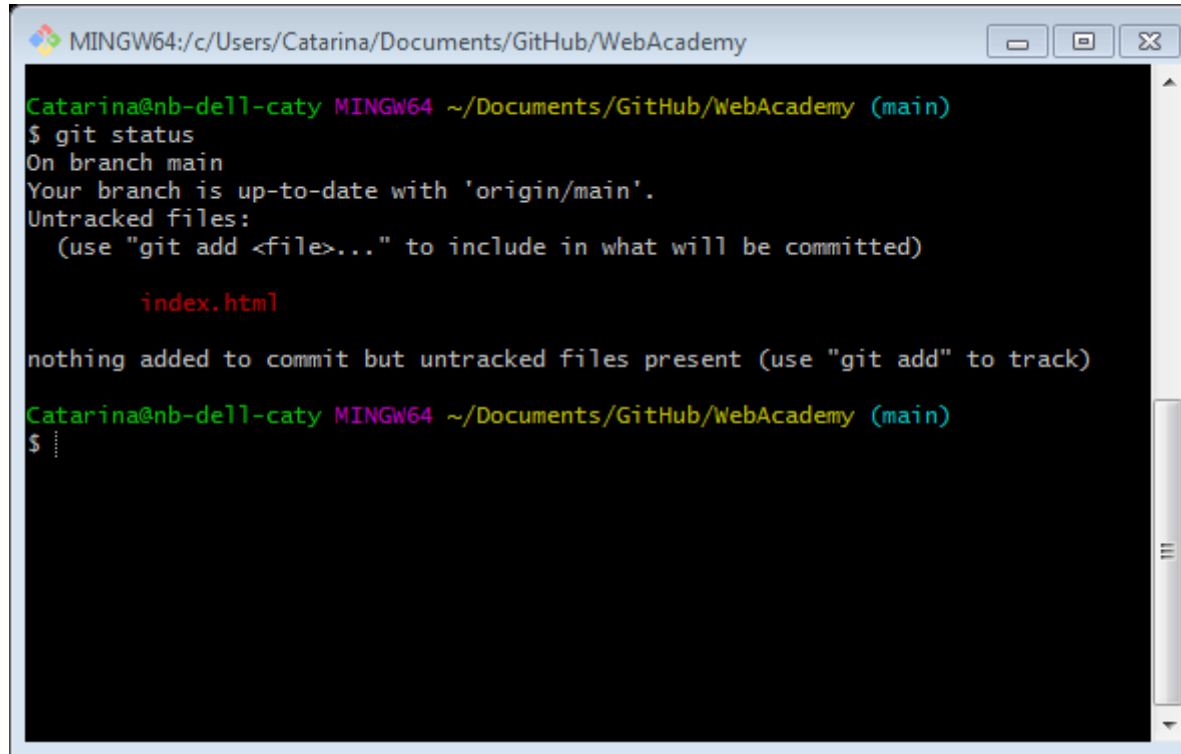
Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub
$ cd WebAcademy/

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$ git status
On branch main
Your branch is up-to-date with 'origin/main'.
nothing to commit, working tree clean

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$ dir
README.md

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$
```

Verificando a alteração realizada

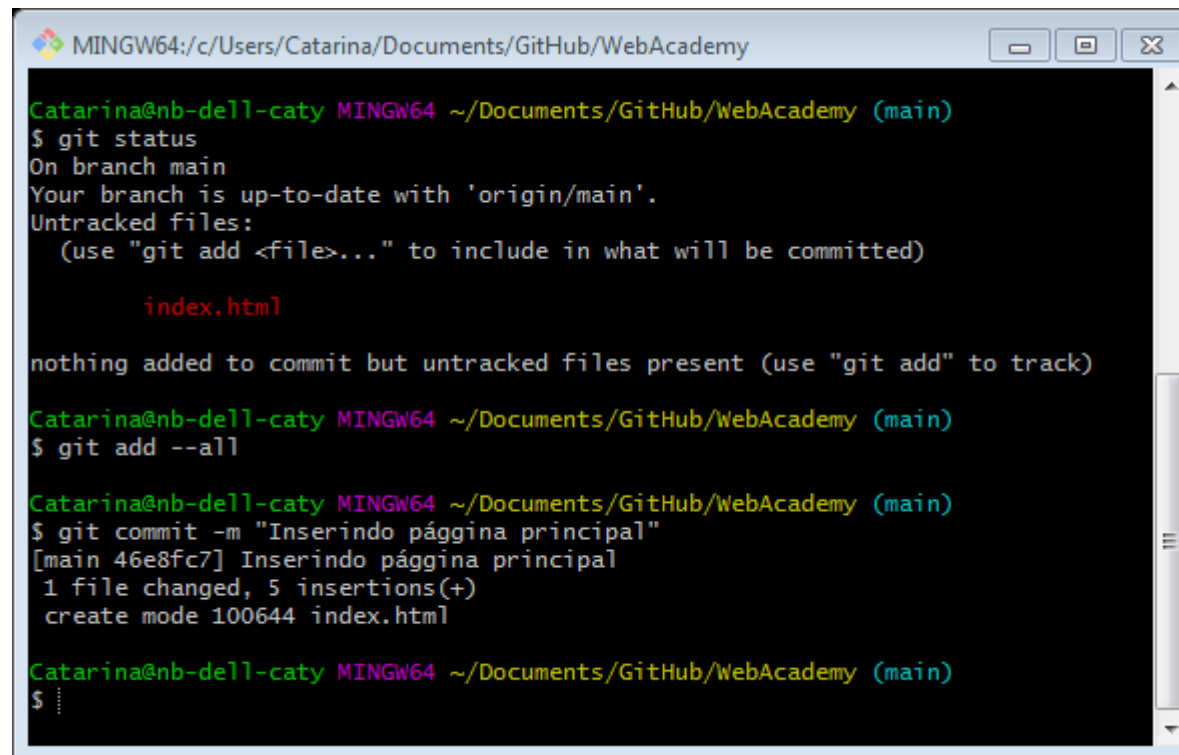
A screenshot of a Windows terminal window titled 'MINGW64:/c/Users/Catarina/Documents/GitHub/WebAcademy'. The window shows the output of the 'git status' command. The text is as follows:

```
Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$ git status
On branch main
Your branch is up-to-date with 'origin/main'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        index.html

nothing added to commit but untracked files present (use "git add" to track)
Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$
```

Adicionando e commitando



```
MINGW64:/c/Users/Catarina/Documents/GitHub/WebAcademy

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$ git status
On branch main
Your branch is up-to-date with 'origin/main'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        index.html

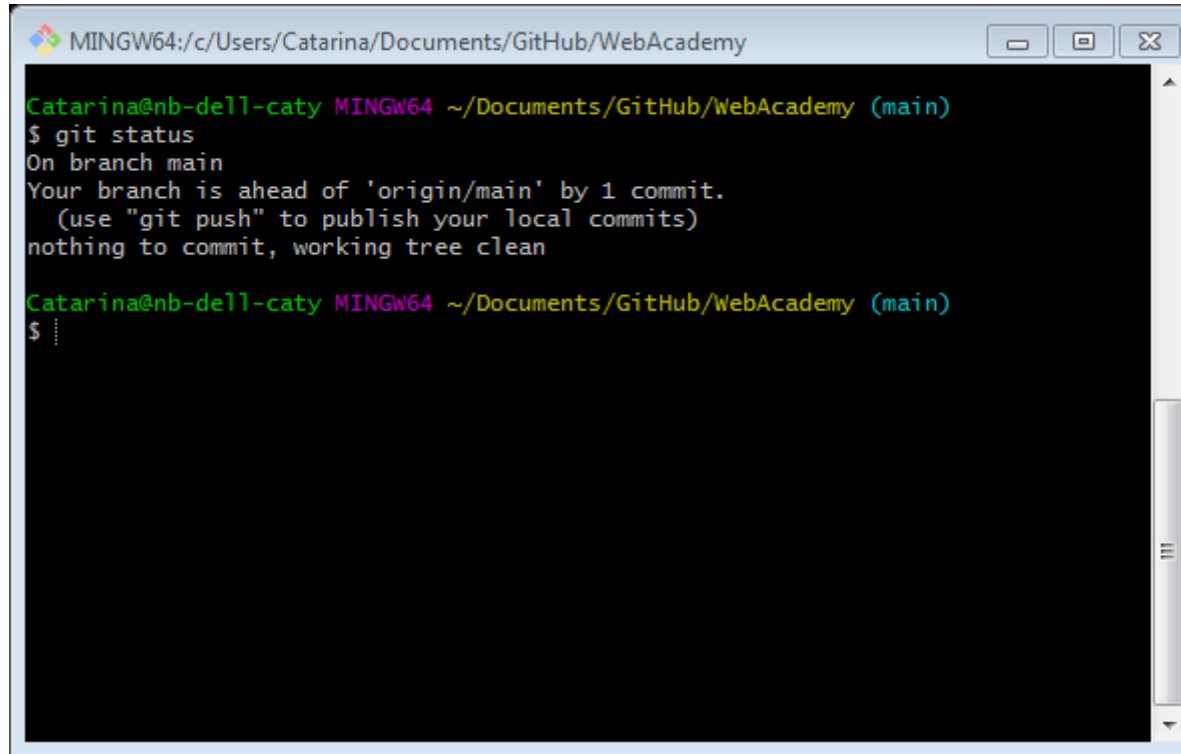
nothing added to commit but untracked files present (use "git add" to track)

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$ git add --all

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$ git commit -m "Inserindo página principal"
[main 46e8fc7] Inserindo página principal
1 file changed, 5 insertions(+)
create mode 100644 index.html

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$
```

Verificando o status

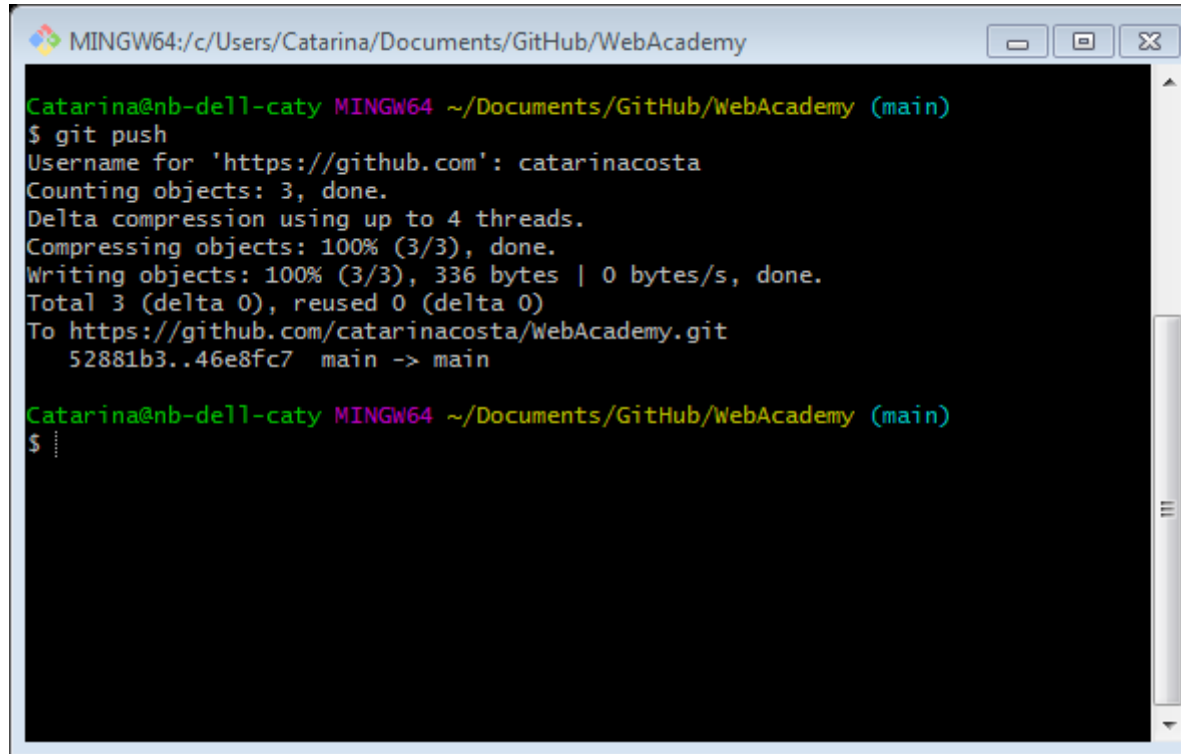


```
MINGW64:/c/Users/Catarina/Documents/GitHub/WebAcademy

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working tree clean

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$
```

Bora criar nosso repositório

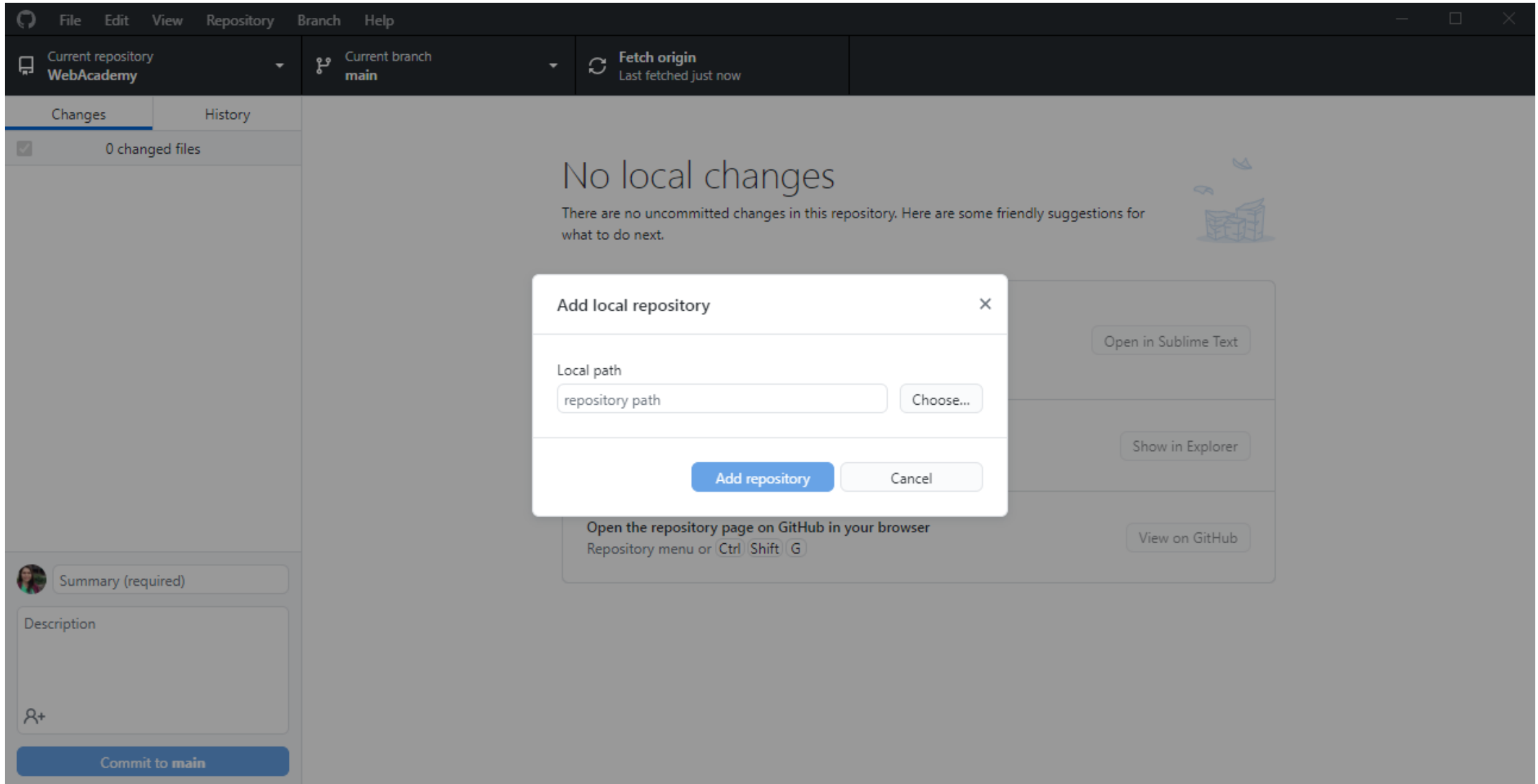
A screenshot of a Windows terminal window titled 'MINGW64:/c/Users/Catarina/Documents/GitHub/WebAcademy'. The window shows the execution of the 'git push' command. The output indicates that the push was successful, with objects being counted, compressed, and written to the repository. The commit hash '52881b3..46e8fc7' is shown, along with the branch 'main' being pushed to 'main'.

```
MINGW64:/c/Users/Catarina/Documents/GitHub/WebAcademy

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$ git push
Username for 'https://github.com': catarinacosta
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 336 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/catarinacosta/WebAcademy.git
    52881b3..46e8fc7  main -> main

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy (main)
$
```


Abrir no Github Desktop



Alterar / Commit / Push

Current repository: WebAcademy

Current branch: main

Fetch origin
Last fetched 2 minutes ago

Changes 1 | History

1 changed file

index.html

```
@@ -2,6 +2,6 @@
2 2 <head>
3 3 </head>
4 4 <body>
5 - <h1>Git</h1>
5 + <h1>Git no Web Academy</h1>
6 6 </body>
7 7 </html>
```

Update index.html

Description

Commit to main

Alterar / Commit / Push

The screenshot displays the Visual Studio Code interface with the following components:

- Menu Bar:** File, Edit, View, Repository, Branch, Help.
- Toolbar:**
 - Current repository: WebAcademy
 - Current branch: main
 - Push origin: Last fetched 2 minutes ago (1 ↑)
- Left Panel:**
 - Changes: 0 changed files
 - History: (empty)
 - Commit Summary:
 - Summary (required): (empty text field)
 - Description: (empty text area)
 - Commit to main: (button)
 - Commit Status: Committed just now, Update index.html, Undo (button)
- Main Area:**
 - Message: No local changes
 - Text: There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.
 - Suggestions:
 - Push commits to the origin remote:** You have 1 local commit waiting to be pushed to GitHub. Always available in the toolbar when there are local commits waiting to be pushed or **Ctrl | P**. (Push origin button)
 - Open the repository in your external editor:** Select your editor in [Options](#). Repository menu or **Ctrl | Shift | A**. (Open in Sublime Text button)
 - View the files of your repository in Explorer:** Repository menu or **Ctrl | Shift | F**. (Show in Explorer button)
 - Open the repository page on GitHub in your browser:** Repository menu or **Ctrl | Shift | G**. (View on GitHub button)



Criar o ramo e mudar para a nova ramificação

```
Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy2 (main)
$ git branch feature
```

```
Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy2 (main)
$ git checkout feature
Switched to branch 'feature'
```

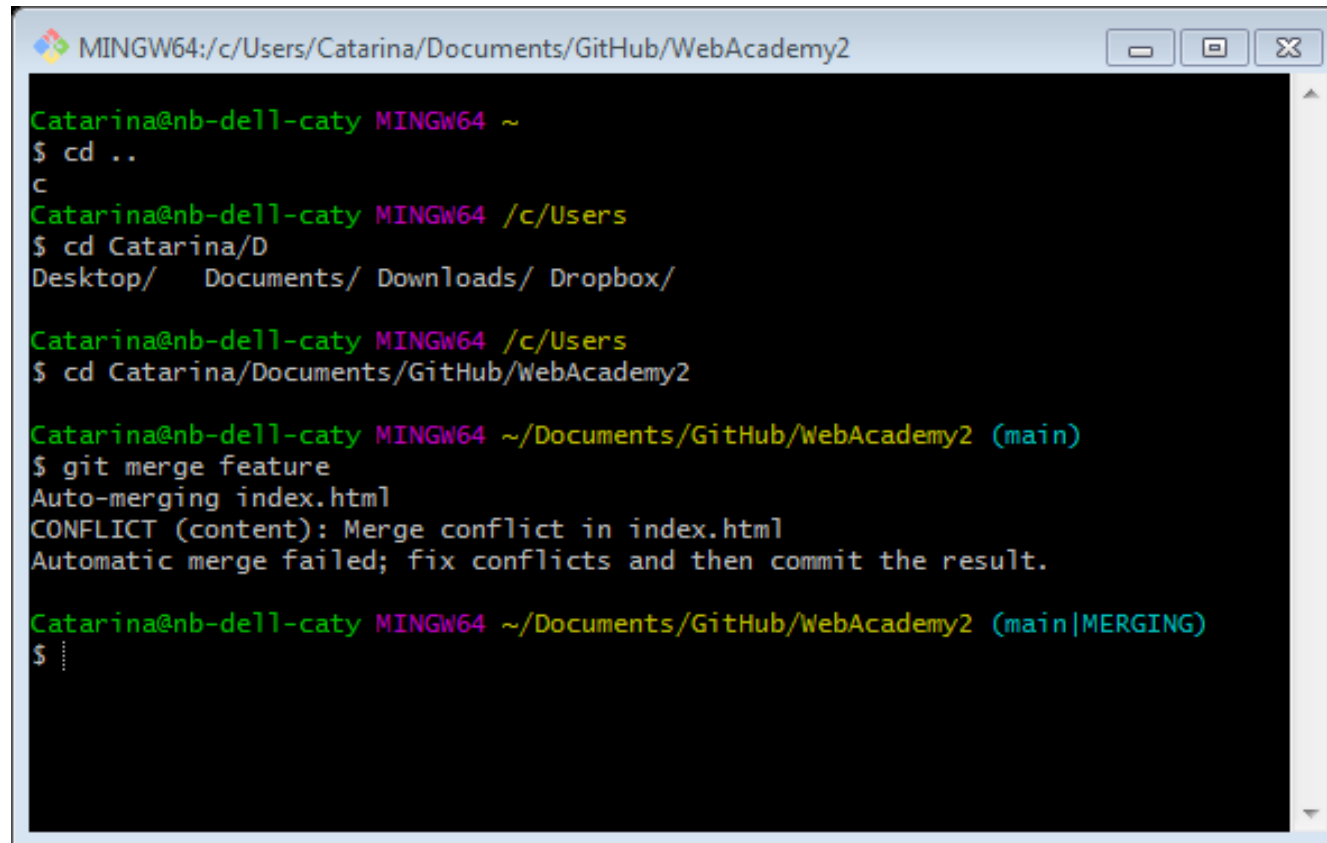
```
Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy2 (feature)
$ git status
On branch feature
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   index.html
```

Para mudar de ramo – git checkout – voltando para o main

```
Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy2 (feature)
$ git checkout main
Switched to branch 'main'
Your branch is up-to-date with 'origin/main'.
```

Tentando fazer o merge -> encontrando conflito



```
MINGW64:/c/Users/Catarina/Documents/GitHub/WebAcademy2

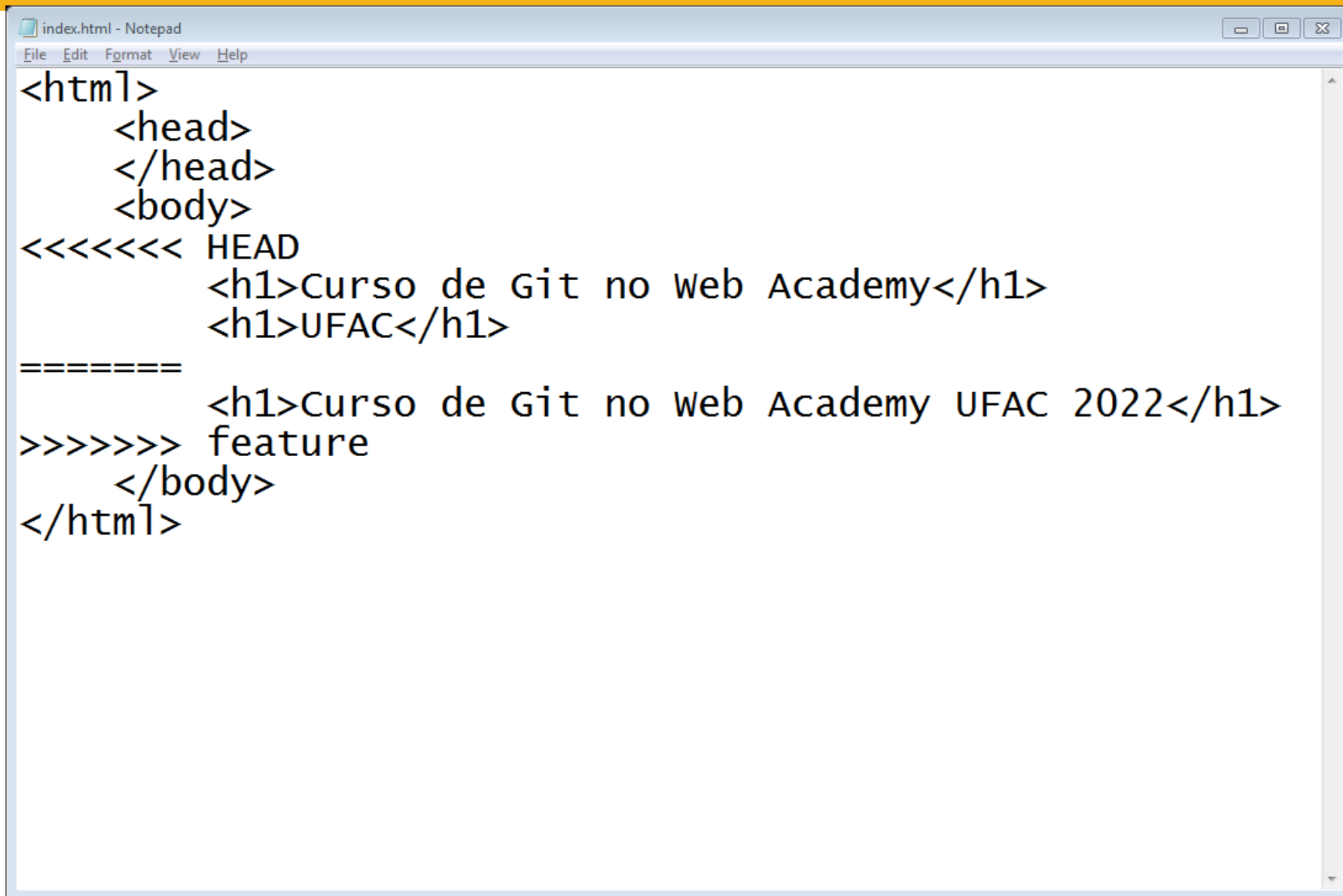
Catarina@nb-dell-caty MINGW64 ~
$ cd ..
c
Catarina@nb-dell-caty MINGW64 /c/Users
$ cd Catarina/D
Desktop/  Documents/  Downloads/  Dropbox/

Catarina@nb-dell-caty MINGW64 /c/Users
$ cd Catarina/Documents/GitHub/WebAcademy2

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy2 (main)
$ git merge feature
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy2 (main|MERGING)
$ ..
```

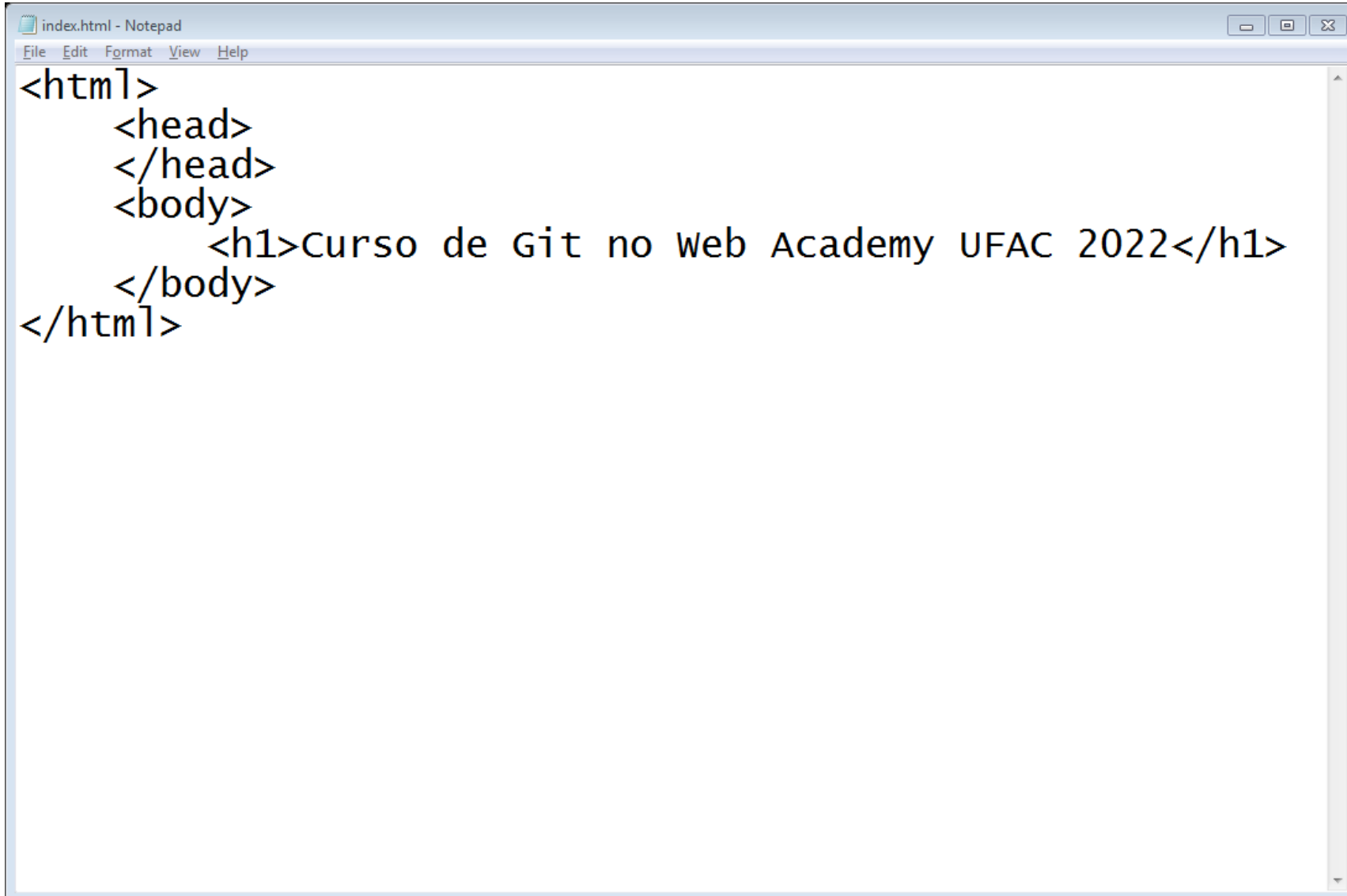
Resolvendo o conflito no editor



The image shows a Notepad window titled "index.html - Notepad". The window contains HTML code with conflict markers. The code is as follows:

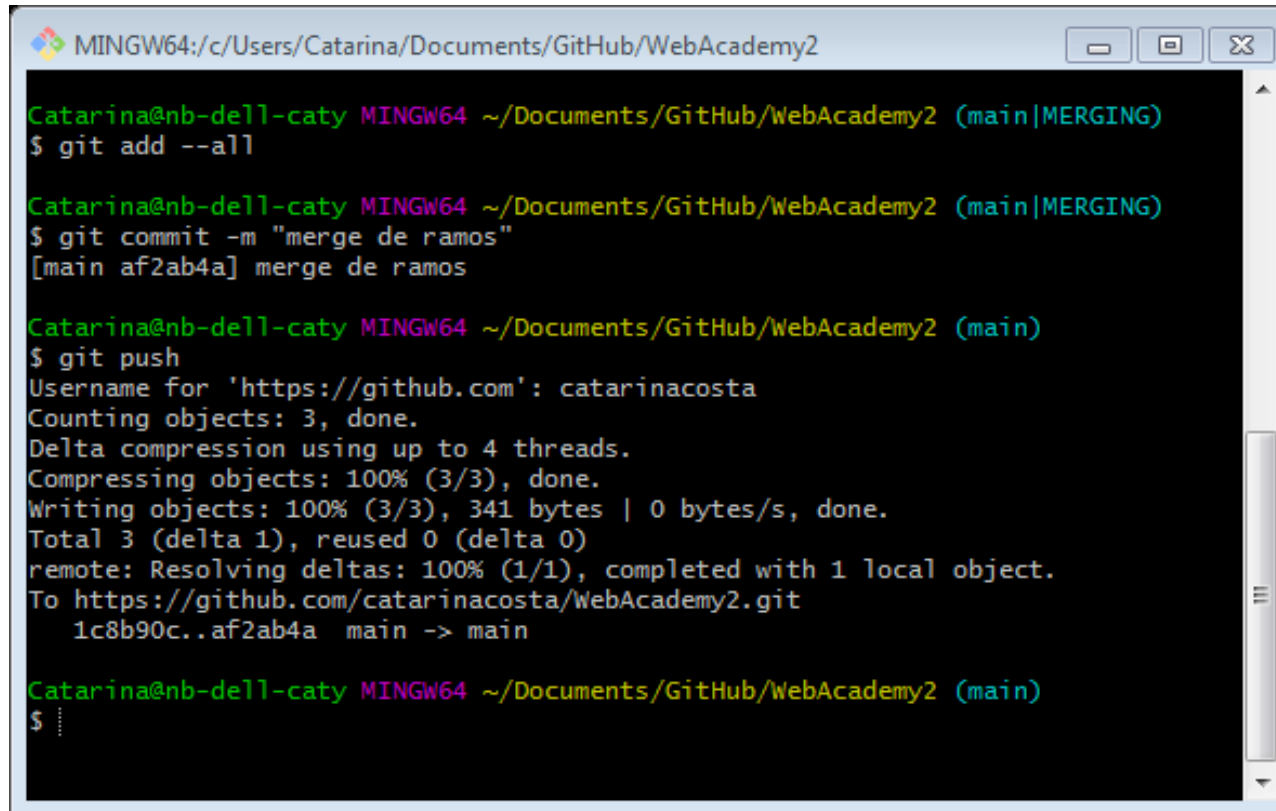
```
<html>
  <head>
  </head>
  <body>
<<<<<<< HEAD
    <h1>Curso de Git no Web Academy</h1>
    <h1>UFAC</h1>
=====
    <h1>Curso de Git no Web Academy UFAC 2022</h1>
>>>>>>> feature
  </body>
</html>
```

Resolvendo o conflito – o que foi escolhido pra ficar e ser commitado

A screenshot of a Notepad window titled 'index.html - Notepad'. The window has a menu bar with 'File', 'Edit', 'Format', 'View', and 'Help'. The text area contains the following HTML code:

```
<html>
  <head>
  </head>
  <body>
    <h1>Curso de Git no Web Academy UFAC 2022</h1>
  </body>
</html>
```


Enviando a solução (fazendo o merge) para o repositório remoto



```
MINGW64:/c/Users/Catarina/Documents/GitHub/WebAcademy2

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy2 (main|MERGING)
$ git add --all

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy2 (main|MERGING)
$ git commit -m "merge de ramos"
[main af2ab4a] merge de ramos

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy2 (main)
$ git push
Username for 'https://github.com': catarinacosta
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 341 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/catarinacosta/WebAcademy2.git
    1c8b90c..af2ab4a  main -> main

Catarina@nb-dell-caty MINGW64 ~/Documents/GitHub/WebAcademy2 (main)
$
```



DICAS DE LEITURA +:

Leituras Git

<https://engsoftmoderna.info/capAp.html>

<https://git-scm.com/book/en/v2>

ATIVIDADE +:

Instalar o Git

Instalar Github desktop

Git bash (ou outro)