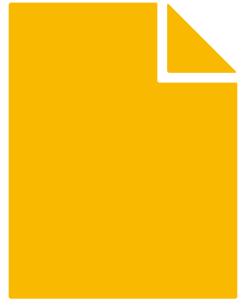


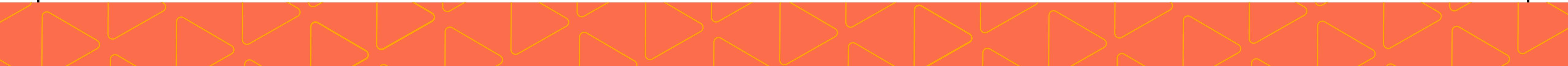
Igor  
iOS Mentor

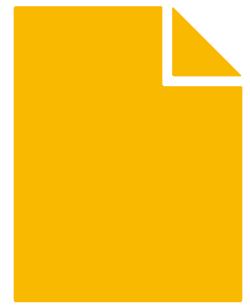


# Controle de versões



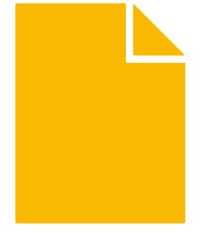
Permite que você registre o **histórico de alterações** em um projeto.





Permite que você registre o **histórico de alterações** em um projeto.

É essencial para **colaborar em equipe**, recuperar **versões antigas** e evitar a **perda de código**.

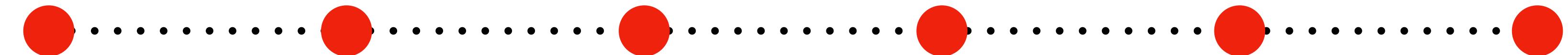


Permite que você registre o histórico de alterações em um projeto.

É essencial para colaborar em equipe, recuperar versões antigas e evitar a perda de código.

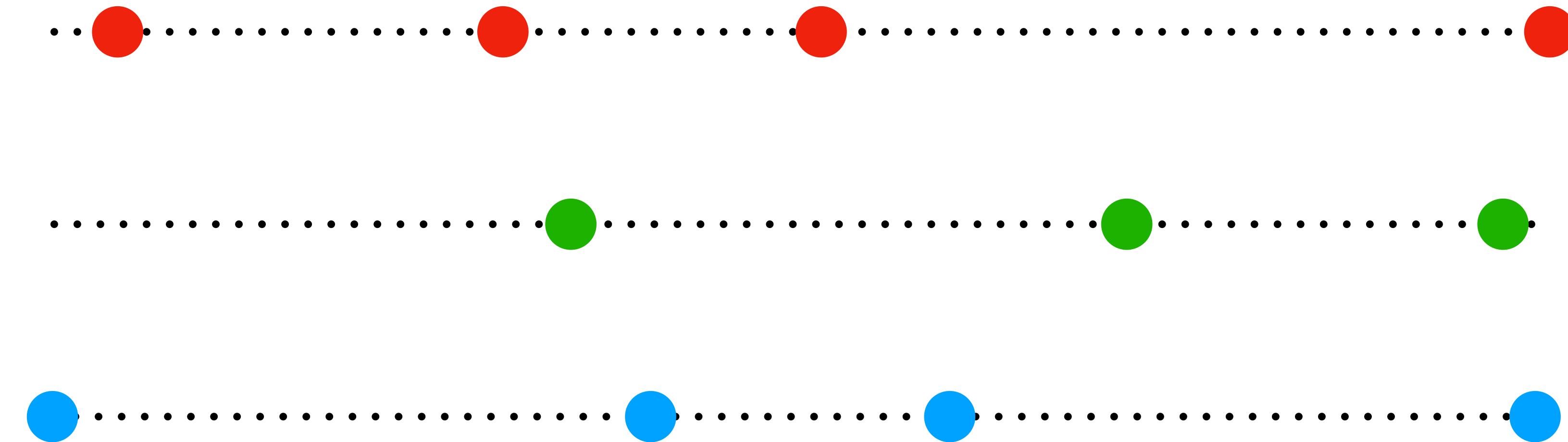
**Git** é a ferramenta mais utilizada para controle de versões.

## Linha do tempo





## Linha do tempo



## Linha do tempo

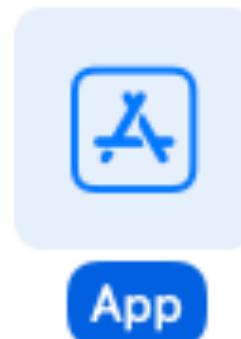


Choose a template for your new project:

Multipurpose iOS macOS watchOS tvOS visionOS DriverKit Other

Filter

Application



App



Document App



Game



Augmented Reality App



App Playground



Sticker Pack App



iMessage App



Safari Extension App

Framework & Library



Framework



Static Library



Metal Library

Cancel

Previous

Next

Choose options for your new project:

Product Name:

Team:  

Organization Identifier:

Bundle Identifier:

Interface:  

Language:  

Testing System:  

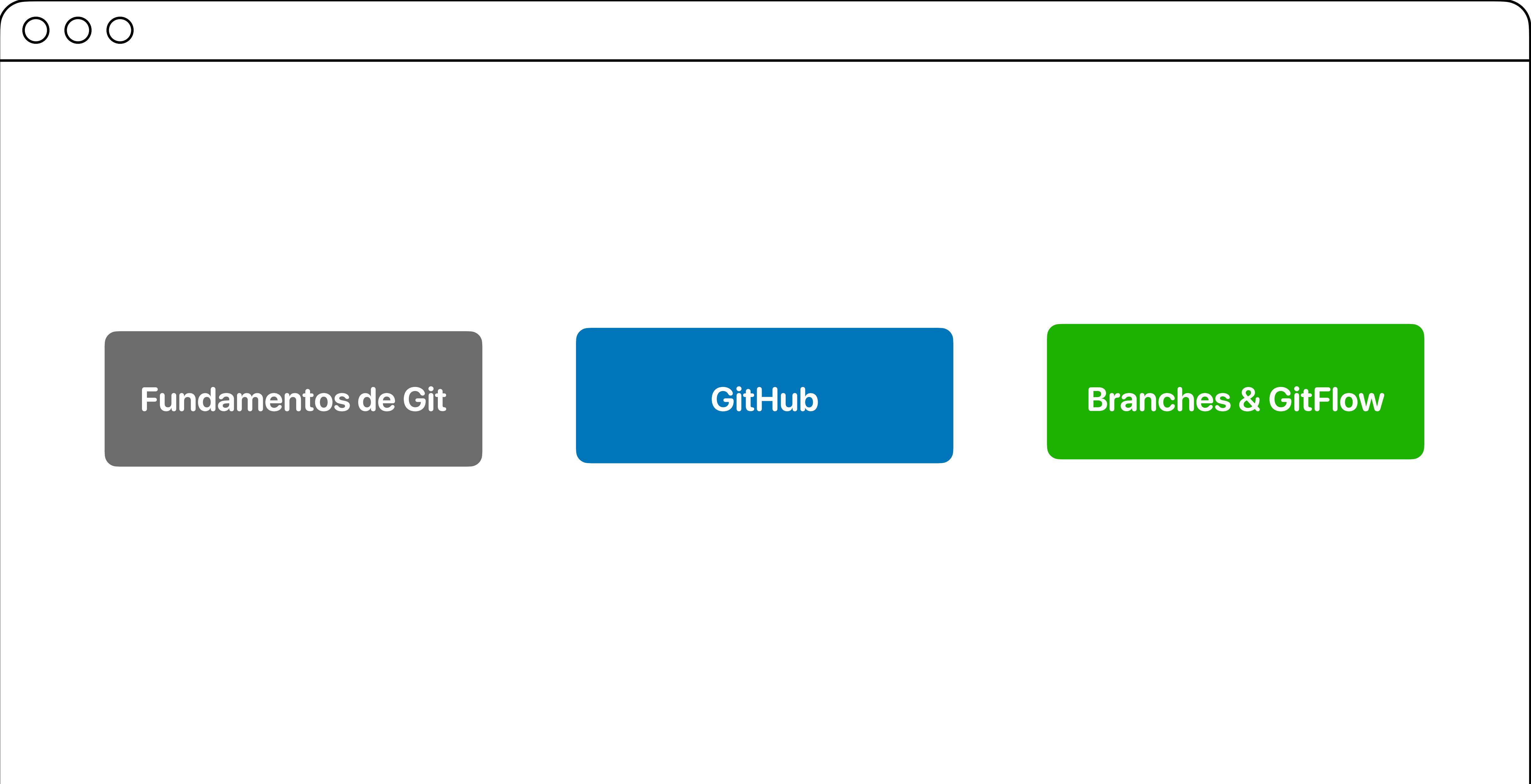
Storage:  

Host in CloudKit

Cancel

Previous

Next

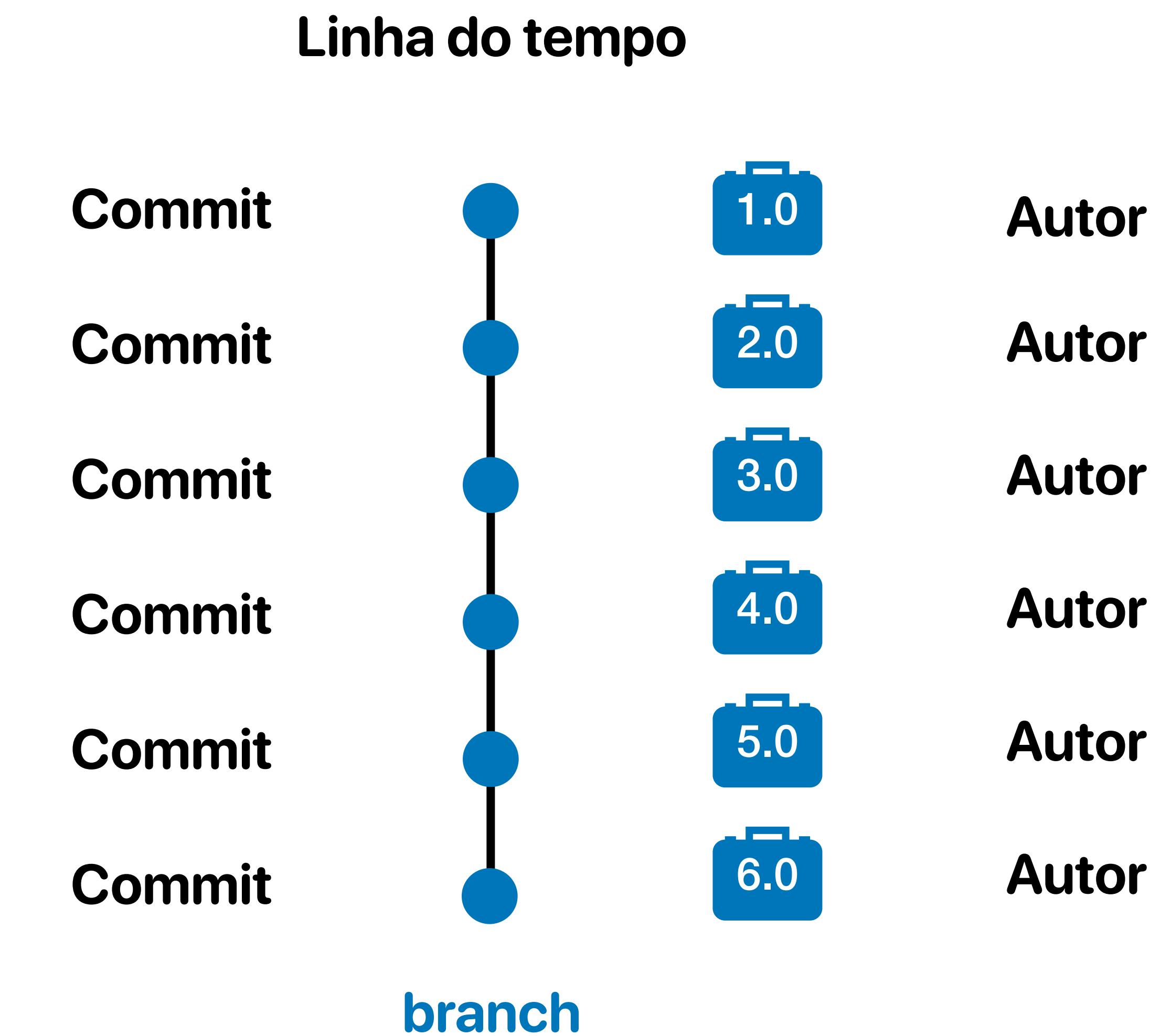


Fundamentos de Git

GitHub

Branches & GitFlow

# Fundamentos de Git





# Configuração

```
git --version
```

```
git config --global user.name "Seu nome"
```

```
git config --global user.email "Seu email"
```

```
git config --list
```

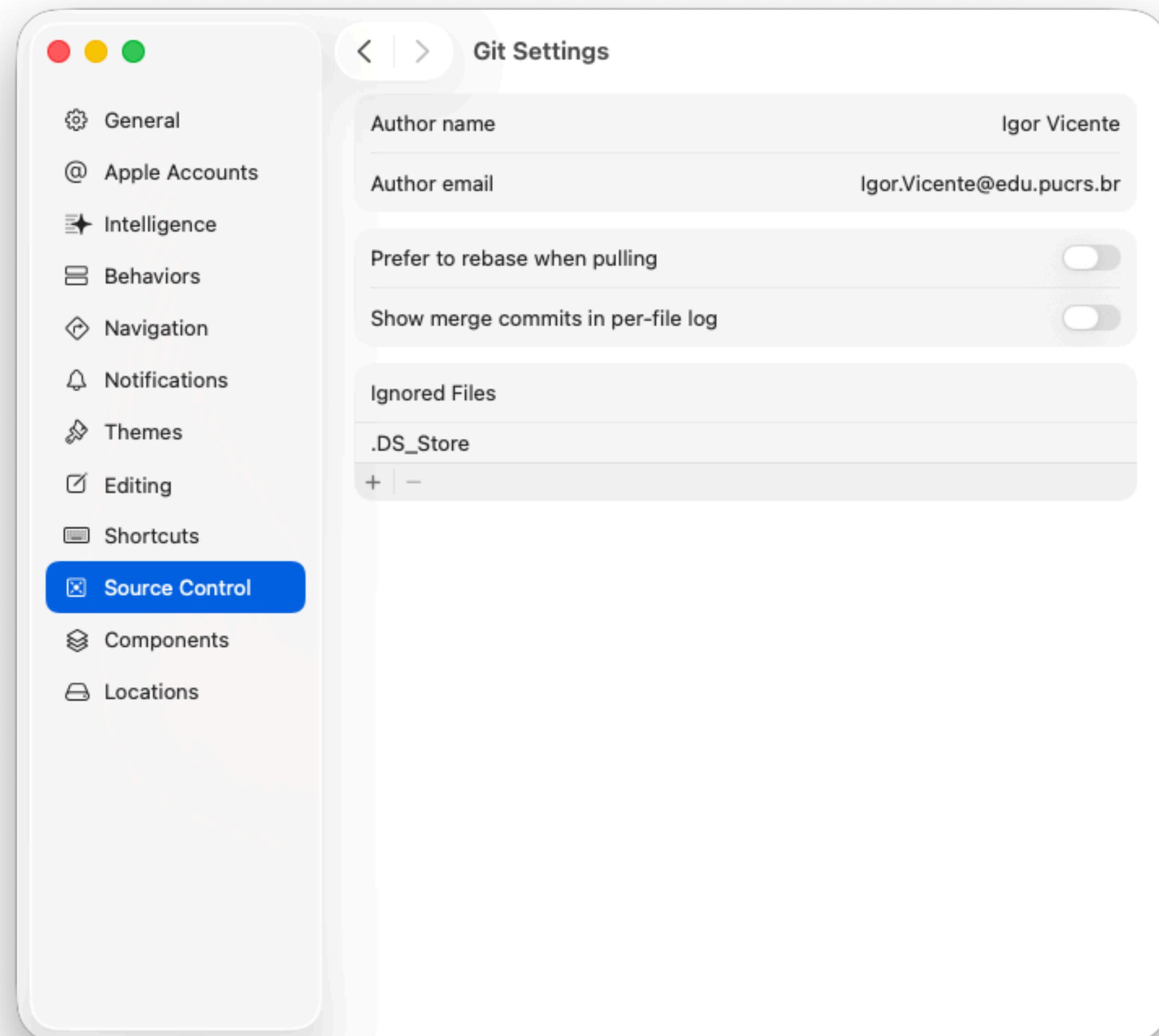


Terminal

# Configuration



Xcode



## Inicialização

git init

Na pasta raiz do projeto



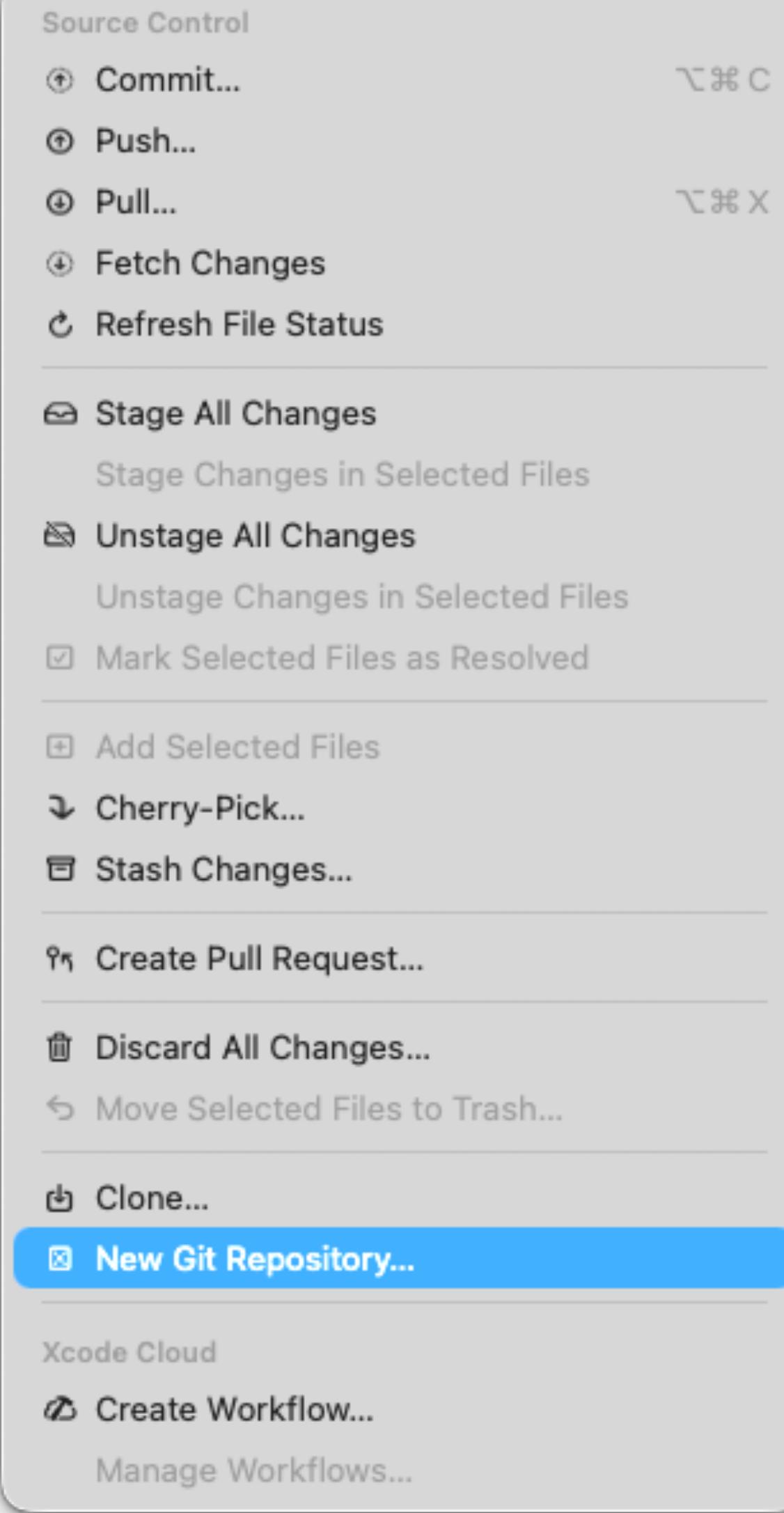
Terminal

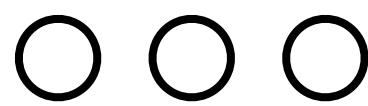


Xcode

Integrate

Create





## Fundamentos de Git

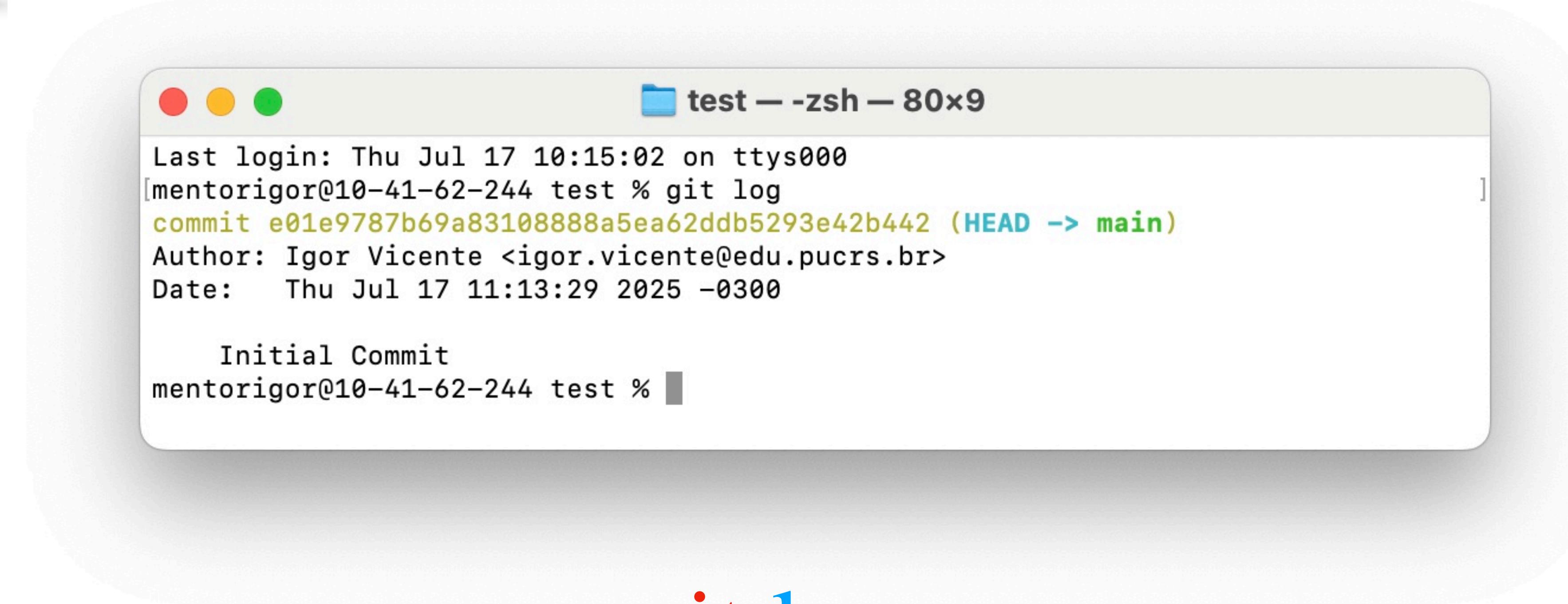


**Xcode fez automaticamente o primeiro commit**





# Como ver os commits no Terminal?



A screenshot of a macOS terminal window titled "test — -zsh — 80x9". The window shows the output of a "git log" command. The output includes:

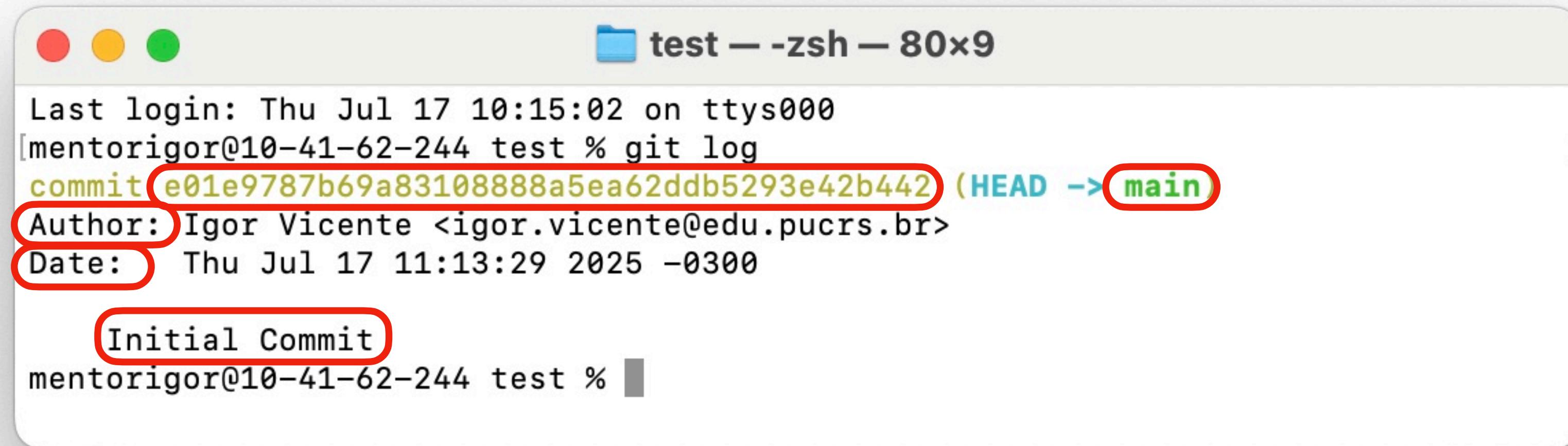
```
Last login: Thu Jul 17 10:15:02 on ttys000
[mentorigor@10-41-62-244 test % git log
commit e01e9787b69a83108888a5ea62ddb5293e42b442 (HEAD -> main)
Author: Igor Vicente <igor.vicente@edu.pucrs.br>
Date:   Thu Jul 17 11:13:29 2025 -0300

Initial Commit
mentorigor@10-41-62-244 test %
```

git log



# Como ver os commits no Terminal?



A screenshot of a macOS terminal window titled "test — -zsh — 80x9". The window shows the output of a "git log" command. The commit details are highlighted with red boxes:

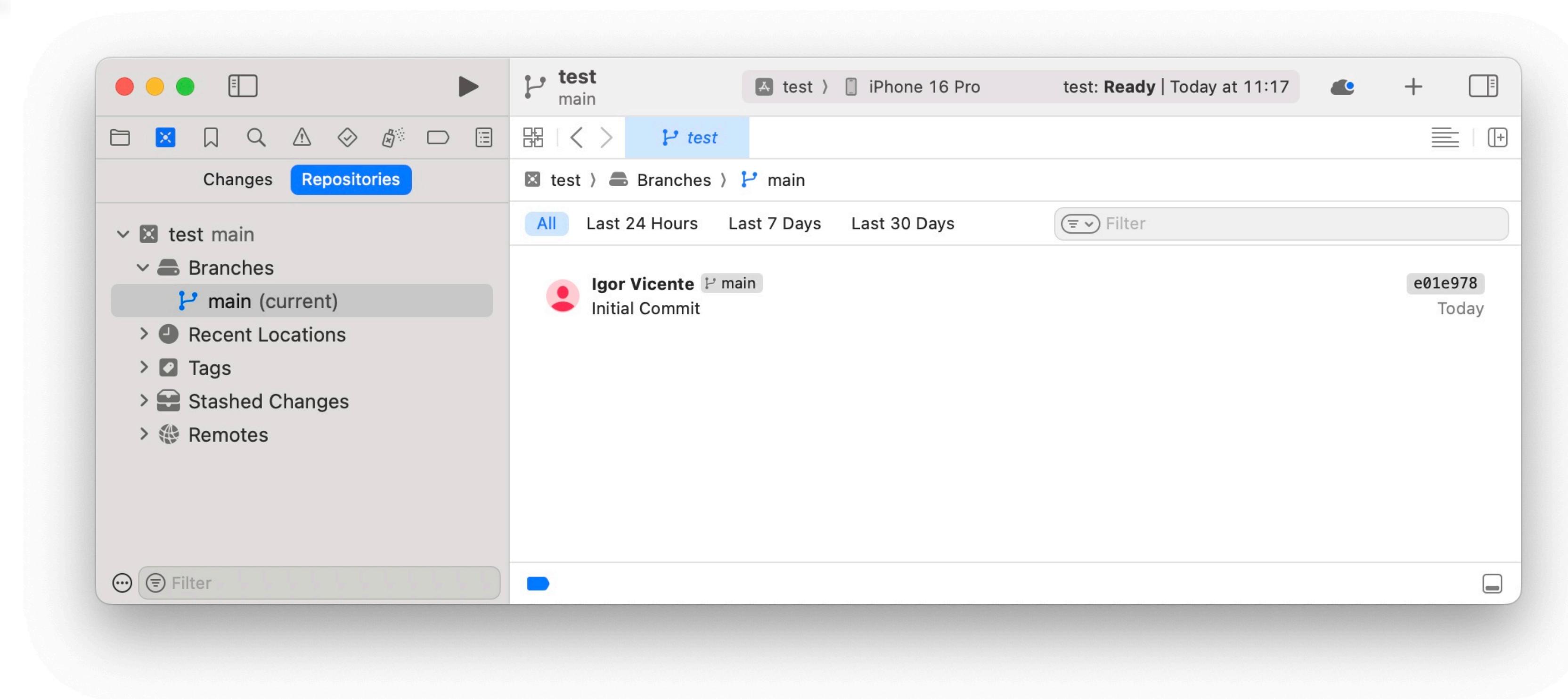
```
Last login: Thu Jul 17 10:15:02 on ttys000
[mentorigor@10-41-62-244 test % git log
commit e01e9787b69a83108888a5ea62ddb5293e42b442 (HEAD -> main)
Author: Igor Vicente <igor.vicente@edu.pucrs.br>
Date:   Thu Jul 17 11:13:29 2025 -0300

Initial Commit
mentorigor@10-41-62-244 test %
```

git log



## Como ver os commits no Xcode?



The screenshot shows the Xcode interface for managing Git repositories. The main window title is "test main". The repository path is "test > iPhone 16 Pro". The status bar indicates "test: Ready | Today at 11:17".

The sidebar on the left shows the project structure:

- test main
- Branches
  - main (current)
- Recent Locations
- Tags
- Stashed Changes
- Remotes

The main pane displays a commit history for the "main" branch. The first commit is highlighted with a red box around the author's name, "Igor Vicente".

Commit details:

- Author: Igor Vicente
- Date: Today
- Hash: e01e978
- Message: Initial Commit

A large red button labeled "Autor" is overlaid on the commit details.

At the bottom of the main pane, there is a "Filter" button.

The screenshot shows the Xcode interface with the Git tool open. The sidebar on the left displays navigation options like 'test main', 'Branches', and 'Recent Locations'. The main area shows a commit history for the 'main' branch. The first commit is from 'Igor Vicente' with the message 'Initial Commit'. A red box highlights this commit. A large red button labeled 'Descrição das alterações' (Description of changes) is overlaid on the commit details. The top status bar indicates the repository path 'test > iPhone 16 Pro', the status 'test: Ready | Today at 11:17', and various system icons.

test main

Changes Repositories

test main

Branches

main (current)

Recent Locations

Tags

Stashed Changes

Remotes

Filter

test > iPhone 16 Pro

test: Ready | Today at 11:17

All Last 24 Hours Last 7 Days Last 30 Days Filter

Igor Vicente main  
Initial Commit

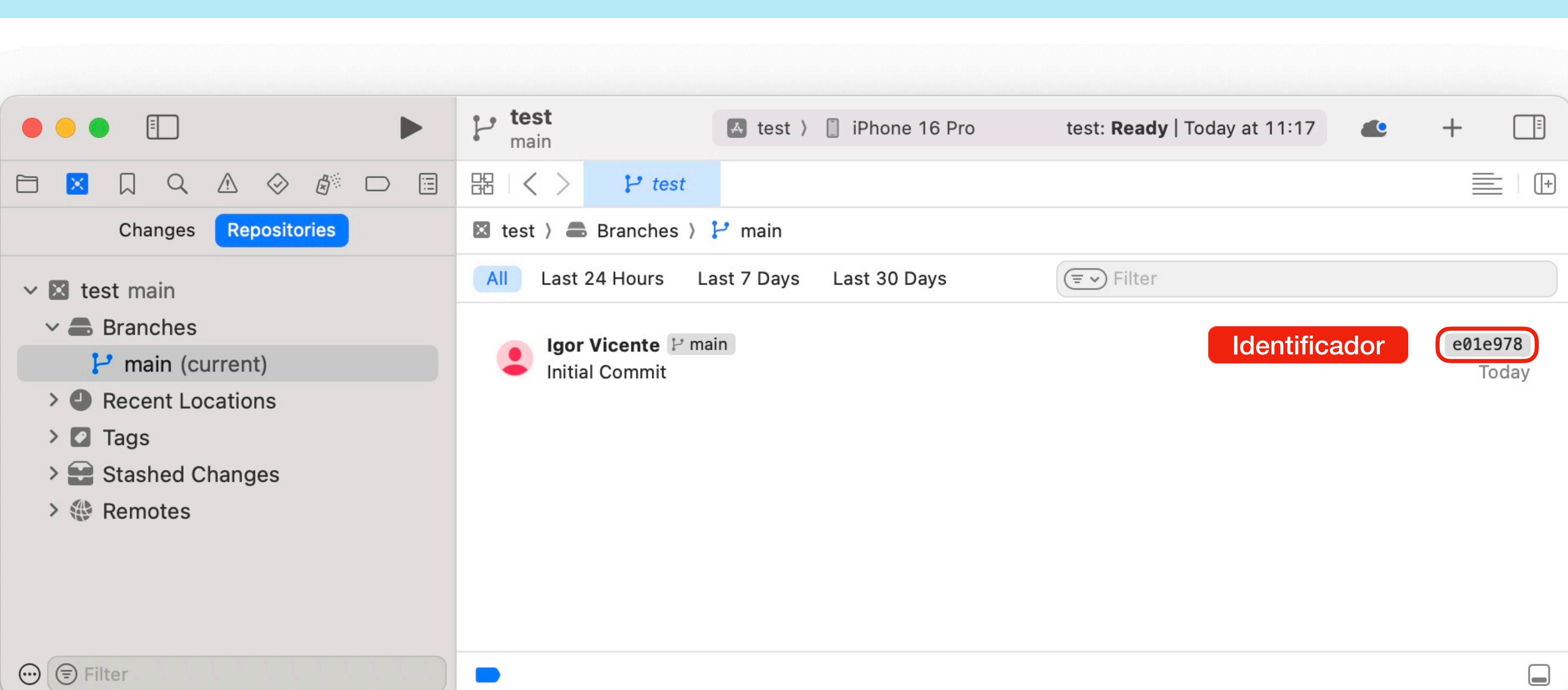
e01e978 Today

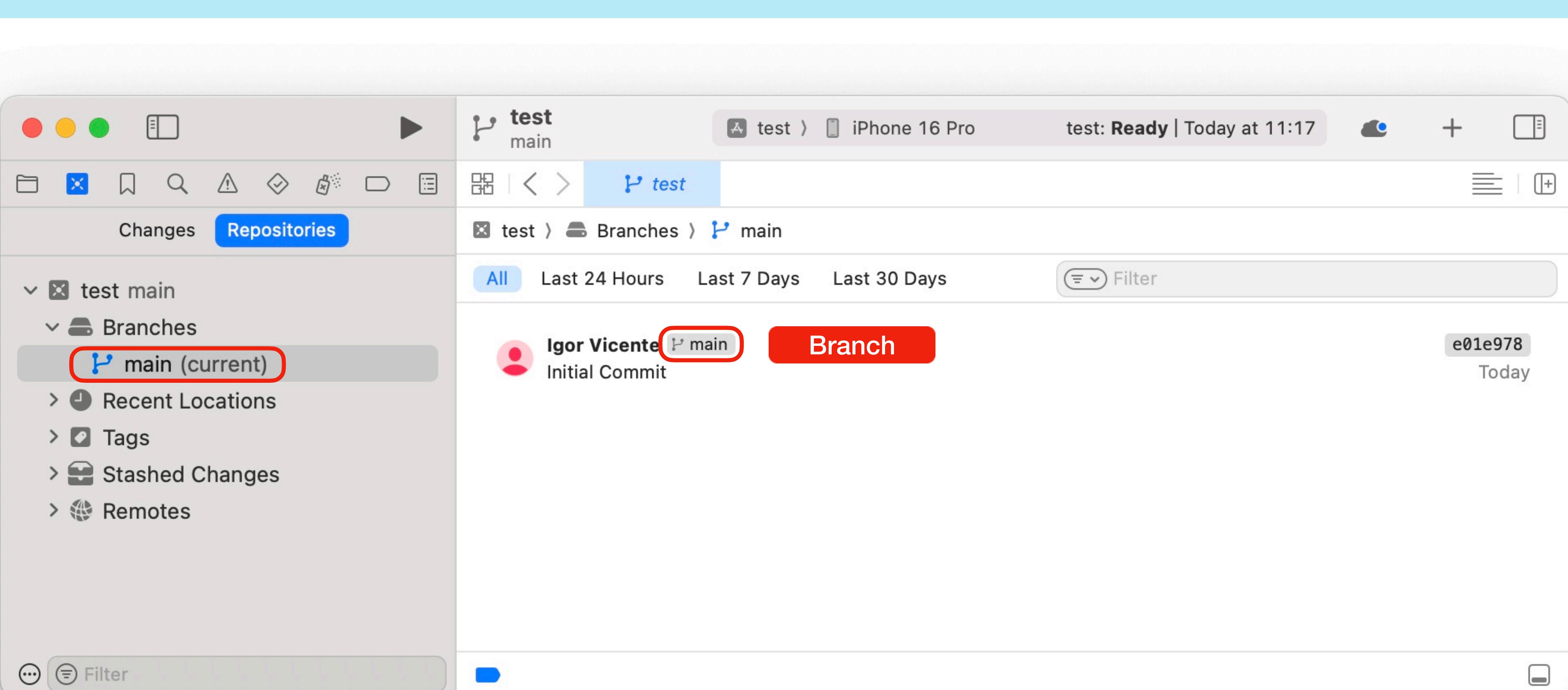
Descrição das alterações

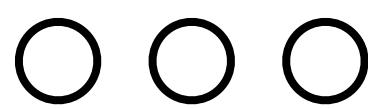
The screenshot shows the Xcode interface for a Git repository named "test". The repository has a single branch named "main". A commit from "Igor Vicente" was made "Today" at 11:17, with the commit hash being "e01e978". The commit message is "Initial Commit".

**Repository Navigation:**

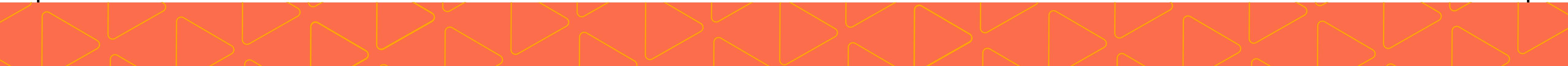
- Left sidebar: Changes, Repositories (selected), Recent Locations, Tags, Stashed Changes, Remotes.
- Top bar: Repository name "test", branch "main", device "iPhone 16 Pro", status "test: Ready | Today at 11:17".
- Central area: Branch selection dropdown showing "test" and "Branches", currently selected "main".
- Timeline: All, Last 24 Hours, Last 7 Days, Last 30 Days, Filter.
- Commit details:
  - User: Igor Vicente
  - Branch: main
  - Date: Today
  - Hash: e01e978
  - Message: Initial Commit
- Bottom right: Data button.







# Fundamentos de Git



# Fundamentos de Git



Working directory

Untracked || Modified



Staging area

Staged



Committed

# Fundamentos de Git



Area de trabalho

Não monitorado || Modificado



Preparados

Preparado



Prontos



# Como trabalhar com git no Terminal?

Working directory

Untracked || Modified

Staging area

Staged

Committed

git status → git add file → git commit -m “Descriptive message”  
← git rm --cached file



# Como trabalhar com git no Xcode?

Working directory 

Untracked || Modified

Staging area 

Staged

Committed 

test main

Changes Repositories

test main

Uncommitted Changes 2

HomeScreen R

testApp M

Igor Vicente Igor.Vicente@edu.pucrs.br

Amend "27a674e" Commit

Commit message (required)

All Changes Unstaged Staged Stage All

HomeScreen

```
3 // test
4 //
5 // Created by Igor Vicente on 17/07/25.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11     struct HomeScreen: View {
12         var body: some View {
13             VStack {
14                 Image(systemName: "globe")
15                     .imageScale(.large)
16                     .foregroundStyle(.tint)
17                 Text("Hello, world!")
18             }
19             .padding()
20             Text("Hello, world!")
21         }
22     }
23
24     #Preview {
25         ContentView()
26         HomeScreen()
27     }
28 }
```

testApp M

```
12     var body: some Scene {
13         WindowGroup {
14             ContentView()
15             HomeScreen()
16         }
17     }
18 }
```

Filter

The screenshot shows the Xcode interface with a focus on a git commit screen. On the left, the sidebar displays the repository structure: `test main`, `Uncommitted Changes 2` (containing `HomeScreen` and `testApp`), and a large red button labeled `git status`. The main area shows a commit screen for the `test` branch. The commit author is `Igor Vicente` (`Igor.Vicente@edu.pucrs.br`) with a placeholder commit message. The commit details show the creation of the `HomeScreen` file with the following code:

```
3 // test
4 //
5 // Created by Igor Vicente on 17/07/25.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11     struct HomeScreen: View {
12         var body: some View {
13             VStack {
14                 Image(systemName: "globe")
```

The screenshot shows the Xcode interface with a git commit message overlay.

**git commit message:**

```
git add HomeScreen.swift
```

**Xcode Status Bar:**

- test main
- Uncommitted Changes (2)
- HomeScreen
- testApp

**File Navigator:**

- Changes (selected)
- Repositories

**Current Branch:** test  
Main

**File Path:** test > HomeScreen > No Selection

**User Information:**

- Igor Vicente
- Igor.Vicente@edu.pucrs.br

**Stage Changes:** Stage Changes (required)

**Staged File Content:**

```
// ContentView.swift
// HomeScreen.swift
// test
// Created by Igor Vicente on 17/07/25.
//
import SwiftUI

struct ContentView: View {
```

The screenshot shows the Xcode interface with a commit message and a code editor.

**Commit Message:**

```
git rm --cached HomeScreen.swift
```

**Code Editor (HomeScreen.swift):**

```
1 // ContentView.swift
2 // HomeScreen.swift
3 // test
4 //
5 // Created by Igor Vicente on 17/07/25.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
```

**Git Changes:**

- test main
- Uncommitted Changes (2)
- HomeScreen ← ContentView
- testApp

**Repository:** test  
Branch: main

**File:** HomeScreen

**Author:** Igor Vicente  
Igor.Vicente@edu.pucrs.br

**Unstage Changes** (required)

**iPhone 16 Pro de Igor Vicente**

test: Ready | Today at 21:01



Amend "27a674e"



Commit | ▾

git add .

Stage All

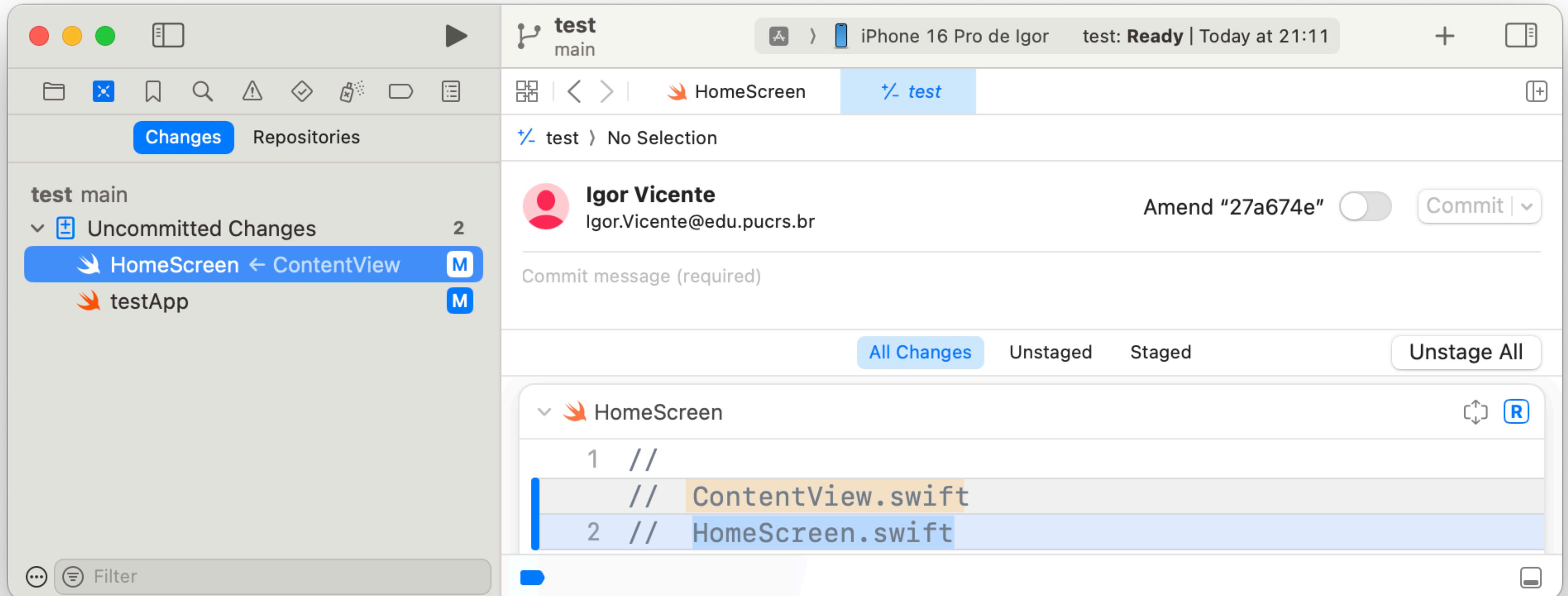
All Changes

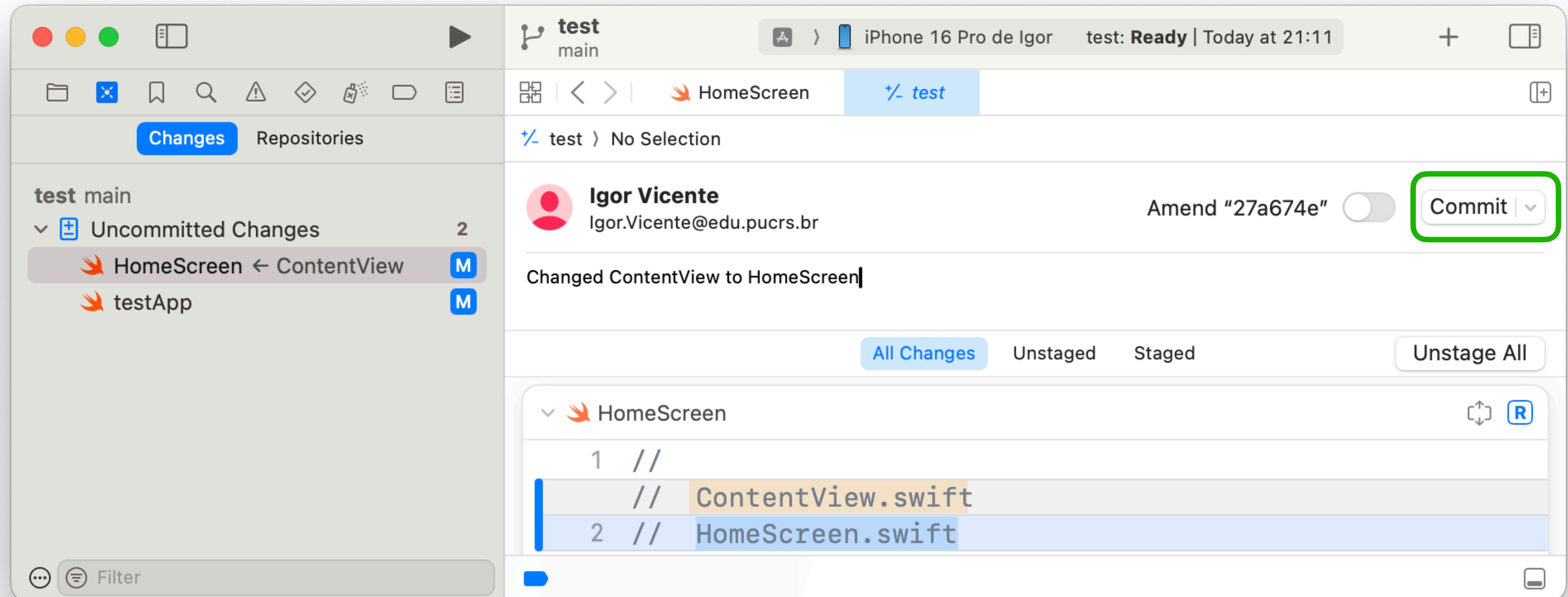
Unstaged

Staged



R





git commit -m “Changed ContentView to HomeScreen”



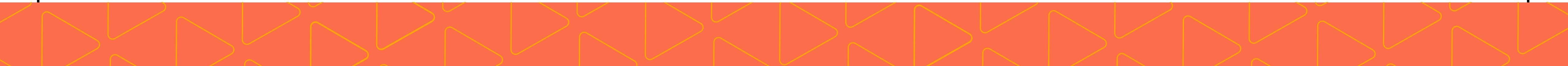
**Fundamentos de Git**

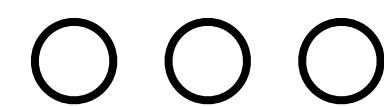
**GitHub**

**Branches & GitFlow**

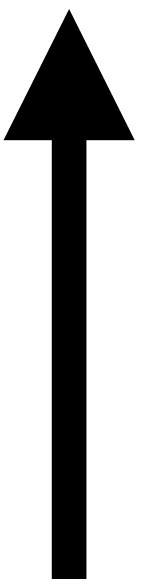


GitHub





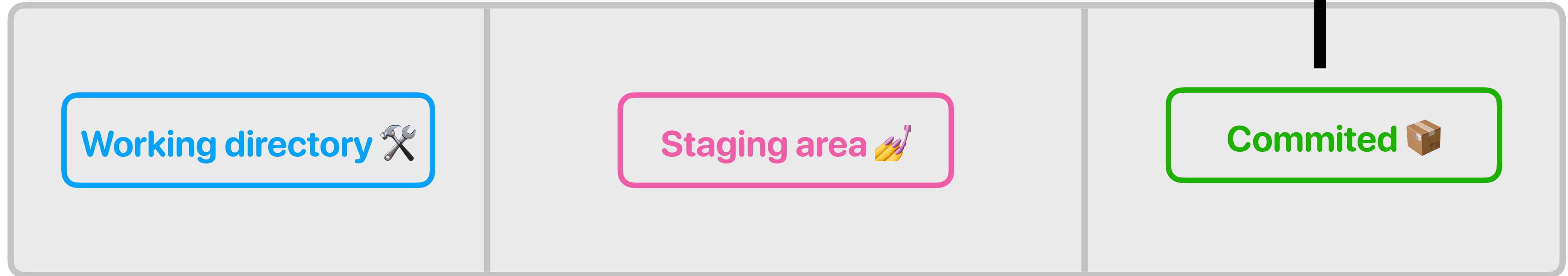
GitHub



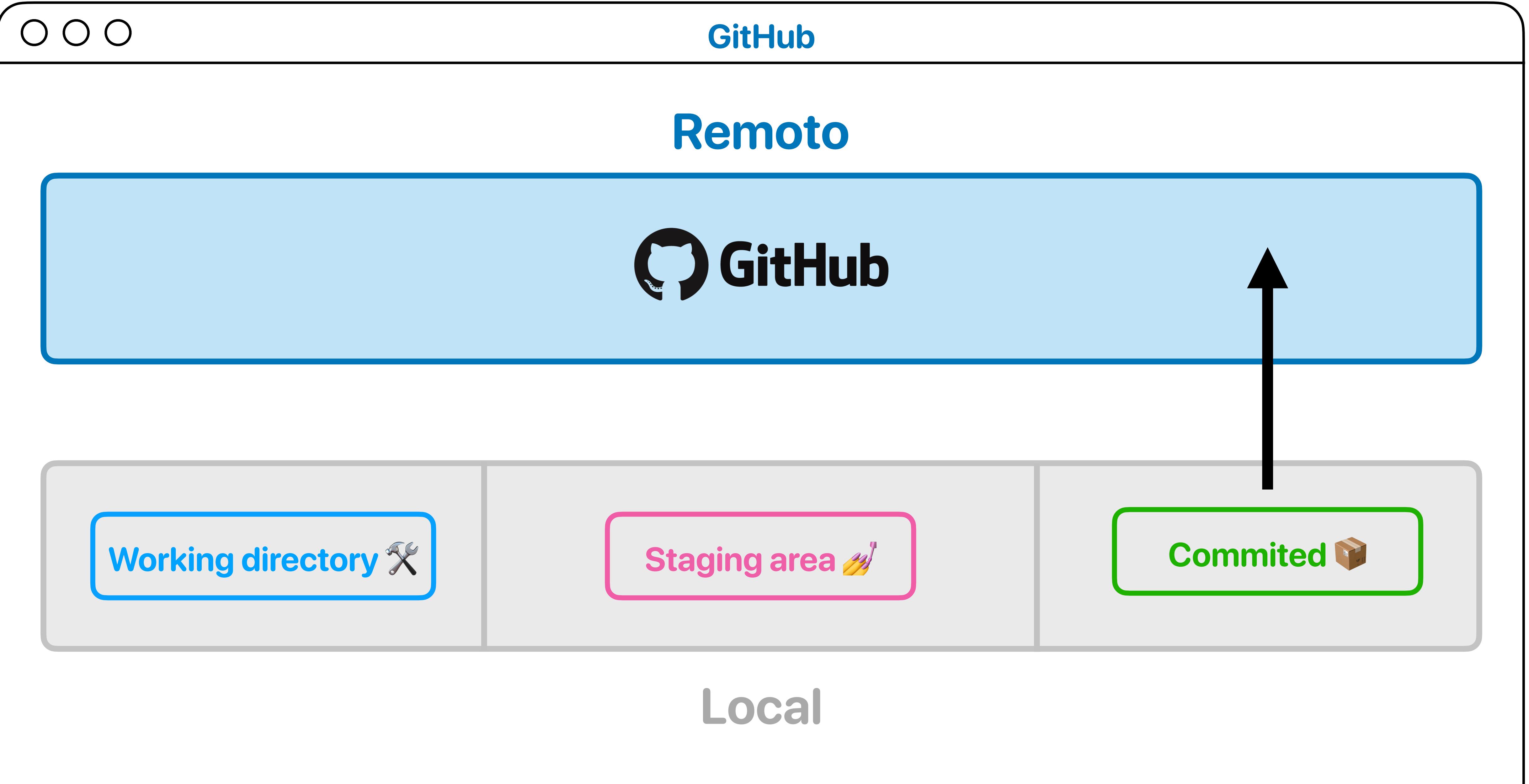


GitHub

# Setup



Local





## Remoto



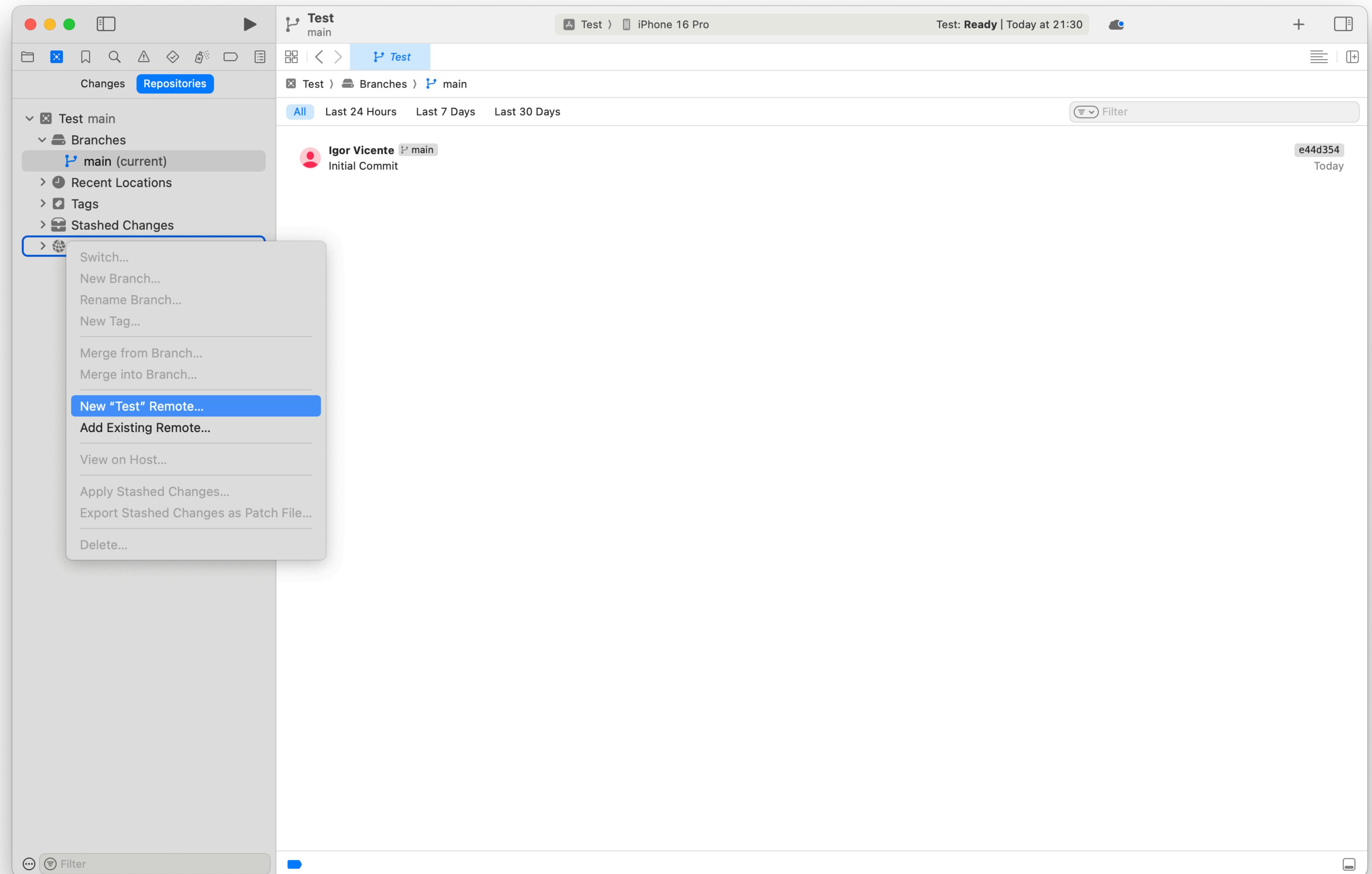
### COMO CONECTAR-SE DO REPOSITÓRIO LOCAL AO REMOTO?

Working directory

Staging area

Committed

Local



The screenshot shows the Xcode interface with the Git tool window open. The main area displays a repository named "Test" with a single commit from "Igor Vicente" on the "main" branch. The sidebar on the left shows the project structure with "Test main" expanded, and "Branches" selected. A context menu is open over the "main" branch, listing options like "Switch...", "New Branch...", "Rename Branch...", "New Tag...", "Merge from Branch...", "Merge into Branch...", "New 'Test' Remote...", "Add Existing Remote...", "View on Host...", "Apply Stashed Changes...", and "Export Stashed Changes as Patch File...".

Test  
main

Changes Repositories

Test main

Branches

main (current)

Recent Locations

Tags

Stashed Changes

Switch...

New Branch...

Rename Branch...

New Tag...

Merge from Branch...

Merge into Branch...

New "Test" Remote...

Add Existing Remote...

View on Host...

Apply Stashed Changes...

Export Stashed Changes as Patch File...

All Last 24 Hours Last 7 Days Last 30 Days

Igor Vicente main  
Initial Commit

Test > Branches > main

Test iPhone 16 Pro

Branches

 main (current)

Recent Locations

Tags

Stashed Changes

Remotes

 Igor Vicente  main

Initial Commit

e44

**Create "Test" remote:**

Account: Select an Account

Owner: Personal

Repository Name: Test

Select an Account

Description: Optional

Visibility:  Public

Anyone can see this repository. You choose who can commit.

 Private

You can choose who can see and commit to this repository.

Remote Name: origin

Cancel

Create

Branches

 main (current)

Recent Locations

Tags

Stashed Changes

Remotes

 Igor Vicente  main

Initial Commit

e44

## Create "Test" remote:

Account: Add an account...

Select the type of account you would like to add:



Bitbucket Cloud



Bitbucket Server



GitHub



GitHub Enterprise

Cancel

Continue

Cancel

Create

### Create "Test" remote:

Account:

#### Sign in to your GitHub account

Account:

Token:

**GitHub personal access tokens must have these scopes set:**

- ✓ admin:public\_key
- ✓ write:discussion
- ✓ repo
- ✓ user

[Create a Token on GitHub](#)

[Cancel](#)

[Sign In](#)

Remote Name:

[Cancel](#)

[Create](#)

<input checked="" type="checkbox"/> <b>repo</b>	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> <b>workflow</b>	Update GitHub Action workflows
<input type="checkbox"/> <b>write:packages</b>	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> <b>delete:packages</b>	Delete packages from GitHub Package Registry
<input type="checkbox"/> <b>admin:org</b>	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> manage_runners:org	Manage org runners and runner groups
<input checked="" type="checkbox"/> <b>admin:public_key</b>	Full control of user public keys
<input checked="" type="checkbox"/> write:public_key	Write user public keys
<input checked="" type="checkbox"/> read:public_key	Read user public keys
<input type="checkbox"/> <b>admin:repo_hook</b>	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks
<input type="checkbox"/> read:repo_hook	Read repository hooks
<input type="checkbox"/> <b>admin:org_hook</b>	Full control of organization hooks
<input type="checkbox"/> <b>gist</b>	Create gists
<input type="checkbox"/> <b>notifications</b>	Access notifications
<input checked="" type="checkbox"/> <b>user</b>	Update ALL user data
<input checked="" type="checkbox"/> read:user	Read ALL user profile data
<input checked="" type="checkbox"/> user:email	Access user email addresses (read-only)
<input checked="" type="checkbox"/> user:follow	Follow and unfollow users
<input type="checkbox"/> <b>delete_repo</b>	Delete repositories
<input checked="" type="checkbox"/> <b>write:discussion</b>	Read and write team discussions
<input checked="" type="checkbox"/> read:discussion	Read team discussions
<input type="checkbox"/> <b>admin:enterprise</b>	Full control of enterprises
<input type="checkbox"/> manage_runners:enterprise	Manage enterprise runners and runner groups
<input type="checkbox"/> manage_billing:enterprise	Read and write enterprise billing data
<input type="checkbox"/> read:enterprise	Read enterprise profile data
<input type="checkbox"/> scim:enterprise	Provisioning of users and groups via SCIM
<input type="checkbox"/> <b>audit_log</b>	Full control of audit log

### Create "Test" remote:

Account:

#### Sign in to your GitHub account

Account:

Token:

##### GitHub personal access tokens must have these scopes set:

- ✓ admin:public\_key
- ✓ write:discussion
- ✓ repo
- ✓ user

[Create a Token on GitHub](#)

Cancel

Sign In

Remote Name:

Cancel

Create

### Create "Test" remote:

Account: iMarVic (GitHub)

Owner: iMarVic

Repository Name: SwitchTaills example

<https://github.com/iMarVic/SwitchTaills%20example>

Description: Optional

Visibility:  Public

Anyone can see this repository. You choose who can commit.

Private

You can choose who can see and commit to this repository.

Remote Name: origin

Cancel

Create



iMarVic / SwitchTails-example

Type / to

Code

Issues

Pull requests

Actions

Projects

Security

Insights

Settings



# SwitchTails-example

Private

Watch 0

main ▾

1 Branch 0 Tags

Go to file

t

Add file ▾

Code ▾



iMarVic Initial Commit

e44d354 · 9 minutes ago

1 Commit

Test.xcodeproj

Initial Commit

9 minutes ago

Test

Initial Commit

9 minutes ago

README



## Add a README

Add a README with an overview of your project.

iMarVic / SwitchTails-example

Type / to search

Code Issues Pull requests Actions Projects Security Insights Settings

General

Access

Collaborators

Private repository

Only those with access to this repository can view it

Manage visibility

Code and automation

Branches Tags Rules Actions Models Webhooks Copilot Environments Codespaces Pages

Preview

Direct access

0 collaborators have access to this repository. Only you can contribute to this repository.

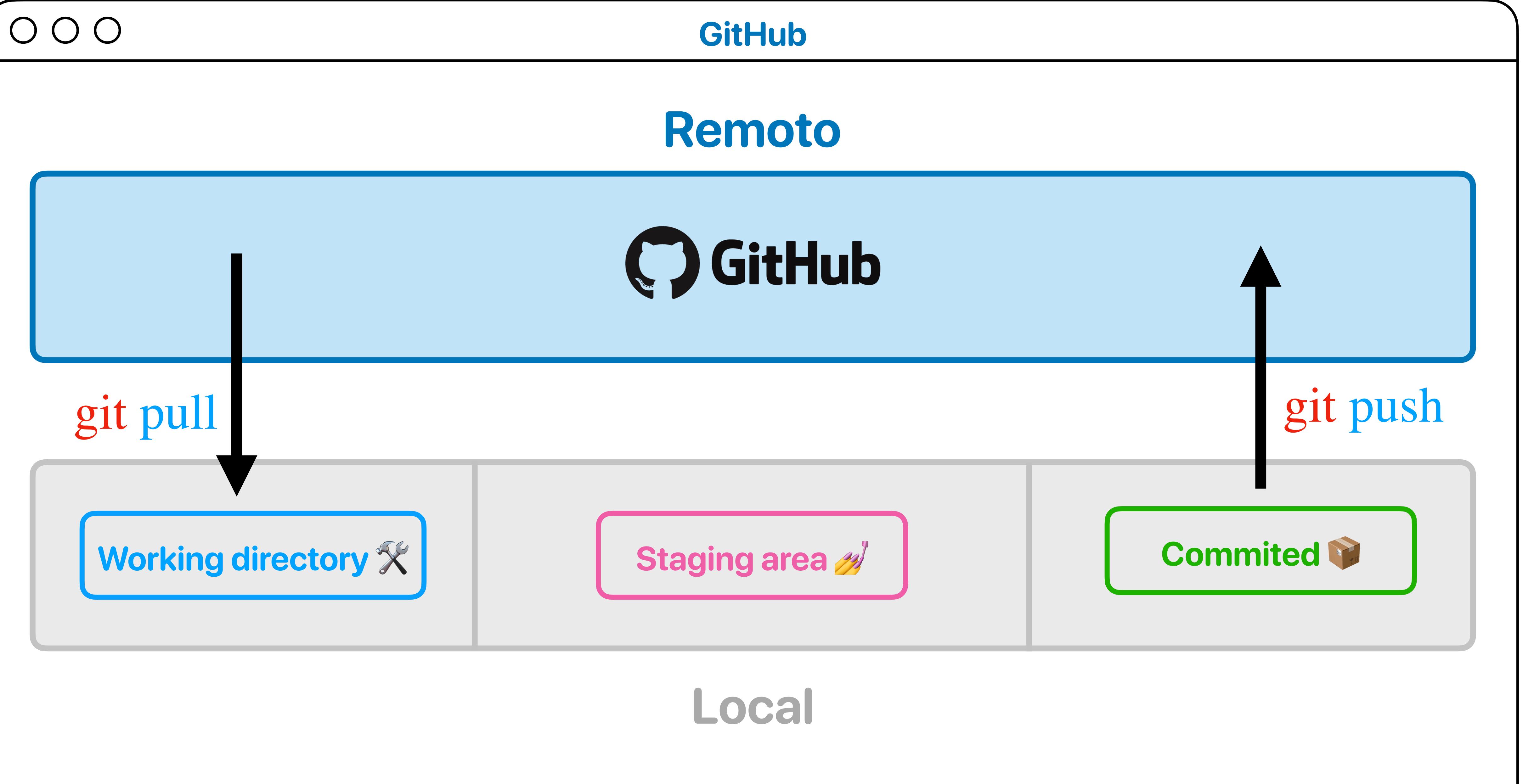
Manage access

You haven't invited any collaborators yet

Add people



The screenshot shows the GitHub repository settings page for "SwitchTails-example". The "Settings" tab is selected. On the left, there's a sidebar with sections like General, Access (which is highlighted with a red box), Code and automation, Security, and more. In the main area, under "Access", the "Collaborators" section is active. It shows that there are currently 0 collaborators. A large pink box highlights the "Add people" button at the bottom of the "Manage access" section. Another pink box highlights the "Collaborators" link in the sidebar. To the right of the main content, there's a large pink box containing the names "iMarVic", "Elsner2002", "Eduardo-Riboli", and "dharanarivas", likely representing users who could be invited as collaborators.



test main

iPhone 16 Pro de Igor

test: Ready | Today at 21:11

Changes Repositories

% test > No Selection

Igor Vicente  
Igor.Vicente@edu.pucrs.br

Amend "27a674e" Commit ▾

Changed ContentView to HomeScreen

All Changes Unstaged Staged Unstage All

HomeScreen

```
1 //  
2 // ContentView.swift  
3 // HomeScreen.swift  
4 // test  
5 // Created by Igor Vicente on 17/07/25.  
6 //  
7  
8 import SwiftUI  
9  
10 struct ContentView: View {  
11 struct HomeScreen: View {  
12     var body: some View {  
13         VStack {  
14             Image(systemName: "globe")  
15                 .imageScale(.large)  
16                 .foregroundStyle(.tint)  
17             Text("Hello, world!")  
18         }  
19         .padding()  
20         Text("Hello, world!")  
21     }  
22 }  
23  
24 #Preview {  
25     ContentView()  
26     HomeScreen()  
27 }  
28  
29  
30 testApp
```

Filter

```
12     var body: some Scene {  
13         WindowGroup {  
14             ContentView()  
15             HomeScreen()
```

test: **Ready** | Today at 21:11



Amend "27a674e"



Commit ▾

aged

Staged

Unstage All



R

test: **Ready** | Today at 21:11



Amend "27a674e"



Commit



Commit and Push...

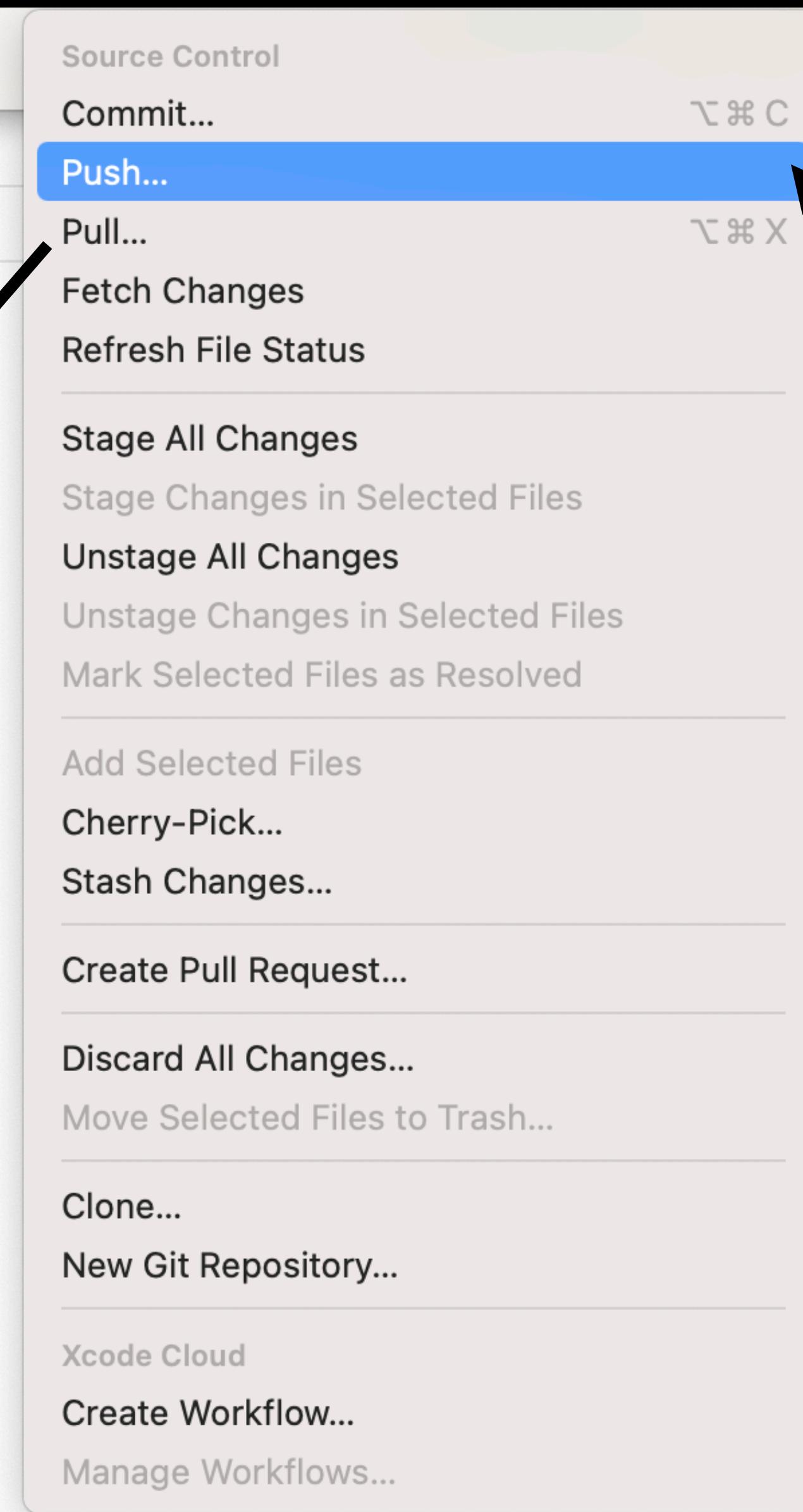
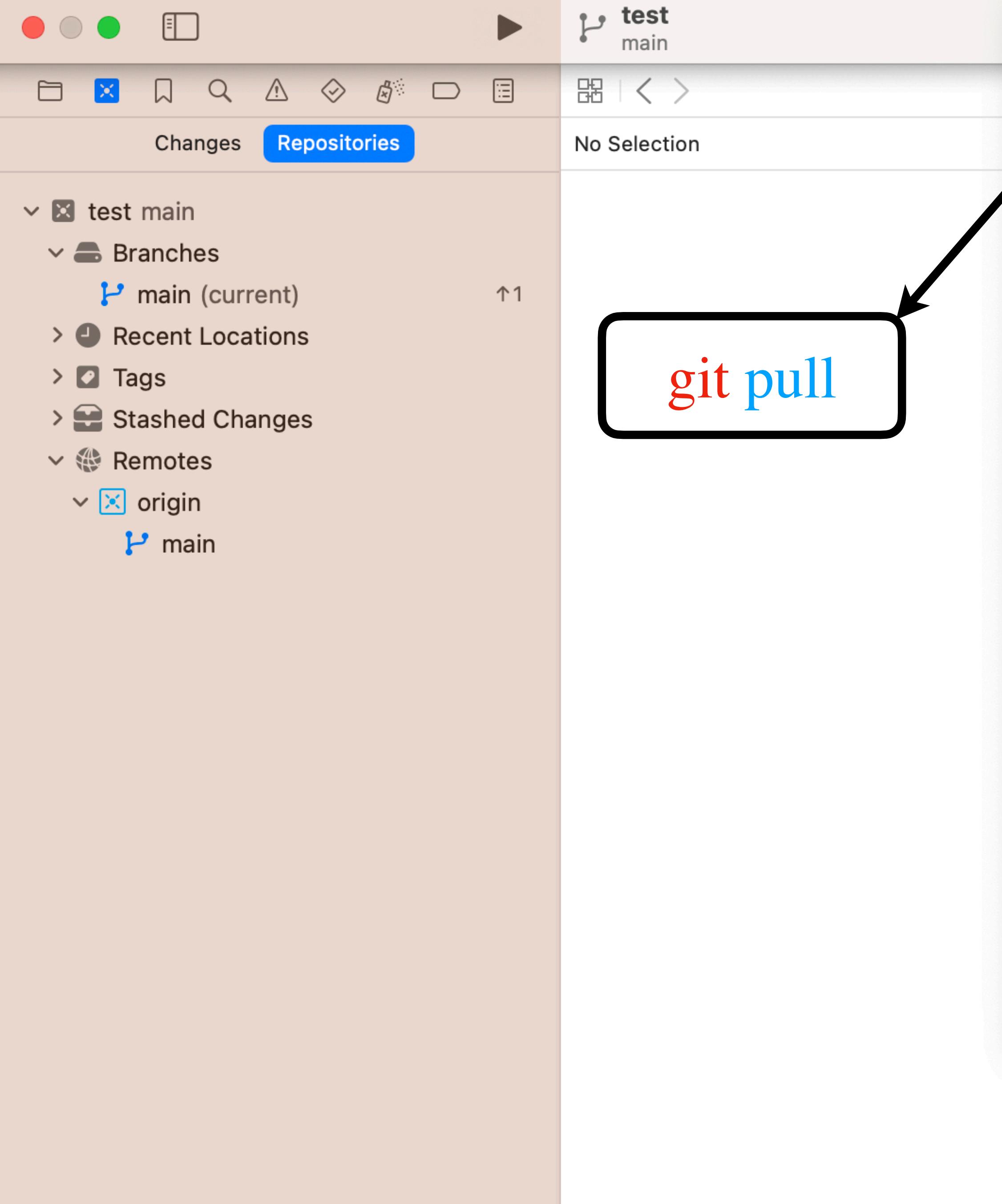
aged

Staged

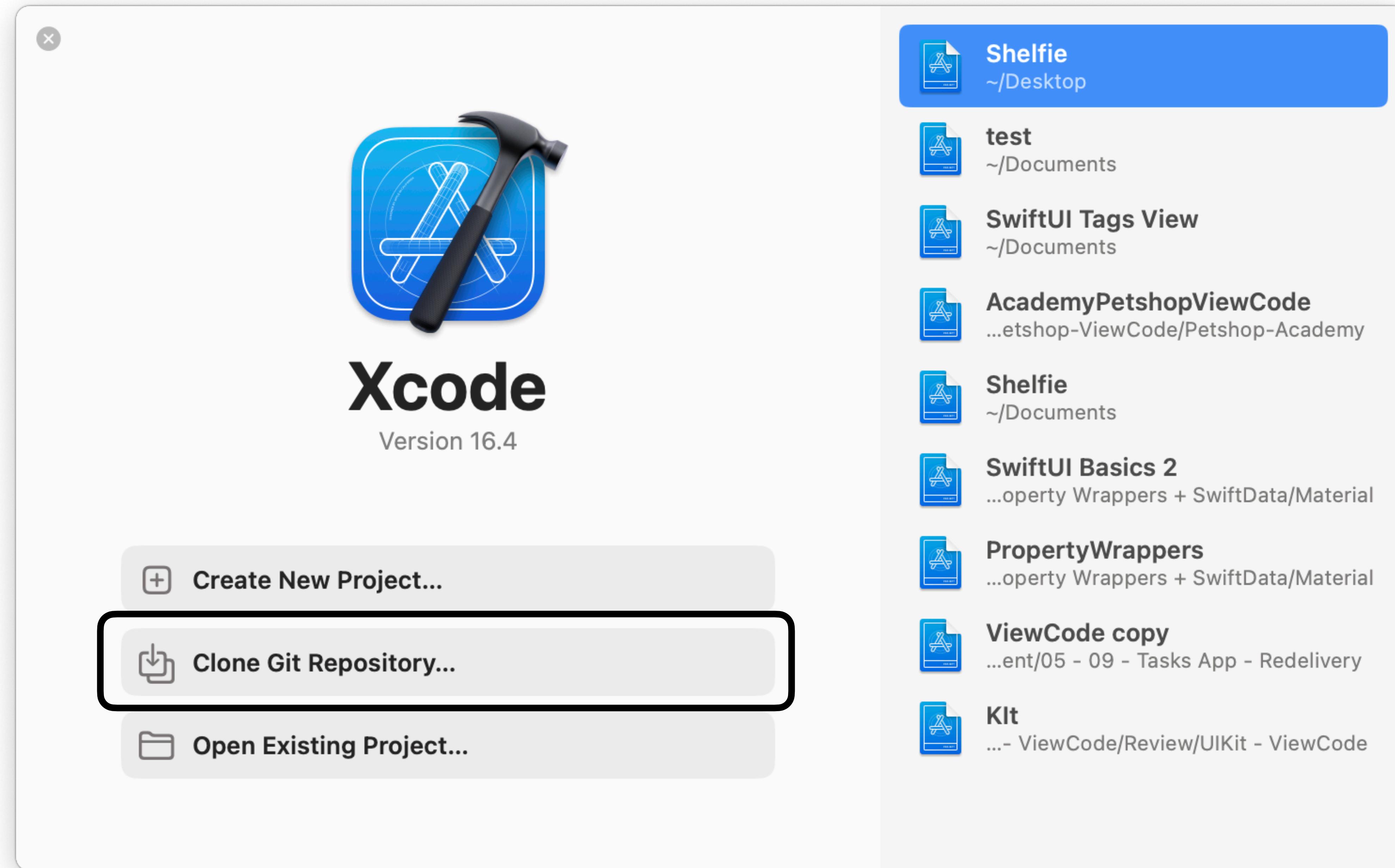
Unstage All



R



No Editor





Search or enter repository URL

Name	Last Updated	Owner
iMarVic (GitHub)		
 CircularProgressView-Tutorial	21 Jun 2018, 12:36	leoiphonedev
 CodeExamples	10 Feb 2025, 11:59	kharrison
 CustomTabBar	11 Jan 2021, 03:43	usmanmukhtar
 Kin	8 May 2025, 13:42	Serchinastico
 Shelfie	Today, 21:42	Academy-POA
 SkrivaForHome	24 Oct 2024, 16:27	DiegoHSO
 wwdc20-submission	18 May 2020, 07:10	marlon360
 BasicOnboarding-UIPageViewController	26 Nov 2021, 17:24	iMarVic
 Core-ML-Hot-N-Cold	9 May 2022, 17:21	iMarVic
 Distribuida-T2-2023	10 Jun 2023, 16:45	iMarVic
 DontFall	16 Oct 2023, 17:56	gabizorzo
 DontFall	28 Nov 2023, 21:24	iMarVic



Done

Clone



**Fundamentos de Git**



**GitHub**

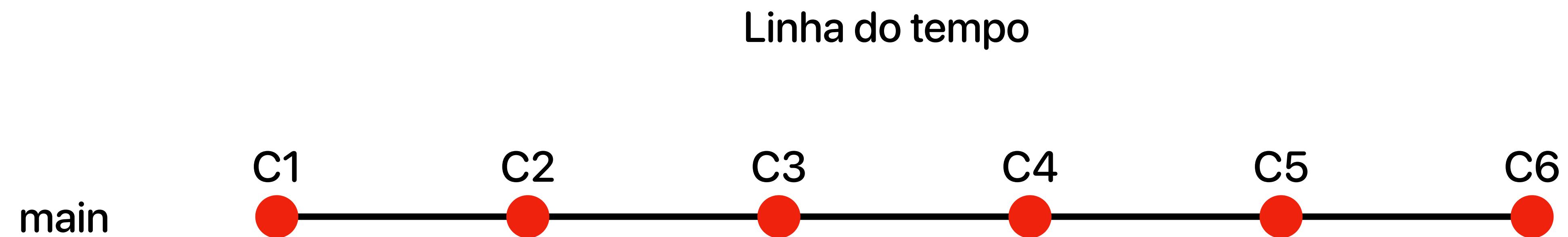
**Branches & GitFlow**

## Branches & GitFlow

Linha do tempo



## Branches & GitFlow

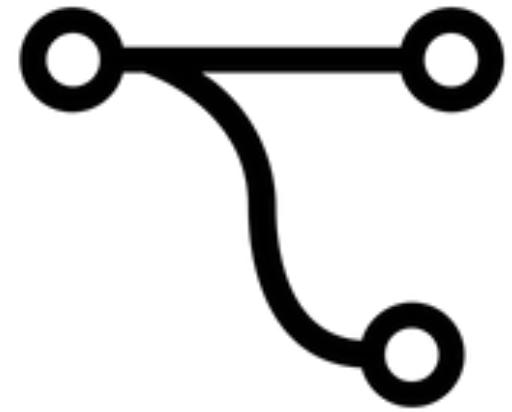
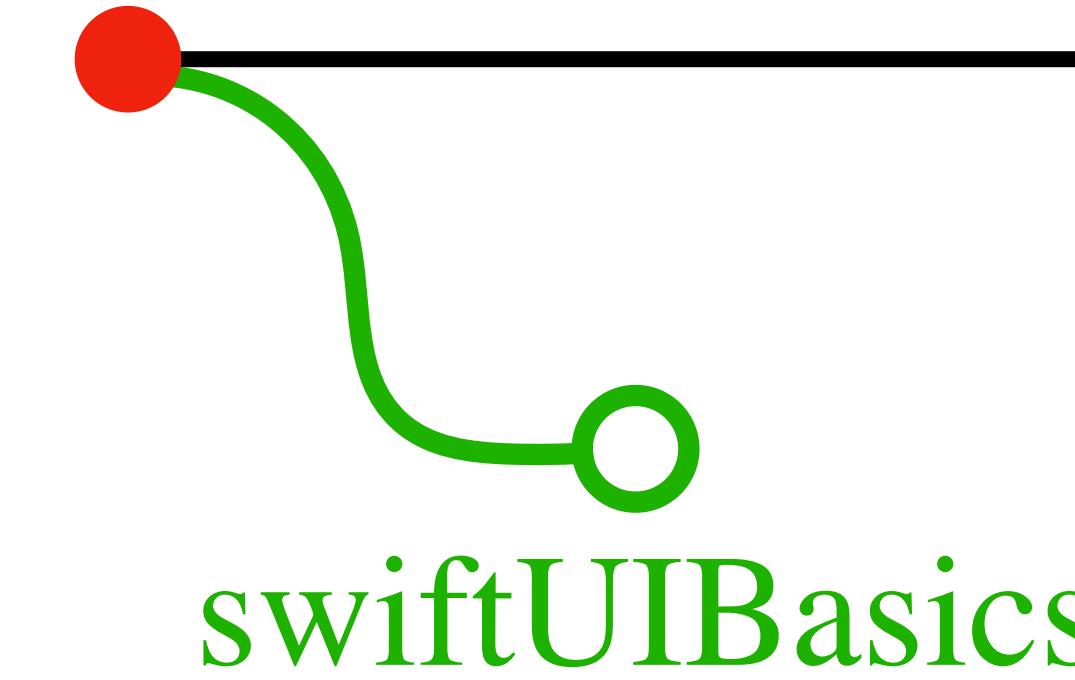




## Branches & GitFlow

Cria uma branch  
a partir de outra

main



git branch newBranch

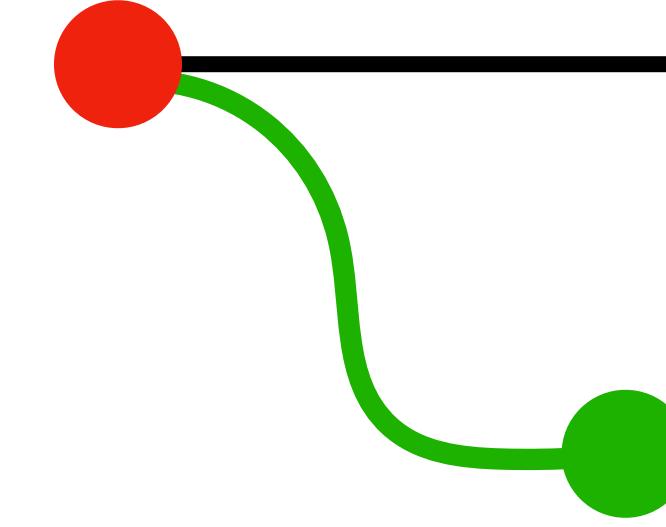
git branch swiftUIBasics



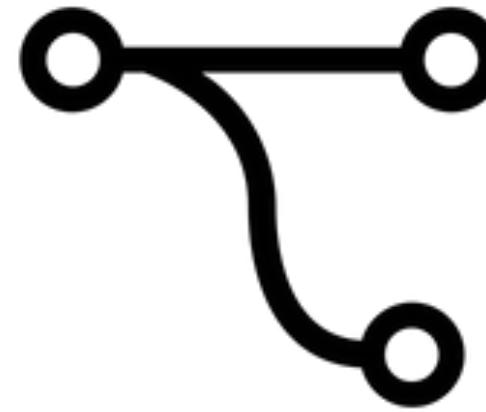
## Branches & GitFlow

Vai para uma branch

main



swiftUIBasics



git checkout newBranch



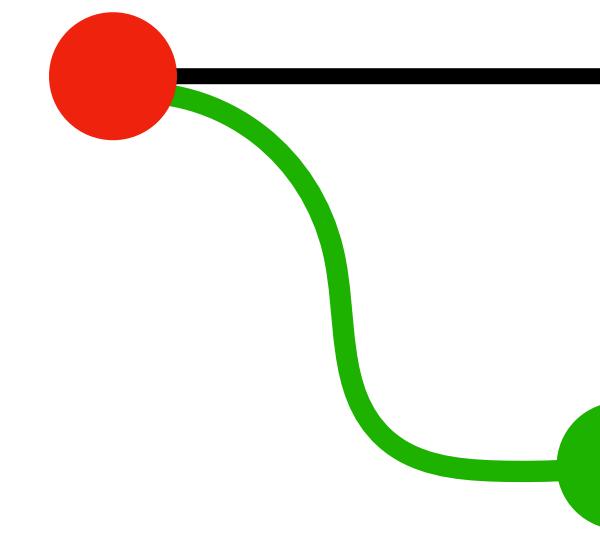
git checkout swiftUIBasics



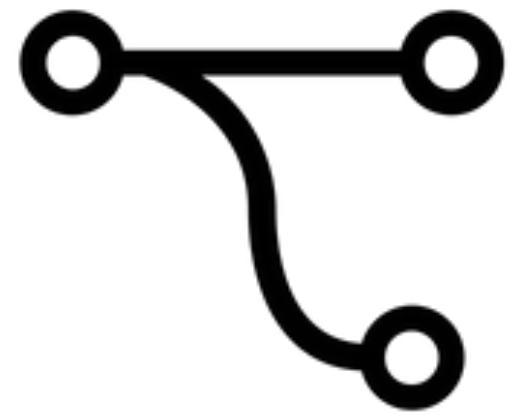
## Branches & GitFlow

**Cria uma branch  
a partir de outra  
&  
Vai para a nova branch**

main



swiftUIBasics



**git checkout -b newBranch**



**git checkout -b swiftUIBasics**



test  
main

test > iPhone 16 Pro de Igor



Changes    **Repositories**

✓ test main

Branches

**main (current)**

> Recent Locations

> Tags

> Stashed Changes

✓ Remotes

> origin

main

grid < > **test**

test > Branches > main

All    Last 24 Hours    Last 7 Days    Last 30 Days

Igor Vicente origin/main main

HomeScreen

Switch...

**New Branch from "main"...**

Rename "main"...

Tag "main"...

Merge from Branch...

Merge into Branch...

New "test" Remote...

Add Existing Remote...

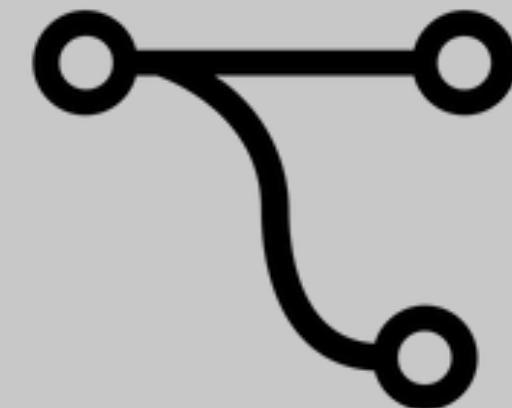
View on GitHub...

Apply Stashed Changes...

Export Stashed Changes as Patch File...

Delete...

**Cria uma branch  
a partir de outra  
&  
Vai para a nova branch**



`git checkout -b newBranch`

**Create new branch in "test":**

Create a branch from the current branch and switch to it. All uncommitted changes will be preserved on the new branch.

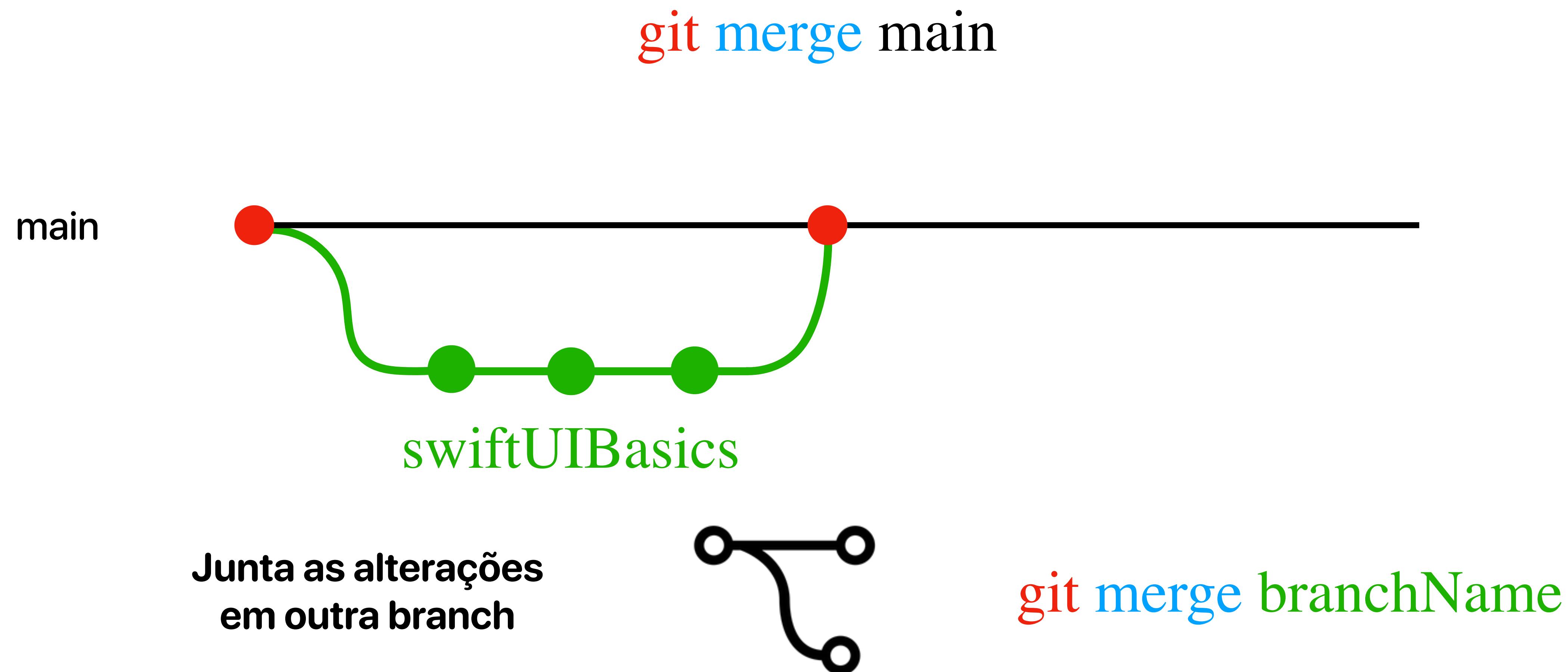
From: main

To:

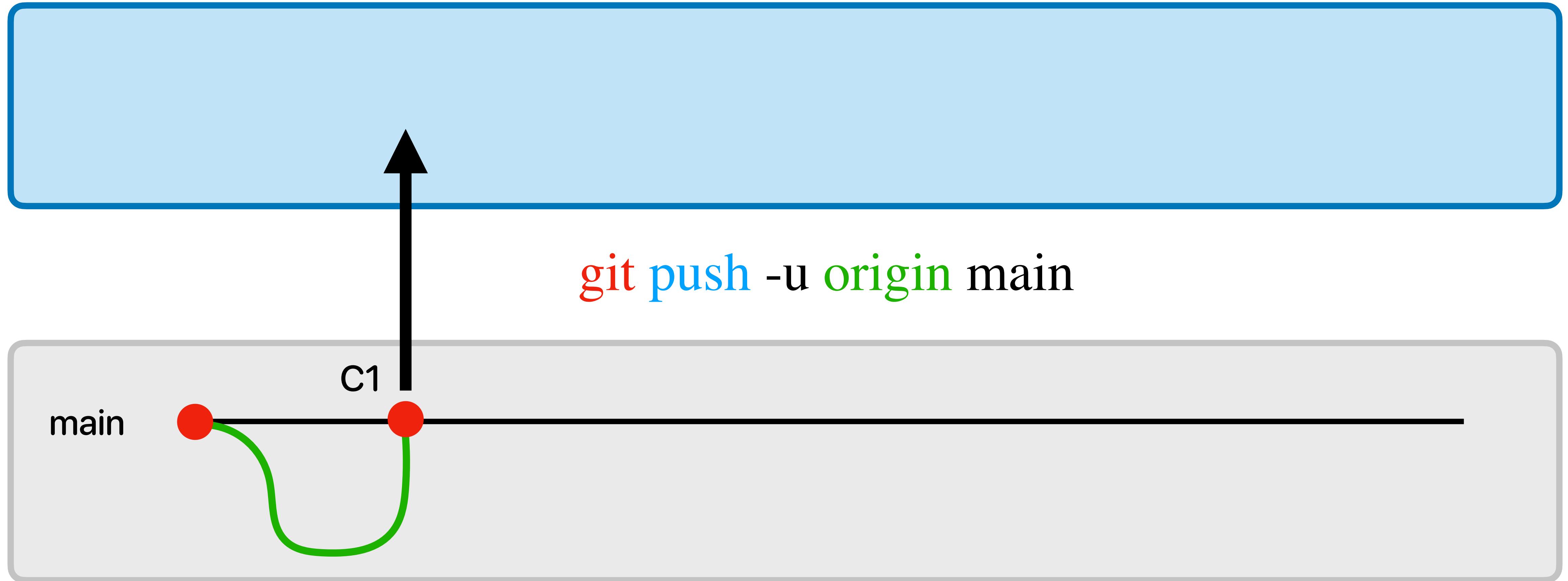
**Cancel**

**Create**

## Branches & GitFlow

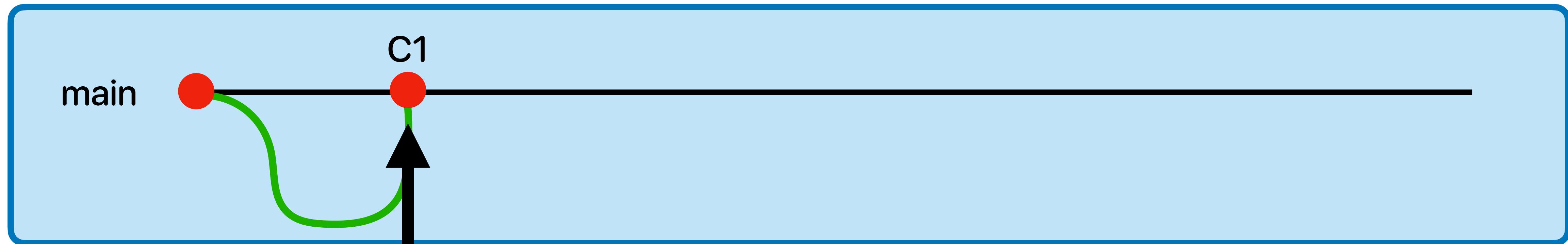


## Remote

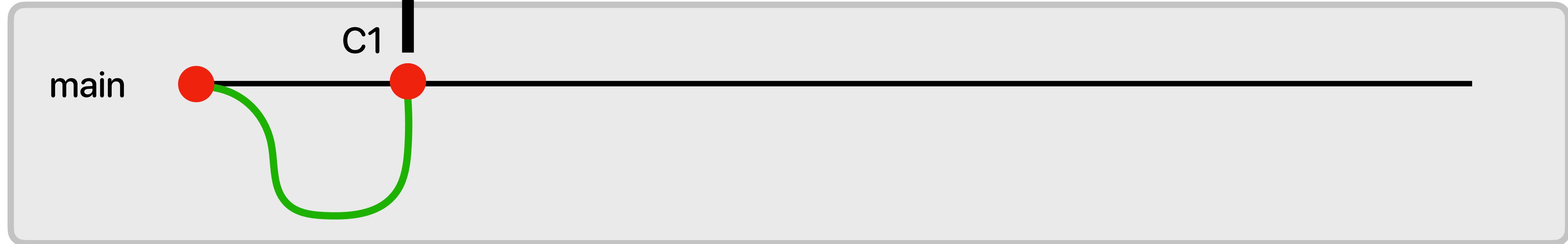


Local

## Remote



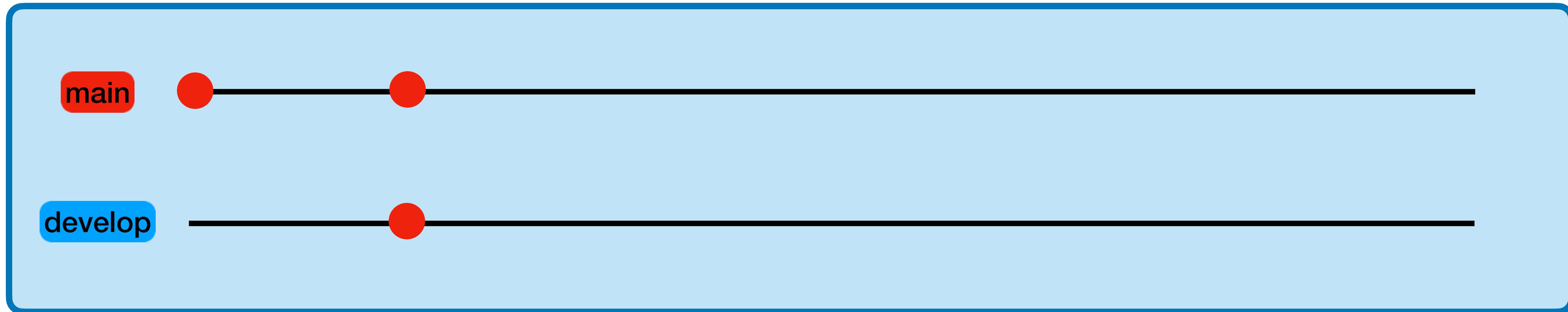
`git push -u origin main`



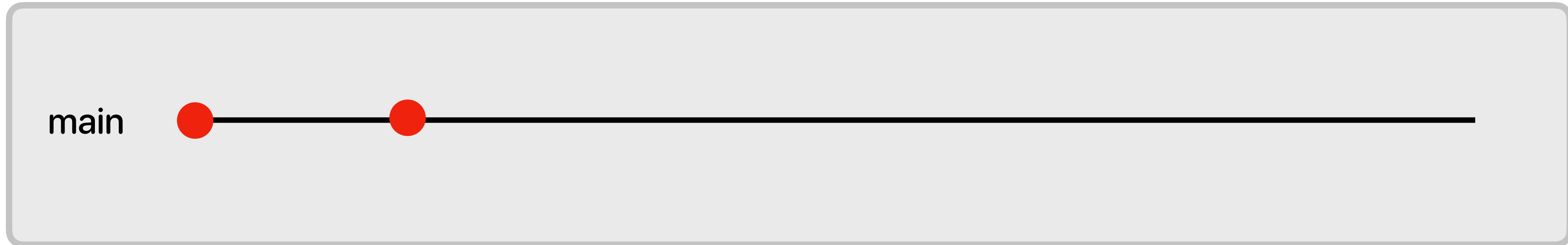
Local

## Branches & GitFlow

### Remote

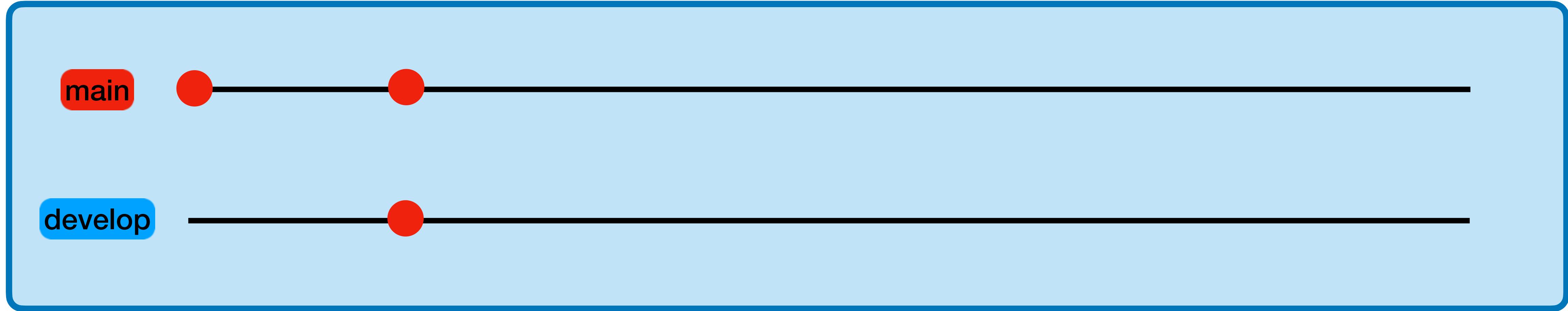


git fetch ➔ main develop

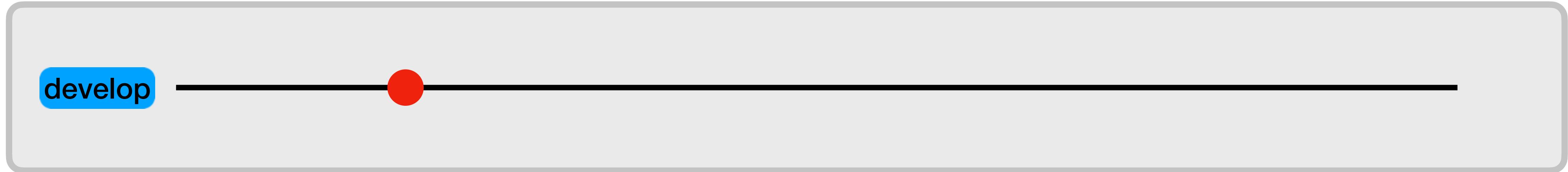


## Branches & GitFlow

### Remote

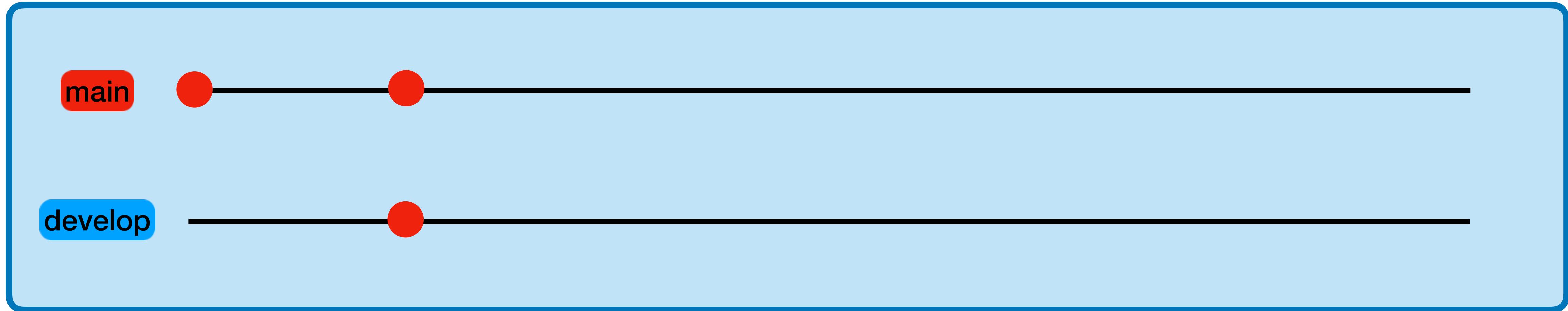


git fetch ➔ main develop  
git checkout develop

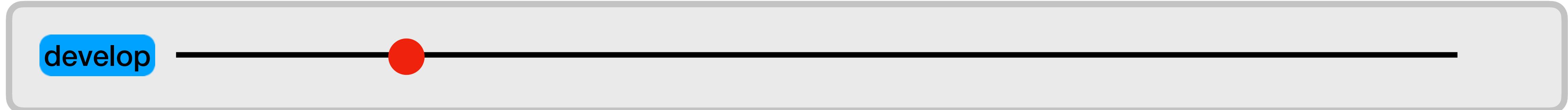


## Branches & GitFlow

### Remote

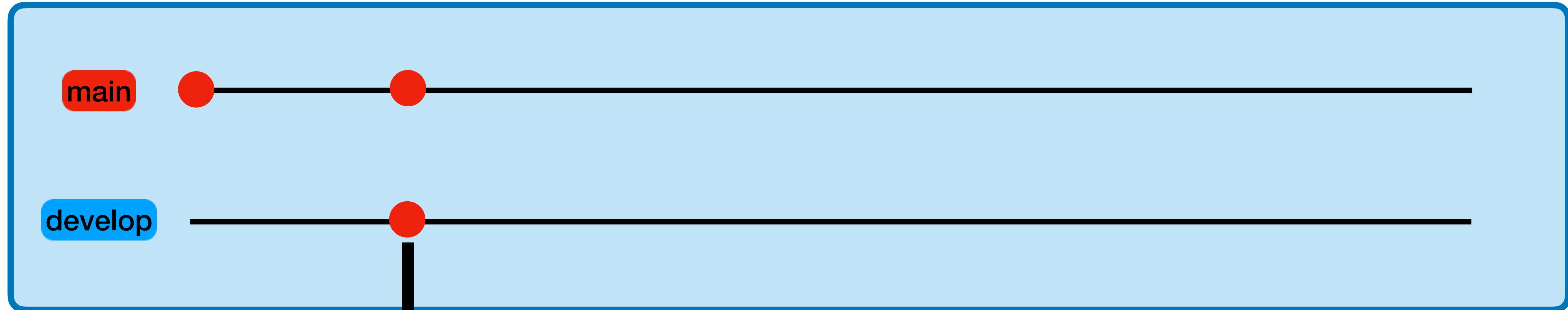


git fetch ➔ main develop  
git checkout develop  
git pull

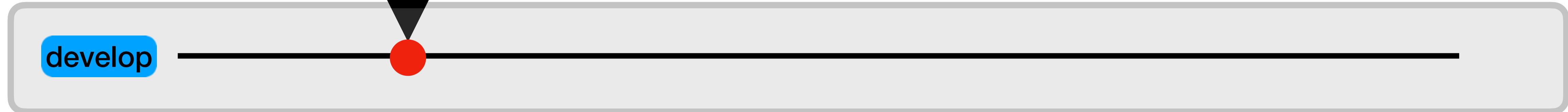


## Branches & GitFlow

### Remote

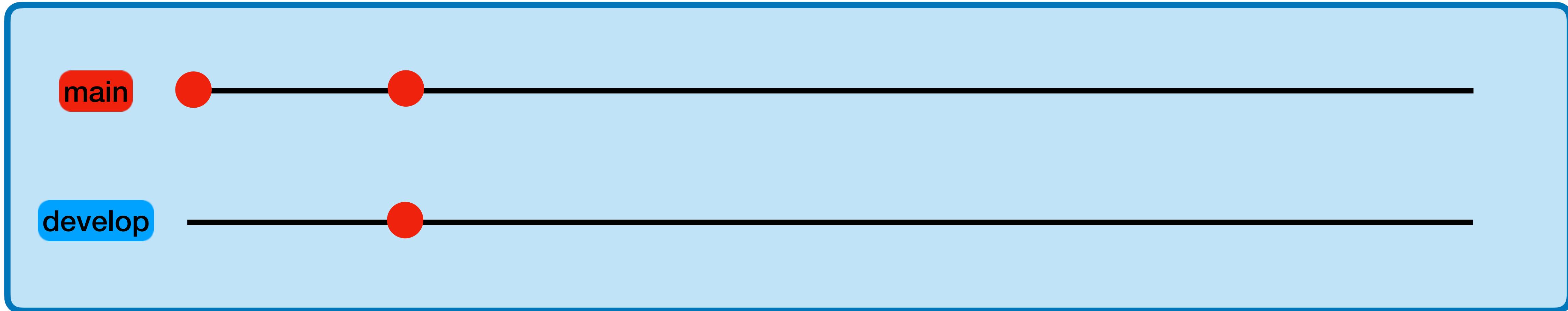


git fetch ➡ main develop  
git checkout develop  
git pull

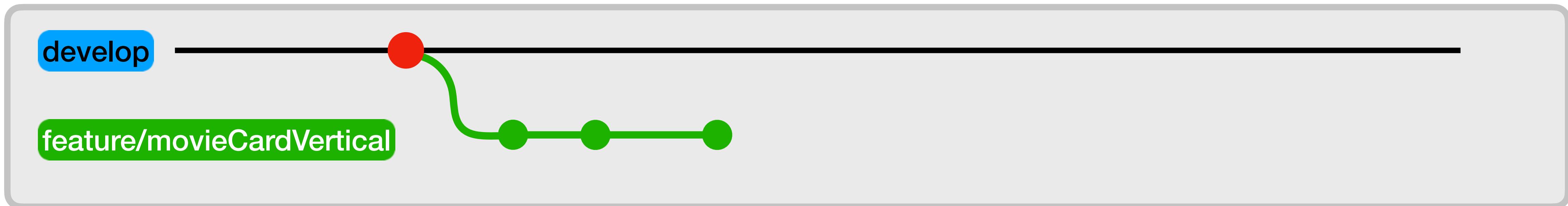


## Branches & GitFlow

### Remote

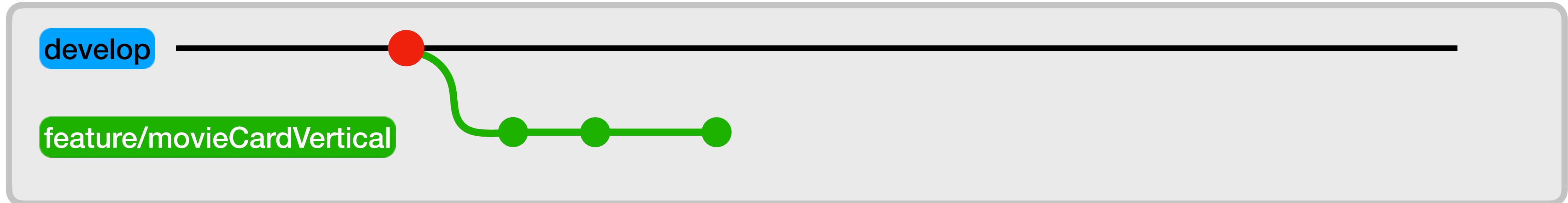
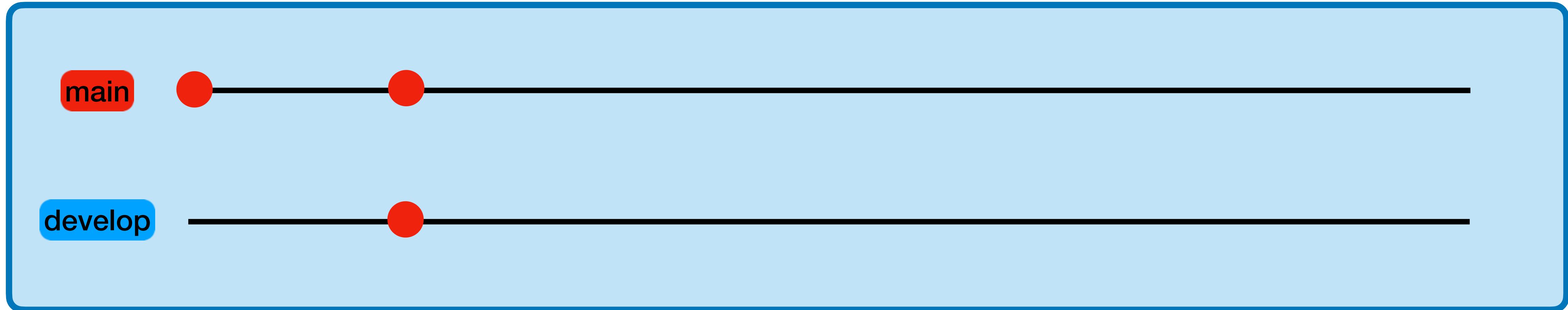


`git checkout -b feature/movieCardVertical`



## Branches & GitFlow

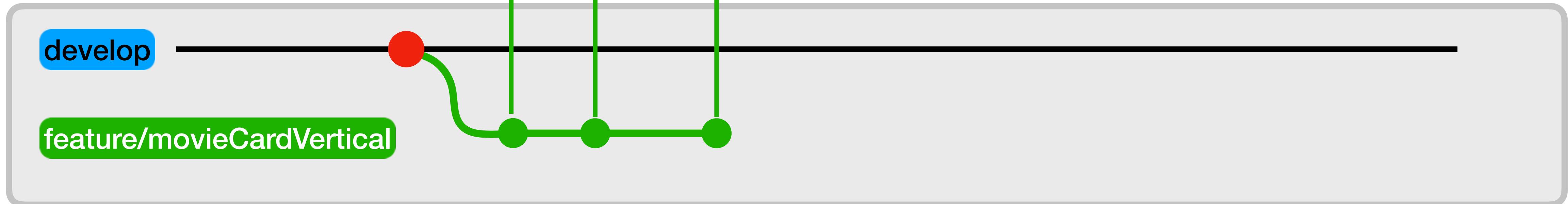
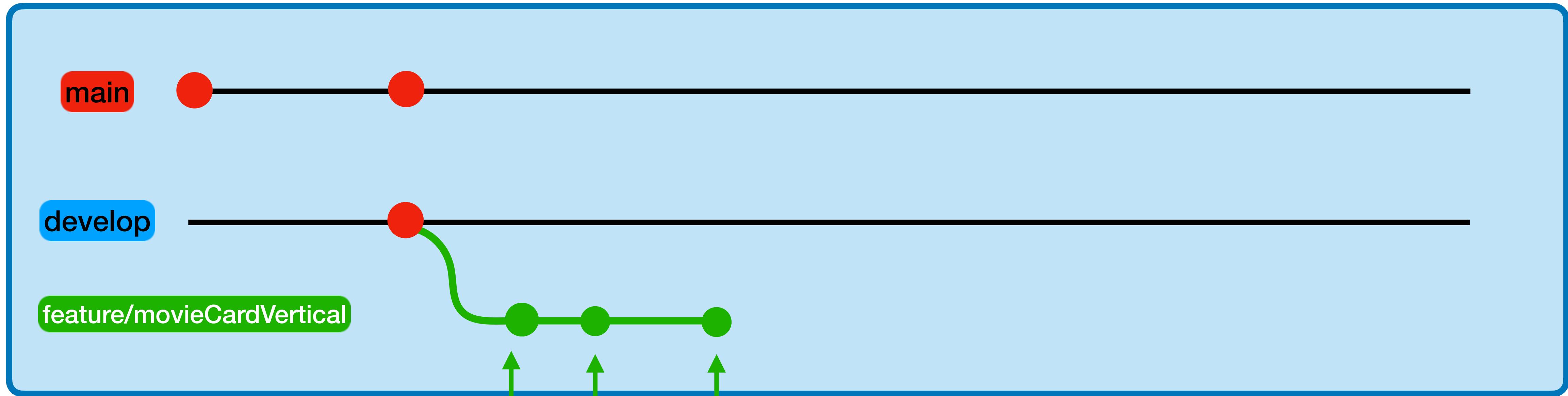
### Remote



git push -u origin feature/movieCardVertical

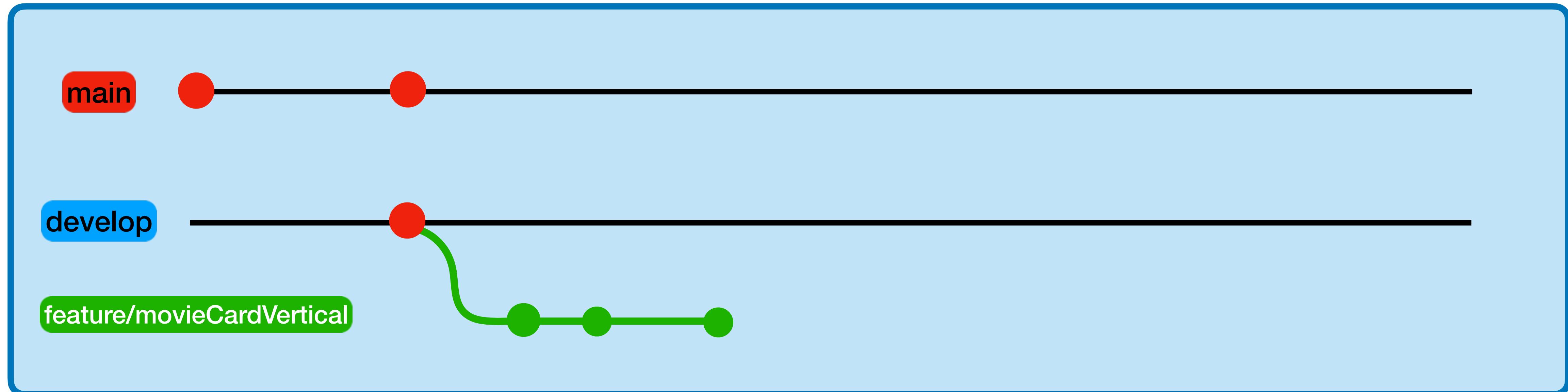
## Branches & GitFlow

### Remote



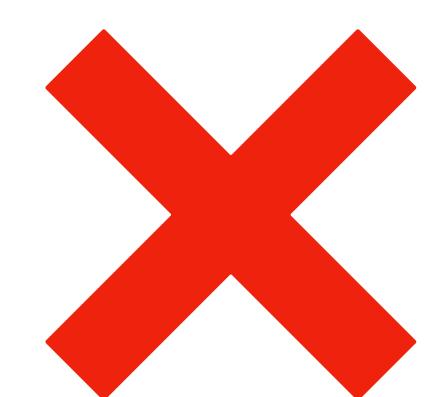
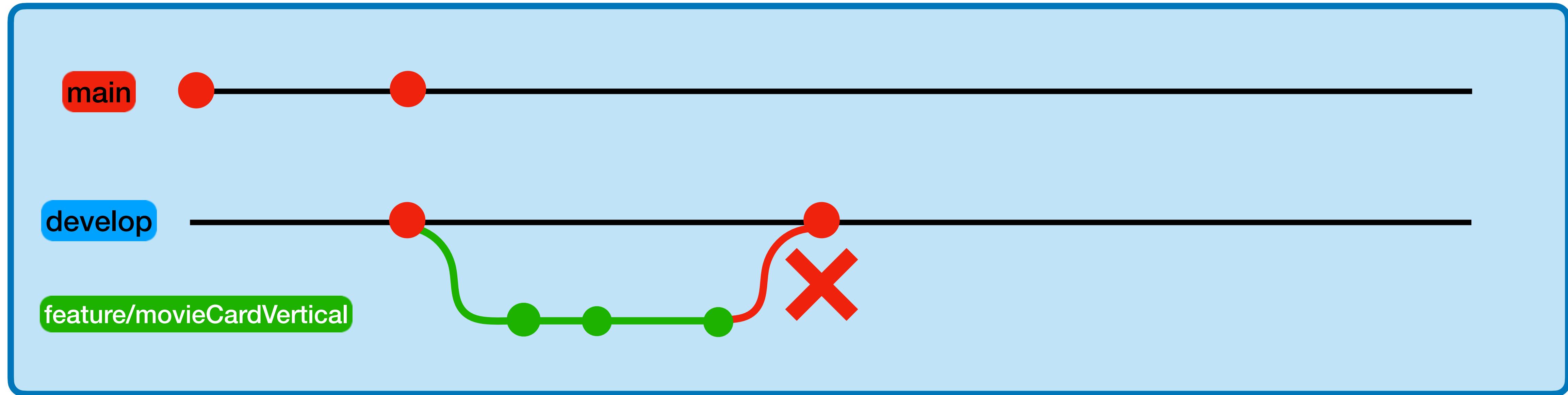
git push -u origin feature/movieCardVertical

## Remote



## Branches & GitFlow

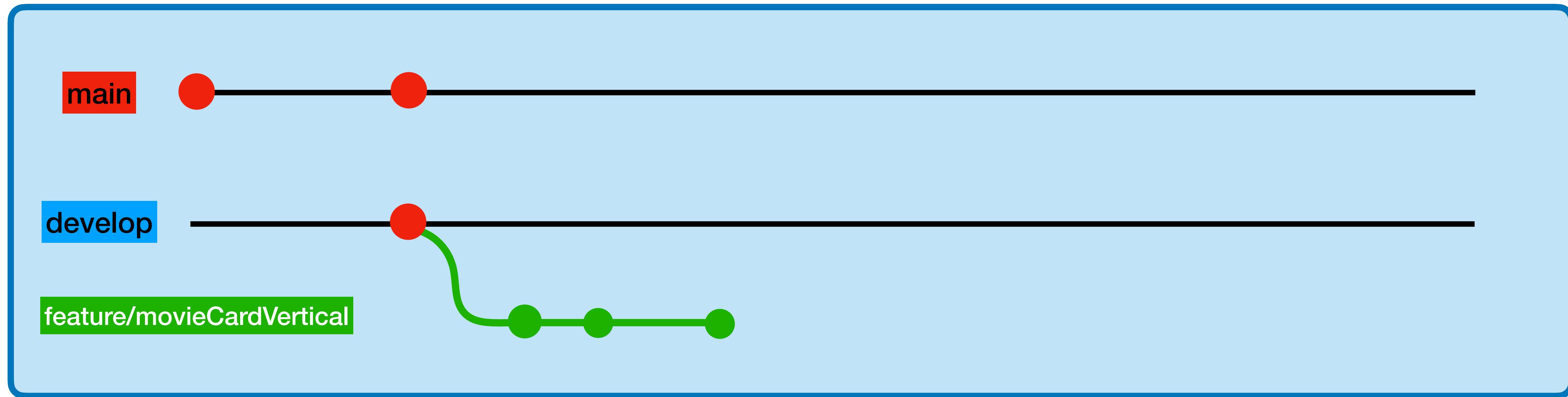
### Remote



git merge develop

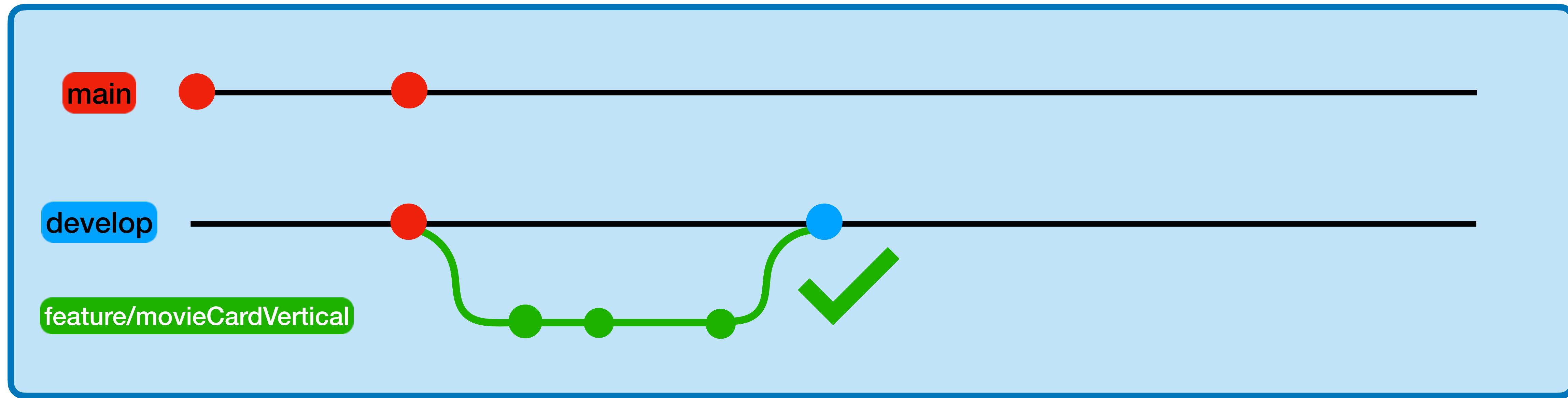


## Remote



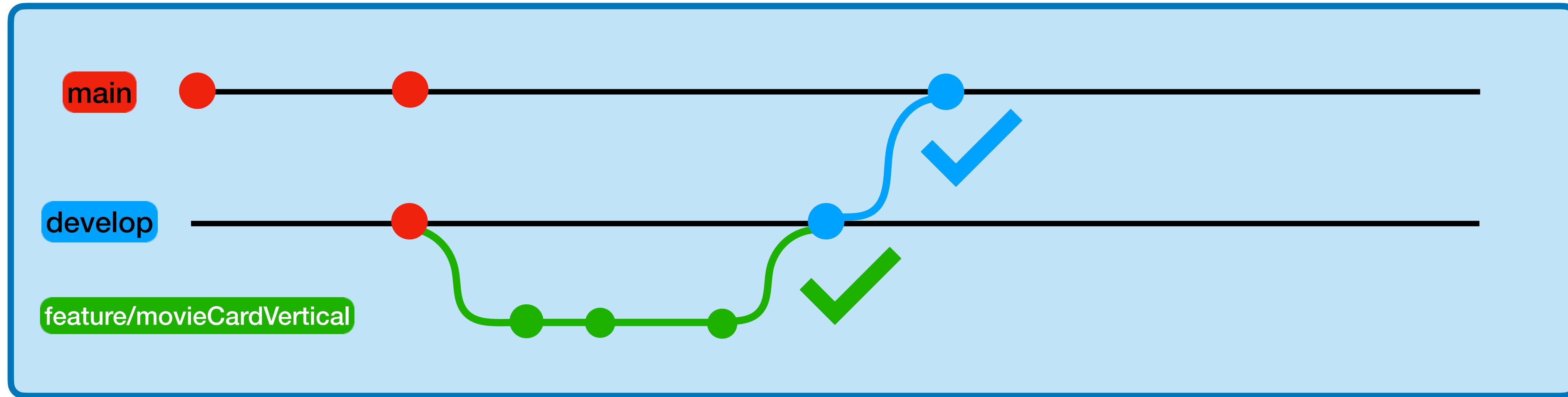
Um **Pull Request** deve ser criado (no GitHub, por exemplo), para que os outros colaboradores do projeto possam **avaliar o código** e autorizar ou não integrá-lo no código da branch solicitada (**develop**)

### Remote



Uma vez que a solicitação foi avaliada e **aprovada** por um certo número de colaboradores, o processo de merge é liberado para ser executado (no GitHub)

## Remote



O mesmo processo (PR) acontece  
da branch **develop** para a **main**



**Fundamentos de Git**



**GitHub**



**Branches & GitFlow**



# Summary of git commands

git --version

git config --global user.name "Your name"

git config --global user.email "Your e-mail"

git config --list

git init

git status

git add file

git commit -m "descriptive message"

git rm --cached file

git log

git branch newBranch

git checkout newBranch

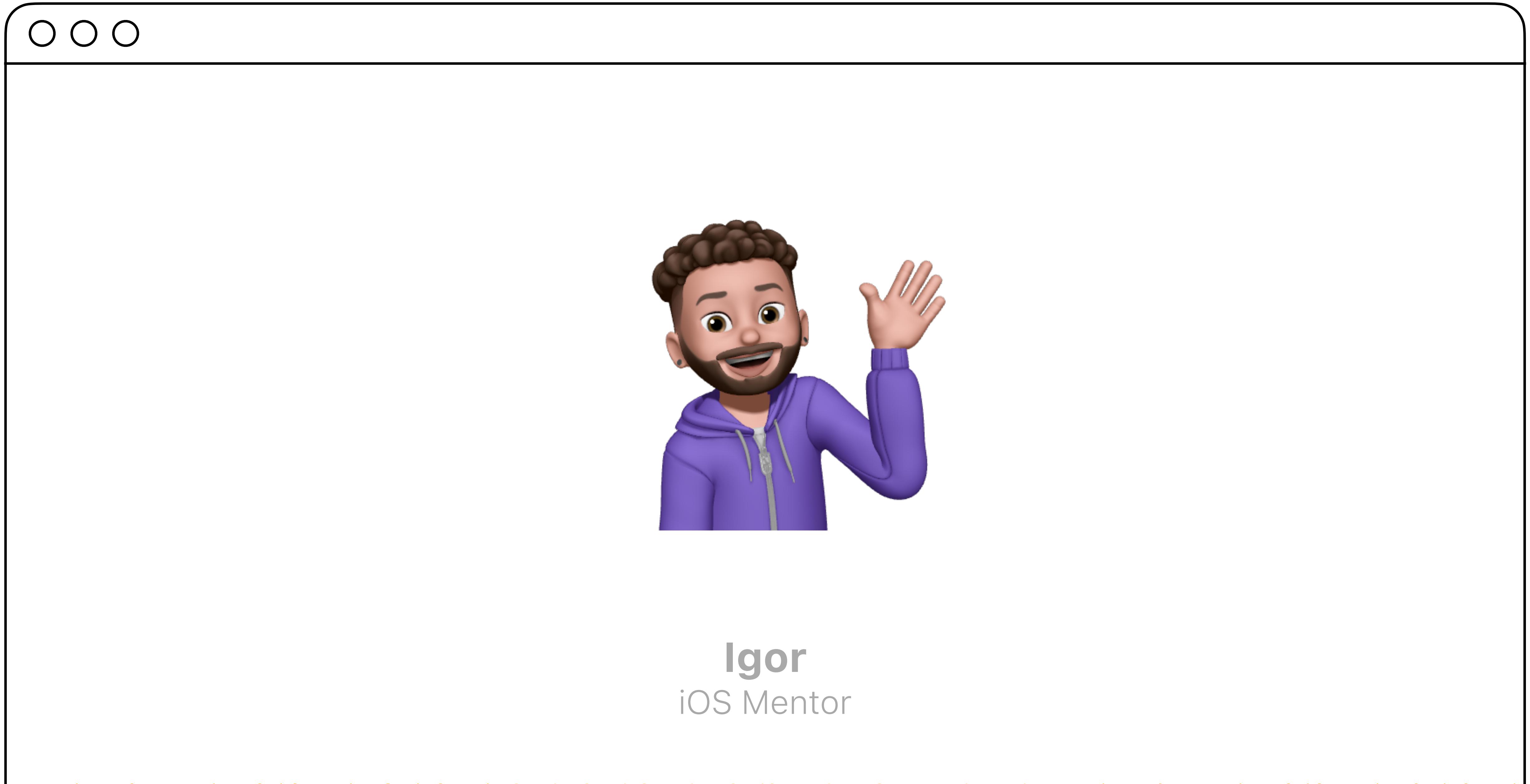
git checkout -b newBranch

git merge branchName

git fetch

git pull

git push



Igor  
iOS Mentor