

08-04-2022

ACADEMY JAVA



Agenda



Agenda

- Piattaforme GIT
- GitHub & GitHub Desktop
- Git tools
- Git Flow
- **Suddivisione in Team**
- **Esercitazione con GIT**



Piattaforme GIT



ACADEMY
JAVA

GitLab

The screenshot shows the GitLab Community Edition repository page. At the top, there's a navigation bar with links for Projects, Groups, Snippets, Help, This project, Search, and Sign in / Register. The main header features the GitLab logo and the text "GitLab Community Edition". Below the header, there's a brief description of what GitLab Community Edition is: "GitLab Community Edition (CE) is an open source end-to-end software development platform with built-in version control, issue tracking, code review, CI/CD, and more. Self-host GitLab CE on your own servers, in a container, or on a cloud provider." There are several status indicators at the top: pipeline (passed), coverage (75.47%), ai best practices (passing), maintainability (B), and chat (on gitter). Below these are standard GitLab interface elements like a star icon, a commit count (4786), HTTPS, and a copy link button. A search bar allows filtering by branch (master) and repository name (gitlab-ce). A recent commit is highlighted: "Merge branch '5594-geo-add-repository-verification-failures-to-api' into 'master'" by Douglas Barbosa Alexandre, authored 5 hours ago, with a commit hash b60364c0. Below this, a table lists repository files and their last commits:

Name	Last commit	Last update
.github	Address feedback about wording.	2 years ago
.gitlab	Merge Templates updates	2 days ago
app	Merge branch 'frozen-string-enable-apps-servi...	9 hours ago
bin	Truncate filenames created by bin/changelog to ...	1 week ago

BitBucket

 BitBucket sidebar icons include: Source (highlighted), Commits, Branches, Pull requests, Pipelines, Deployments, Downloads, Boards, and Settings.

teams-in-space

Stephen Hale

teams-in-space

Here's where you'll find this repository's source files. To give your users an idea of what they'll find here, [add a description to your repository](#).

Clone 

master 

Name	Size	Last commit	Message
images		2018-12-20	Initial commit
.gitignore	42 B	2018-12-20	Initial commit
README.md	3.66 KB	2019-02-06	README.md edited online with Bitbucket
bitbucket-pipelines.yml	575 B	2018-12-20	Initial commit
index.js	865 B	2018-12-20	Initial commit
index.test.js	977 B	2018-12-20	Initial commit
package.json	322 B	2018-12-20	Initial commit

README.md

Pipelines Node.js

Pipelines allows you to put your Bitbucket hosted code to work. It enables you to build, test, and deploy your code using the cloud and the principals of CI/CD. You might like to run tests triggered by any git push to Bitbucket, to confirm that your commit did not introduce any new problems. Or, you could deploy a new version of your code, automatically, whenever your tests complete successfully; turning on features at your leisure using feature flags. Let's get started!

This is an example repo showing Bitbucket Pipelines in a Node.js environment.

Repository details

Last updated 2 hours ago

Open pull requests 1	Branches 5
Watchers 1	Forks 0
Version control system Git	Language Node.js
Access level Admin	
Fork of bitbucketpipelines/pipelines-guid...	

1 of 1 build passed

 Pipeline #5 for master · 2019-02-06

Give feedback

Azure DevOPS

Contoso / AdventureWorks Mobile / Test Plans / Web Team

[Create New](#)

Test Plan: Web Team

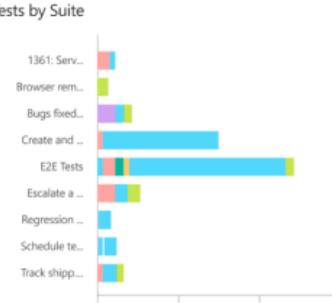
[Overview](#) [Boards](#) [Repos](#) [Pipelines](#) [Test Plans](#) [Test plans](#) [Parameters](#) [Configurations](#) [Runs](#) [Artifacts](#)

Overall Execution State



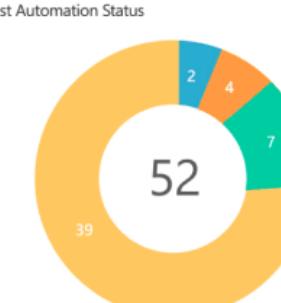
Status	Count
Not run	80
Failed	11
Passed	10
Blocked	6
In progress	1
Not applicable	1

Tests by Suite



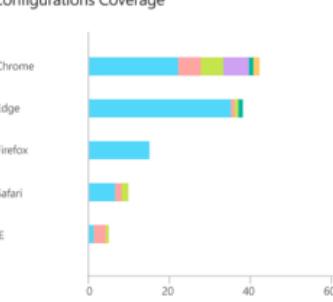
Suite	Not run	Failed	Passed	Blocked	In progress
1361: Serv...	1	1	0	0	0
Browser rem...	1	0	0	0	0
Bugs fixed...	2	0	0	0	0
Create and ...	30	0	0	0	0
E2E Tests	1	1	0	0	0
Escalate a ...	2	0	0	0	0
Regression ...	1	0	0	0	0
Schedule te...	1	0	0	0	0
Track shipp...	1	0	0	0	0

Test Automation Status



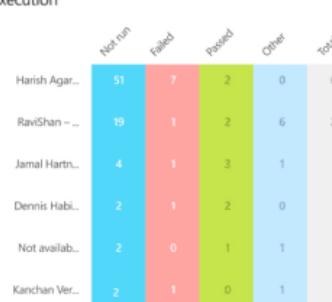
Status	Count
Idea	2
Development	4
Testing	7
Ready	39

Configurations Coverage



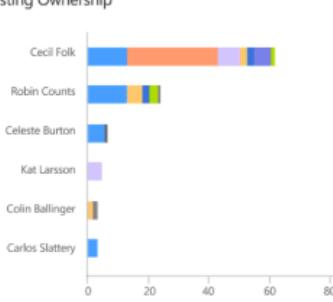
Configuration	Not run	Failed	Passed	Blocked
Chrome	35	5	10	0
Edge	38	0	1	0
Firefox	15	0	0	0
Safari	10	2	0	0
IE	5	1	0	0

Execution



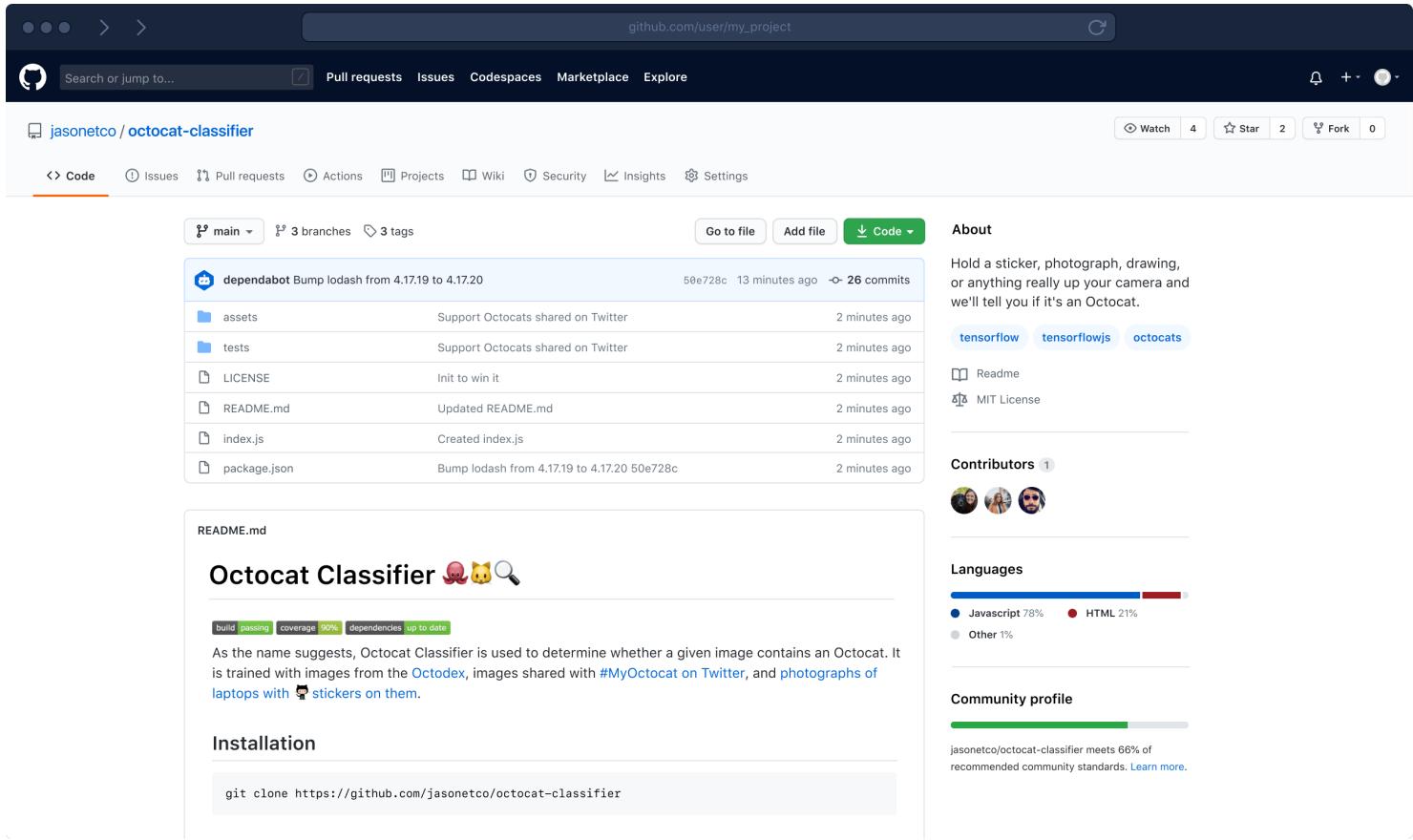
Team Member	Not run	Failed	Passed	Other	Total
Harish Agar...	51	7	3	0	60
RaviShan ...	19	1	2	6	28
Jamal Hartn...	4	1	3	1	9
Dennis Habi...	2	1	2	0	5
Not availab...	2	0	1	1	4
Kanchan Ver...	2	1	0	1	4
Total	51	51	51	51	51

Testing Ownership



Team Member	E2E Tests	Create and ...	Escalate a ...	Track shipp ...
Cecil Folk	10	15	5	2
Robin Counts	10	5	1	1
Celeste Burton	5	0	0	0
Kat Larsson	0	0	5	0
Colin Ballinger	0	1	0	0
Carlos Slattery	0	0	0	5

GitHub



The screenshot shows a GitHub repository page for `jasonetco/octocat-classifier`. The repository has 4 watches, 2 stars, and 0 forks. The code tab is selected, showing a recent commit from dependabot bumping lodash from 4.17.19 to 4.17.20. The repository contains assets, tests, LICENSE, README.md, index.js, and package.json files. The README.md file describes the Octocat Classifier, which uses images from the Octodex and photographs of laptops with stickers to identify Octocats. The installation section provides a command to clone the repository. The repository's community profile shows it meets 66% of recommended community standards.

github.com/user/my_project

jasonetco / octocat-classifier

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 3 branches 3 tags

dependabot Bump lodash from 4.17.19 to 4.17.20 50e728c 13 minutes ago ~ 26 commits

assets Support Octocats shared on Twitter 2 minutes ago

tests Support Octocats shared on Twitter 2 minutes ago

LICENSE Init to win it 2 minutes ago

README.md Updated README.md 2 minutes ago

index.js Created index.js 2 minutes ago

package.json Bump lodash from 4.17.19 to 4.17.20 50e728c 2 minutes ago

Go to file Add file Code

About

Hold a sticker, photograph, drawing, or anything really up your camera and we'll tell you if it's an Octocat.

tensorflow tensorflowjs octocats

Readme MIT License

Contributors 1

Javascript 78% HTML 21% Other 1%

Community profile

jasonetco/octocat-classifier meets 66% of recommended community standards. Learn more.

Octocat Classifier 🐕🔍

build passing coverage 90% dependencies up to date

As the name suggests, Octocat Classifier is used to determine whether a given image contains an Octocat. It is trained with images from the [Octodex](#), images shared with [#MyOctocat on Twitter](#), and [photographs of laptops with stickers on them](#).

Installation

```
git clone https://github.com/jasonetco/octocat-classifier
```

GitHub & GitHub Desktop



Cos'è GitHub

Il sito è principalmente utilizzato dagli sviluppatori, che caricano il codice sorgente dei loro programmi e lo rendono scaricabile dagli utenti. Questi ultimi possono interagire con lo sviluppatore tramite un sistema di issue tracking, pull request e commenti che permette di migliorare il codice del repository risolvendo bug o aggiungendo funzionalità. Inoltre Github elabora dettagliate pagine che riassumono come gli sviluppatori lavorano sulle varie versioni dei repository.



Cos'è GitHub Desktop

GitHub Desktop è un'app per utenti Mac e PC che porta il controllo della versione dalla riga di comando al desktop. La riga di comando, sebbene potente, ha una curva di apprendimento più ripida. GitHub ha creato questo programma per semplificare il controllo della versione.



A cosa serve GitHub Desktop

GitHub Desktop può eseguire le stesse attività utilizzate per controllare i repository, tra cui:

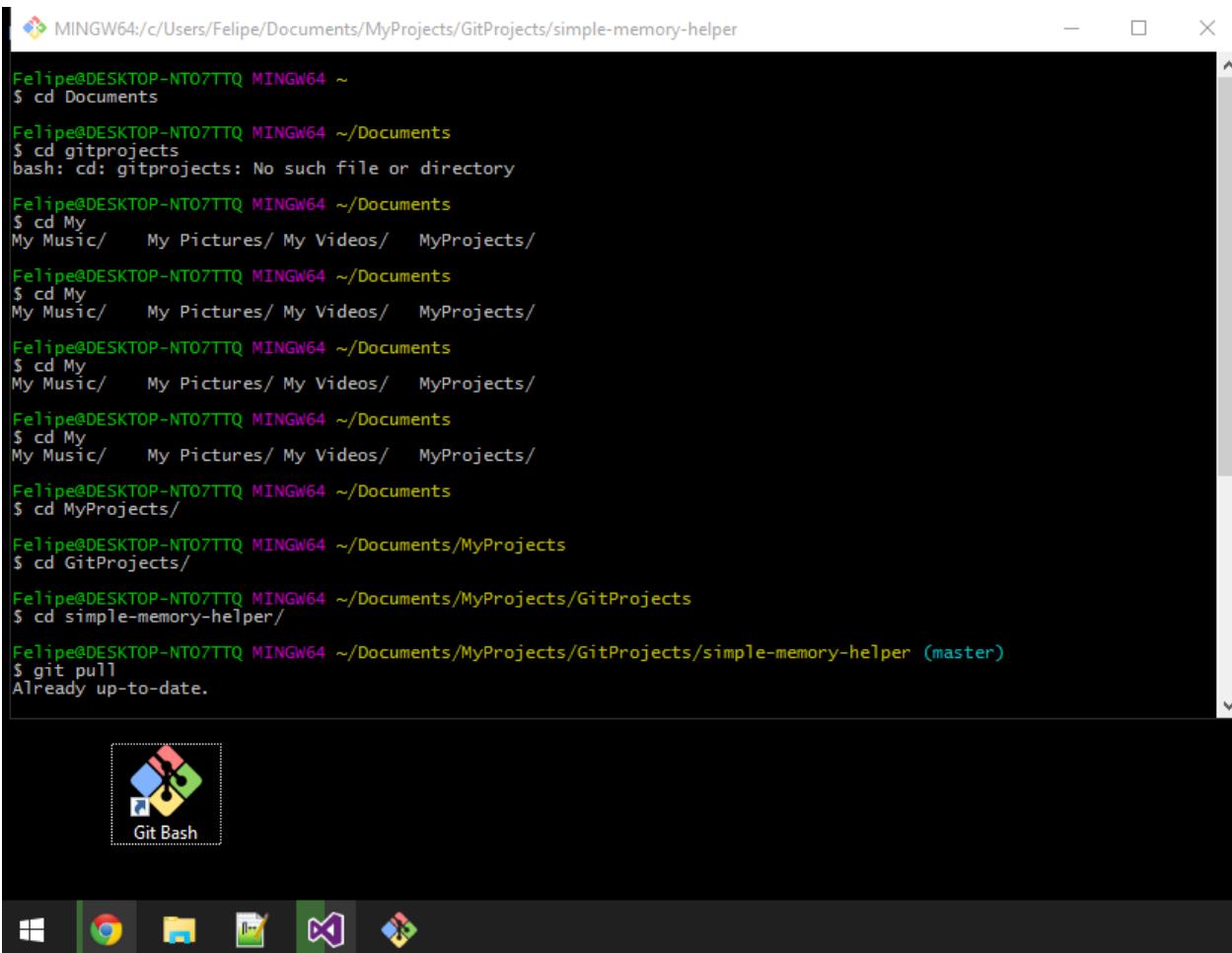
- Clonazione dei repository
- Eliminazione di repository
- Aggiornamento e salvataggio dei repository



Git Tools



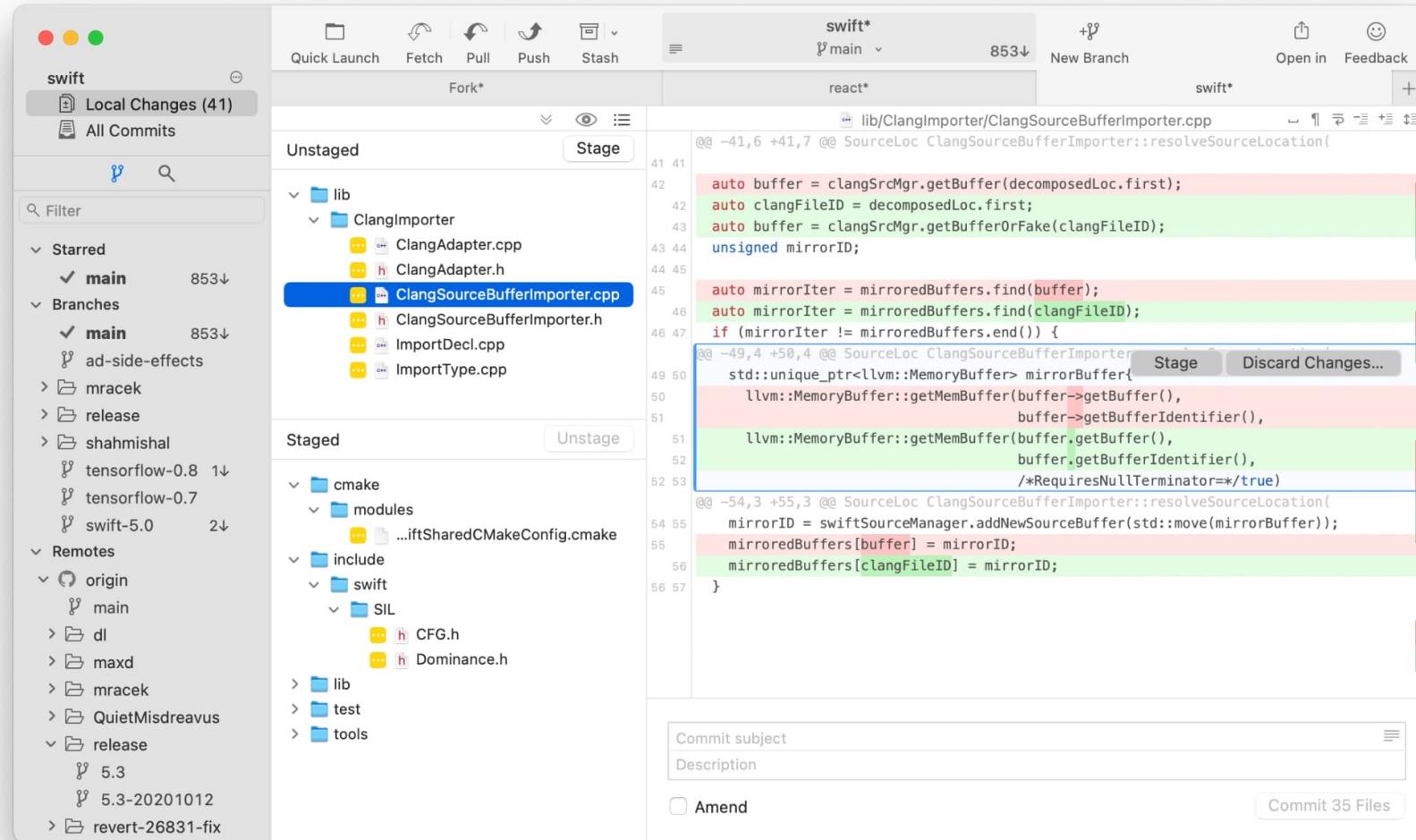
Git Bash



```
Felipe@DESKTOP-NT07TTQ MINGW64 ~
$ cd Documents
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents
$ cd gitprojects
bash: cd: gitprojects: No such file or directory
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents
$ cd My
My Music/ My Pictures/ My Videos/ MyProjects/
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents
$ cd My
My Music/ My Pictures/ My Videos/ MyProjects/
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents
$ cd My
My Music/ My Pictures/ My Videos/ MyProjects/
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents
$ cd My
My Music/ My Pictures/ My Videos/ MyProjects/
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents/MyProjects
$ cd GitProjects/
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents/MyProjects/GitProjects
$ cd simple-memory-helper
Felipe@DESKTOP-NT07TTQ MINGW64 ~/Documents/MyProjects/GitProjects/simple-memory-helper (master)
$ git pull
Already up-to-date.
```



Fork



The screenshot shows a GitHub fork interface for the 'swift' repository. The left sidebar lists local changes (41), starred branches (main), and remotes (origin). The main area displays the 'Fork*' branch with the commit 'react*' (853↓). The 'Unstaged' section shows files in the 'lib/ClangImporter' directory, with 'ClangSourceBufferImporter.cpp' selected. The 'Staged' section shows files in 'cmake/modules', 'include/swift/SIL', and other directories. A diff view at the bottom shows changes in 'ClangSourceBufferImporter.cpp'. The commit subject and description fields are empty, and there is an 'Amend' checkbox.

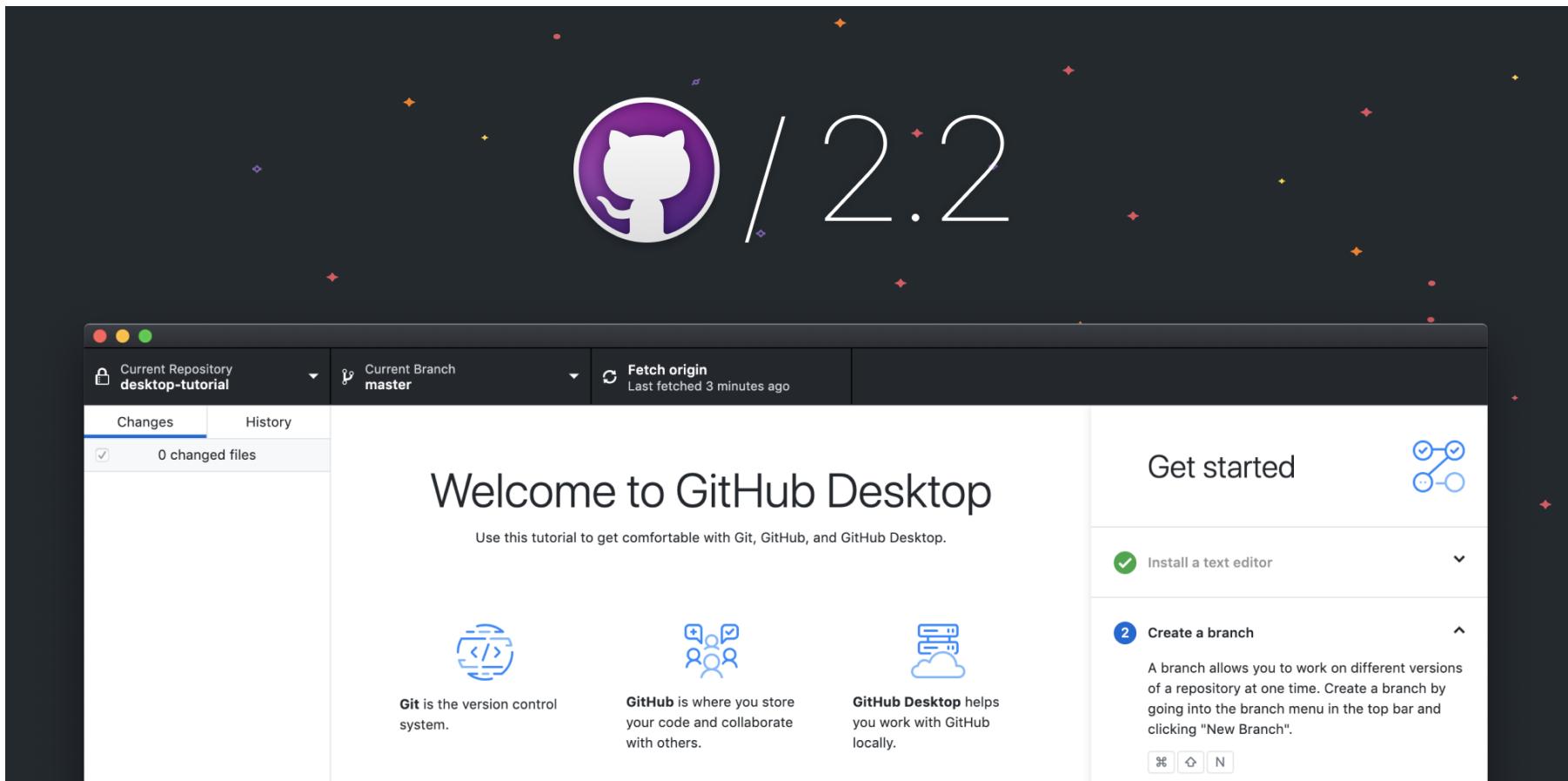
```

diff --git a/lib/ClangImporter/ClangSourceBufferImporter.cpp b/lib/ClangImporter/ClangSourceBufferImporter.cpp
@@ -41,6 +41,7 @@ @ SourceLoc ClangSourceBufferImporter::resolveSourceLocation(
auto buffer = clangSrcMgr.getBuffer(decomposedLoc.first);
auto clangFileID = decomposedLoc.first;
auto buffer = clangSrcMgr.getBufferOrFake(clangFileID);
unsigned mirrorID;

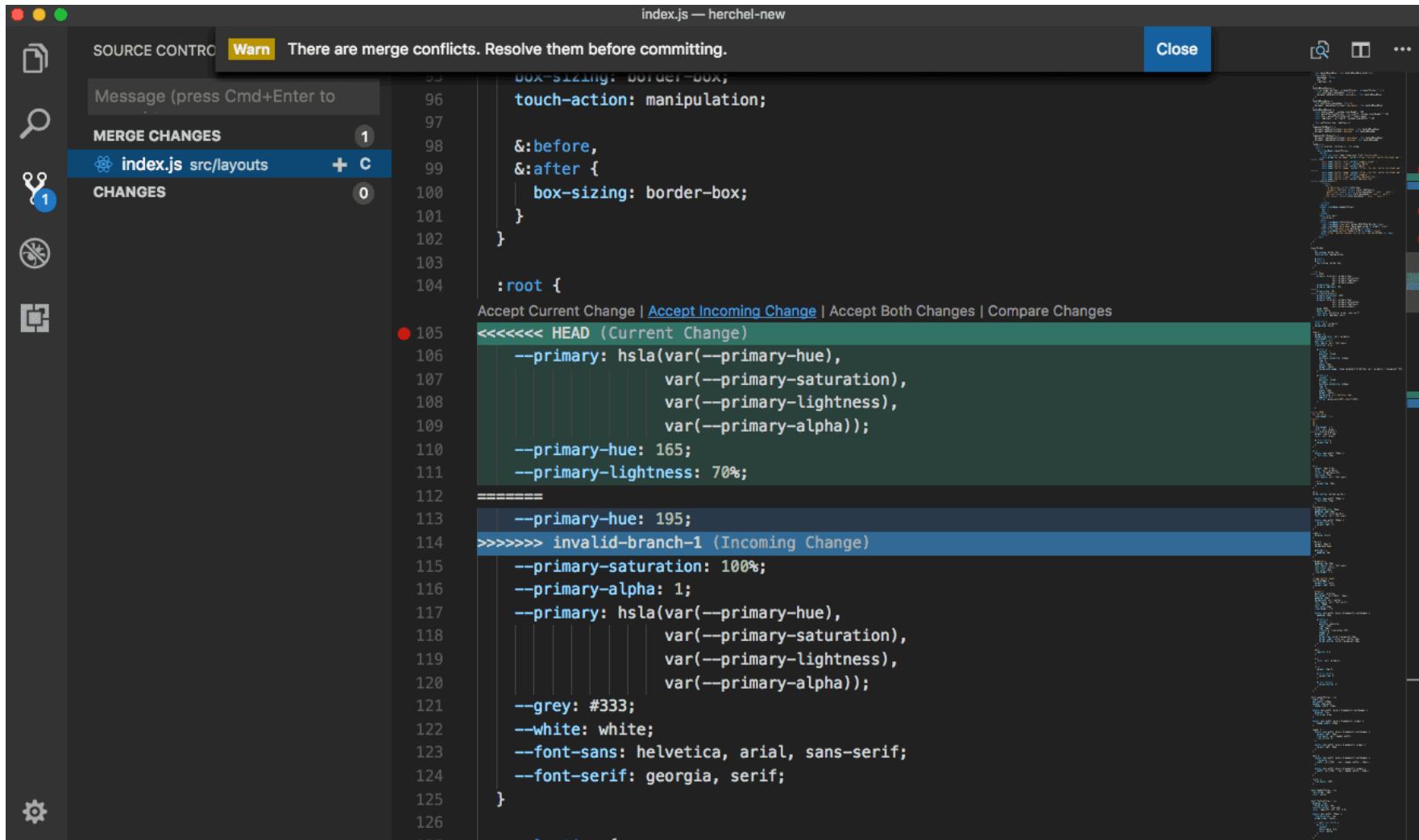
+auto mirrorIter = mirroredBuffers.find(buffer);
+auto mirrorIter = mirroredBuffers.find(clangFileID);
+if (mirrorIter != mirroredBuffers.end()) {
@@ -49,4 +50,4 @@ @ SourceLoc ClangSourceBufferImporter
std::unique_ptr<llvm::MemoryBuffer> mirrorBuffer;
 llvm::MemoryBuffer::getMemBuffer(buffer->getBuffer(),
buffer->getBufferIdentifier(),
@@ -51,2 +52,3 @@ @ SourceLoc ClangSourceBufferImporter
 llvm::MemoryBuffer::getMemBuffer(buffer.getBuffer(),
buffer.getBufferIdentifier(),
+/*RequiresNullTerminator=*/true)
@@ -54,3 +55,3 @@ @ SourceLoc ClangSourceBufferImporter::resolveSourceLocation(
mirrorID = swiftSourceManager.addNewSourceBuffer(std::move(mirrorBuffer));
mirroredBuffers[buffer] = mirrorID;
mirroredBuffers[clangFileID] = mirrorID;
}

```

GitHub Desktop



VS Code



The screenshot shows a merge conflict in the VS Code editor for the file `index.js`. A warning message at the top states: "There are merge conflicts. Resolve them before committing." The code editor displays two diverging branches of the same code block. The left side shows the current HEAD (Current Change) with the following CSS snippet:

```
95    box-sizing: border-box;
96    touch-action: manipulation;
97
98    &:before,
99    &:after {
100      box-sizing: border-box;
101    }
102
103
104 :root {
105   Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
106   <<<<< HEAD (Current Change)
107   --primary: hsla(var(--primary-hue),
108             var(--primary-saturation),
109             var(--primary-lightness),
110             var(--primary-alpha));
111   --primary-hue: 165;
112   --primary-lightness: 70%;
113
114   =====
115   --primary-hue: 195;
116   >>>> invalid-branch-1 (Incoming Change)
117   --primary-saturation: 100%;
118   --primary-alpha: 1;
119   --primary: hsla(var(--primary-hue),
120                 var(--primary-saturation),
121                 var(--primary-lightness),
122                 var(--primary-alpha));
123   --grey: #333;
124   --white: white;
125   --font-sans: helvetica, arial, sans-serif;
126   --font-serif: georgia, serif;
127 }
```

Git Flow



Introduzione a Git Flow

Git Flow è uno degli approcci utilizzati nella gestione delle repository GIT. Consente di gestire in maniera ordinata e funzionale una repository utilizzata da un team di sviluppo anche di grandi dimensioni.

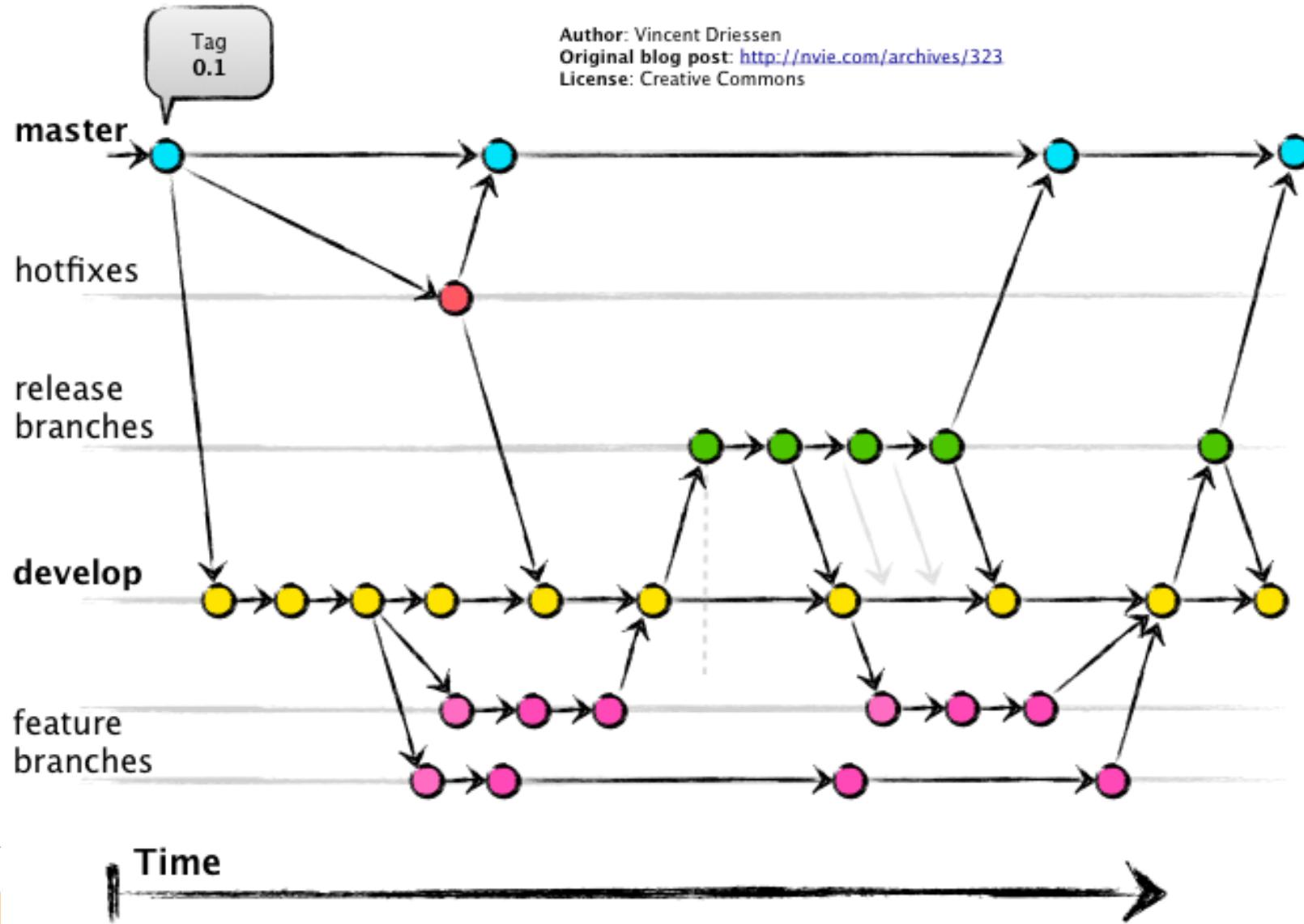
È basata su principi:

- Un ramo **develop** deve essere creato a partire da quello **master**
- Un ramo **release** deve essere creato a partire da **develop**
- Un ramo **feature** deve essere creato a partire da **develop**
- Una volta finito un ramo **feature**, lo si unisce in **develop**
- Una volta pronto il ramo **release**, per pubblicarlo lo si unisce in **develop e master**
- Un ramo **hotfix** viene creato a partire da **master**
- Una volta pronto il ramo **hotfix**, viene unito in **develop e master**





Author: Vincent Driessen
Original blog post: <http://nvie.com/archives/323>
License: Creative Commons



Practice



ACADEMY
JAVA

Esercizio

1. Crea un nuovo progetto su GitHub
2. Clona il progetto su una tua cartella locale
3. Utilizza **git status** per controllare su che branch sei
4. Crea un nuovo file nella cartella
5. Utilizza **git status** per controllare cosa succede
6. Aggiungi il file nell'area di staging
7. Utilizza di nuovo **git status** per controllare cosa succede
8. Effettua una commit del codice
9. Utilizza di nuovo **git status** per controllare cosa succede
10. Cambia il contenuto del file che hai creato
11. Utilizza di nuovo **git status** per controllare cosa succede
12. Aggiungi il file modificato nell'area di staging
13. Utilizza di nuovo **git status** per controllare cosa succede
14. Modifica di nuovo il file
15. Effettua una commit
- 16. Cosa succede in questo caso?**
17. Aggiungi ed effettua una commit delle ultime modifiche

GRAZIE

ACADEMY
JAVA

