

# WIZELINE

## Hands On Git + Github

## Meet the team



Software Engineer

### Javier Carmona Gallegos

- 30 years old.
- From Cortazar Guanajuato
- Computer Science Engineer from ITC
- 8 years of experience
- Software Engineer at Wizeline



## Where did I collaborated?



The Washington Post



WIZELINE



UNIVERSIDAD DE  
GUANAJUATO



ROSETTA



# What to expect of this workshop



- Know What is Git
- Know What is Github
- Describe the difference between Git and Github
- Know the very basic Git commands
- Know to setup a Git repository\*
- Know to upload a Git repository to Github\*
- Know the basics about collaborating

# What is Git?



git

# **For who is Git?**

**A: For anyone who  
code**

# **What happens if I decide not to use git?**

# **When I should start using Git?**

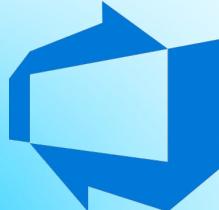
# What is Github?



# Alternatives to Github



GitLab



Azure  
DevOps



Gitea

# **What is a good practice to track?**

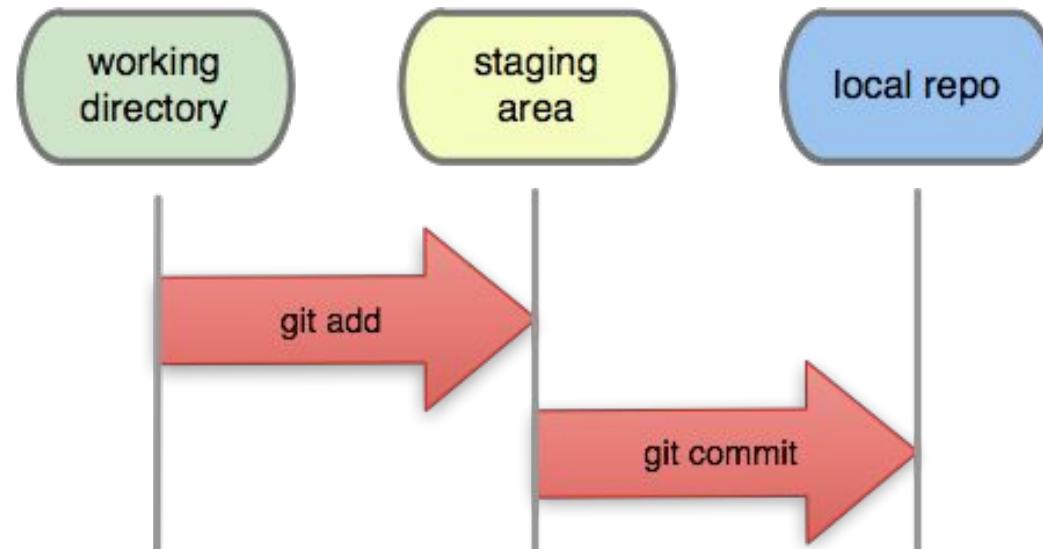
- **Code Java, C, C++, HTML, CSS, JS, C#, Swift, PHP, Assembly**
- **Basic Assets like Icons, Custom Icons, Some Images\***

# What IS NOT a GOOD PRACTICE TO TRACK?

- **Large Files (Like videos, Profile Pics, Audio\*)**
- **.env Files (like secret keys, Database keys, third party apikeys)**
- **Large Dependencies (node\_modules)**
- **Ofimatic Files, PDF Files**

# How git works?

Local



# Knowing if you have Git

On a terminal execute:

```
git --version
```

# Recipe to setup a Git Repositor y (In your local machine)

On a terminal go to the root of your project and execute:

```
git init
```

Then Execute:

```
git status
```

Then Execute:

```
git config --global core.editor nano
```

# Commit our changes

Execute:

```
git add .
```

Then Execute:

```
git commit -m "First
```

```
Commit"
```

# Check the logs and time travel

Execute:

```
git log
```

Take the hash of the commit and execute:

```
git checkout [commit HASH]
```

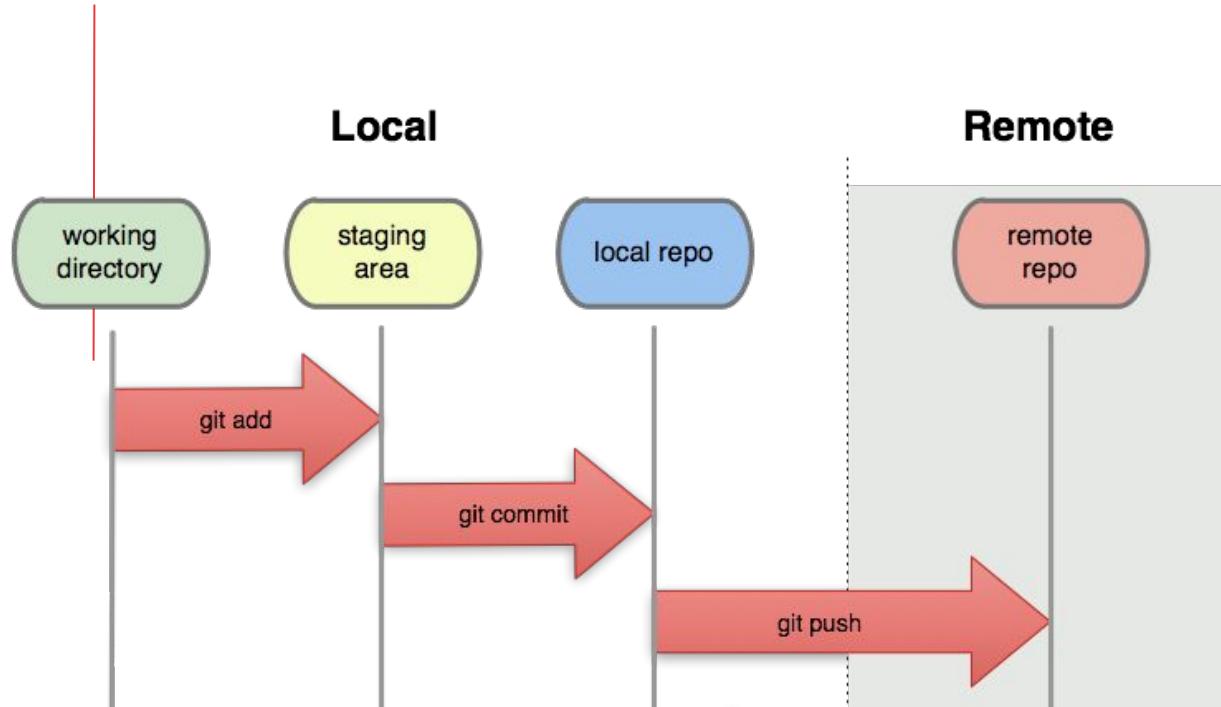
And for return to your work:

```
git checkout [branch]
```



# Demonstration: Setup a Git repository + Basic Commands

# How I upload a local git repo to Github?



# Pushing your changes to remote repo

On a terminal go to the root of your project and execute:

```
git push
```

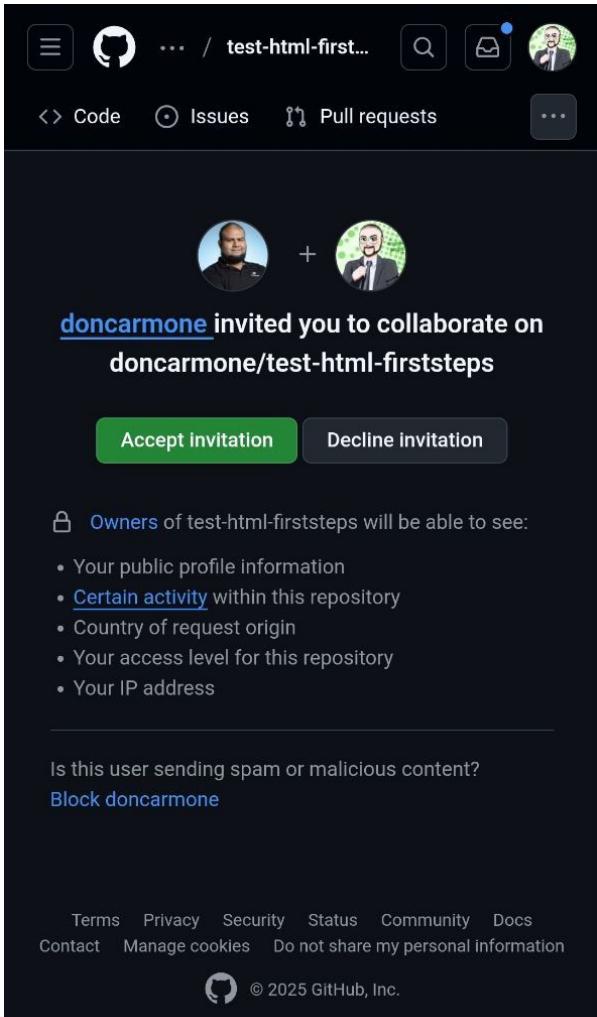
Then Execute:

```
git remote add origin [REPO]
```

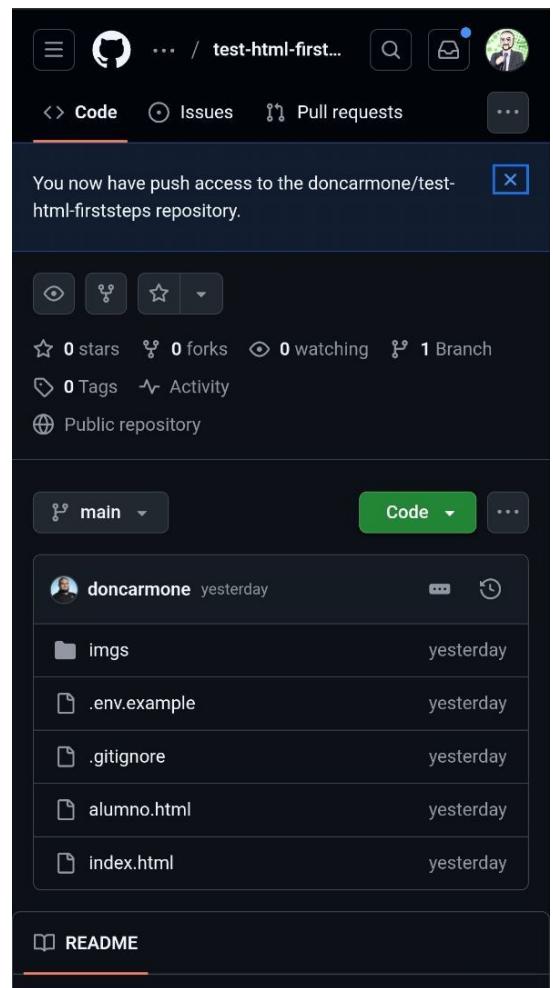
Then Execute:

```
git push -u origin main
```

# Invitation Example



The screenshot shows a GitHub invitation interface. At the top, there are navigation icons for code, issues, pull requests, and more. Below that, two user profile pictures are shown with a plus sign between them, indicating a collaboration invite. The main message reads: "doncarmone invited you to collaborate on doncarmone/test-html-firststeps". Two buttons are present: "Accept invitation" (green) and "Decline invitation" (grey). A note below states: "Owners of test-html-firststeps will be able to see:" followed by a bulleted list: "Your public profile information", "Certain activity within this repository", "Country of request origin", "Your access level for this repository", and "Your IP address". A "Block" link is also provided. At the bottom, there are links for Terms, Privacy, Security, Status, Community, Docs, Contact, Manage cookies, and a "Do not share my personal information" checkbox. The footer includes the GitHub logo and the text "© 2025 GitHub, Inc.".



The screenshot shows a GitHub repository page for "doncarmone/test-html-firststeps". The top bar has "Code" selected. A message says: "You now have push access to the doncarmone/test-html-firststeps repository." Below the message are standard repository stats: 0 stars, 0 forks, 0 watching, 1 Branch, 0 Tags, and Activity. It's listed as a Public repository. The main content area shows a list of files in the "main" branch: "img" (yesterday), ".env.example" (yesterday), ".gitignore" (yesterday), "alumno.html" (yesterday), and "index.html" (yesterday). A "README" file is also visible at the bottom. The GitHub logo and "© 2025 GitHub, Inc." are at the very bottom.



## Part II: Collaborating

# How to clone a repo in my local machine

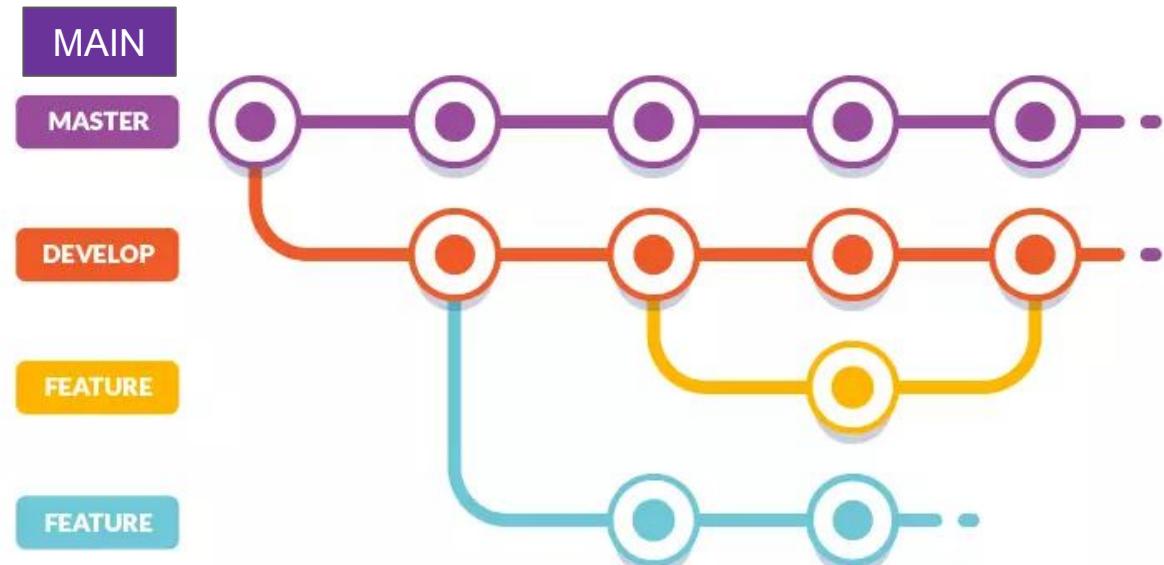
On a terminal go to the root of your project and execute:

```
git clone [REPO]
```

Then Execute:

```
git status
```

# What is a branch in Git?



# Basic Branch Command

S

Print the branch where you are:

```
git branch
```

Create new branch:

```
git checkout -b [name]
```

Rename branch name:

```
git branch -m [new name]
```



# Demonstration of a real world example collaborating with github.

# Basic troubleshootin g

Reset all the changes on your file (Not on Stage):

```
git checkout -- [The path to  
your file]
```

Unstage a file:

```
git restore --staged  
[Path to the file]
```

# Basic troubleshootin g

Change the message from a commit:

```
git commit --amend
```

Reset all the changes on the unstaged branch:

```
git reset --hard
```



# Basic Conflict resolution

Reset all the changes on your file (Not on Stage):

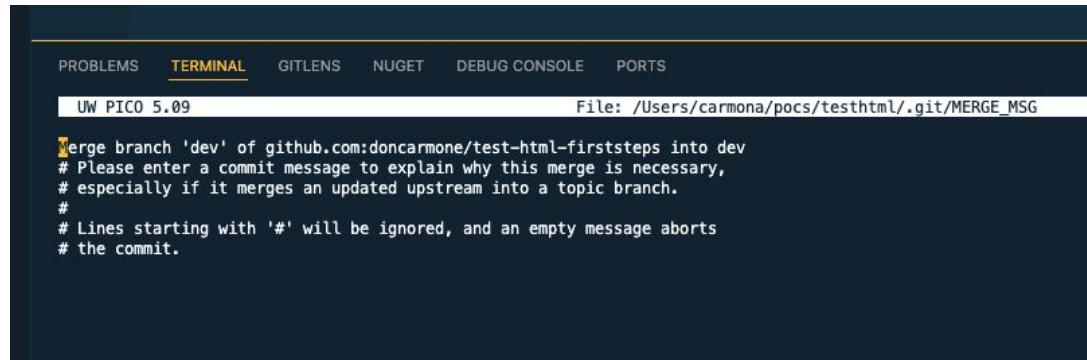
```
git pull --no-ff
```

And resolve the issues and execute a normal commit

```
git add .
```

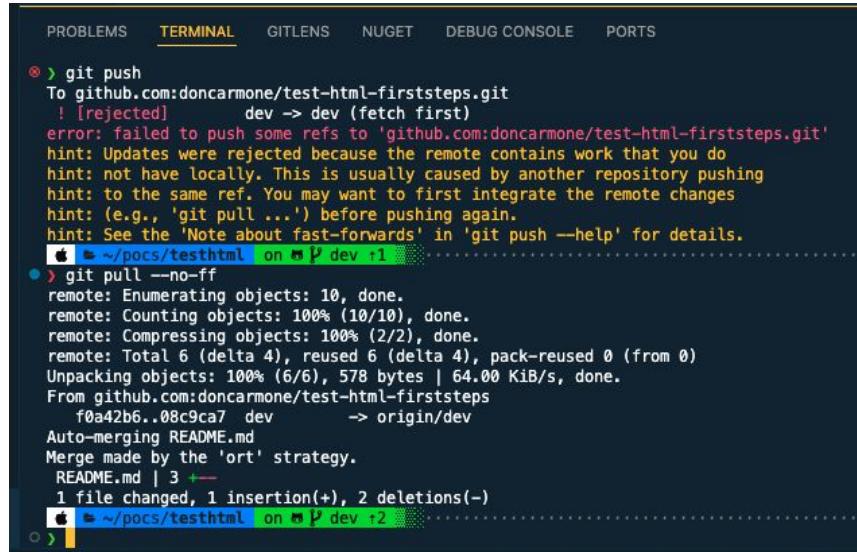
```
git commit -m "message"
```

# Basic Conflict resolution



The screenshot shows the VS Code interface with the 'TERMINAL' tab selected. The title bar indicates 'UW PICO 5.09' and the file path 'File: /Users/carmona/pocs/testhtml/.git/MERGE\_MSG'. The terminal content displays a merge message template:

```
Merge branch 'dev' of github.com:doncarmone/test-html-firststeps into dev
# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
```



The screenshot shows a terminal window with the 'TERMINAL' tab selected. The user has run a 'git push' command, which failed because the remote repository contained work that was not locally available. The output shows the error message and some hints:

```
✖ > git push
To github.com:doncarmone/test-html-firststeps.git
 ! [rejected]      dev -> dev (fetch first)
error: failed to push some refs to 'github.com:doncarmone/test-html-firststeps.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
apple ➜ ~/pocs/testhtml on M P dev +1
```

After the failed push, the user runs a 'git pull --no-ff' command to update their local repository:

```
● > git pull --no-ff
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 4), reused 6 (delta 4), pack-reused 0 (from 0)
Unpacking objects: 100% (6/6), 578 bytes | 64.00 KiB/s, done.
From github.com:doncarmone/test-html-firststeps
 f0a42b6..08c9ca7 dev -> origin/dev
Auto-merging README.md
Merge made by the 'ort' strategy.
 README.md | 3 +--
 1 file changed, 1 insertion(+), 2 deletions(-)
apple ➜ ~/pocs/testhtml on M P dev +2
```



# Practice time!

# Further steps

Take a in a deep git course:

[Tutorial de Git Makigas](#)

Knowing about undo commits.

Know about rename, delete, and manage branches.

Knowing about stash command.

Knowing about git alias.

Knowing about internal conflict resolution (Rebase vs Fast Forward).

Knowing about cherry pick command.

# WIZELINE

# Thanks!