

# Aprende a Programar

Por Guillermo Farías

# Agenda

## Sesión 1

- Introducción
- Teoría
- HTML

## Sesión 2

- HTML
- CSS
- ~~Bootstrap~~

## Sesión 3

- CSS
- Bootstrap

## Sesión 4

- Bootstrap
- JavaScript
- jQuery

## Sesión 5

- jQuery

## Sesión 6

- Ejercicios Frontend
- Git

## Sesión 7

- Servers
- MySQL

## Sesión 8 a 10

- MVC
- Ruby on Rails

# Actividad 1

Crea una página para una tienda con un menú de navegación horizontal en la parte superior.

Al dar click en cada item del menu debe de agregarle la clase “active”

Utilizar:

Bootstrap  
jQuery



# Actividad 2

Continuando sobre la actividad anterior, cuando des click en un elemento del menu debe de quitarle la clase active a todos los demás items.



# Actividad 3

Agregar un 2do nivel de navegación en el costado izquierdo de la página y replicar el mismo funcionamiento de las actividades anteriores, al dar click activar la opción seleccionada.

The image shows a navigation bar at the top with several categories: eCommerce, Libros (which is underlined), Tecnología, Animales, and Electrodomésticos. Below this, a sidebar is visible on the left side of the page, containing a list of sub-categories under the heading 'Negocios': Tecnología, Ficción, and Novelas.

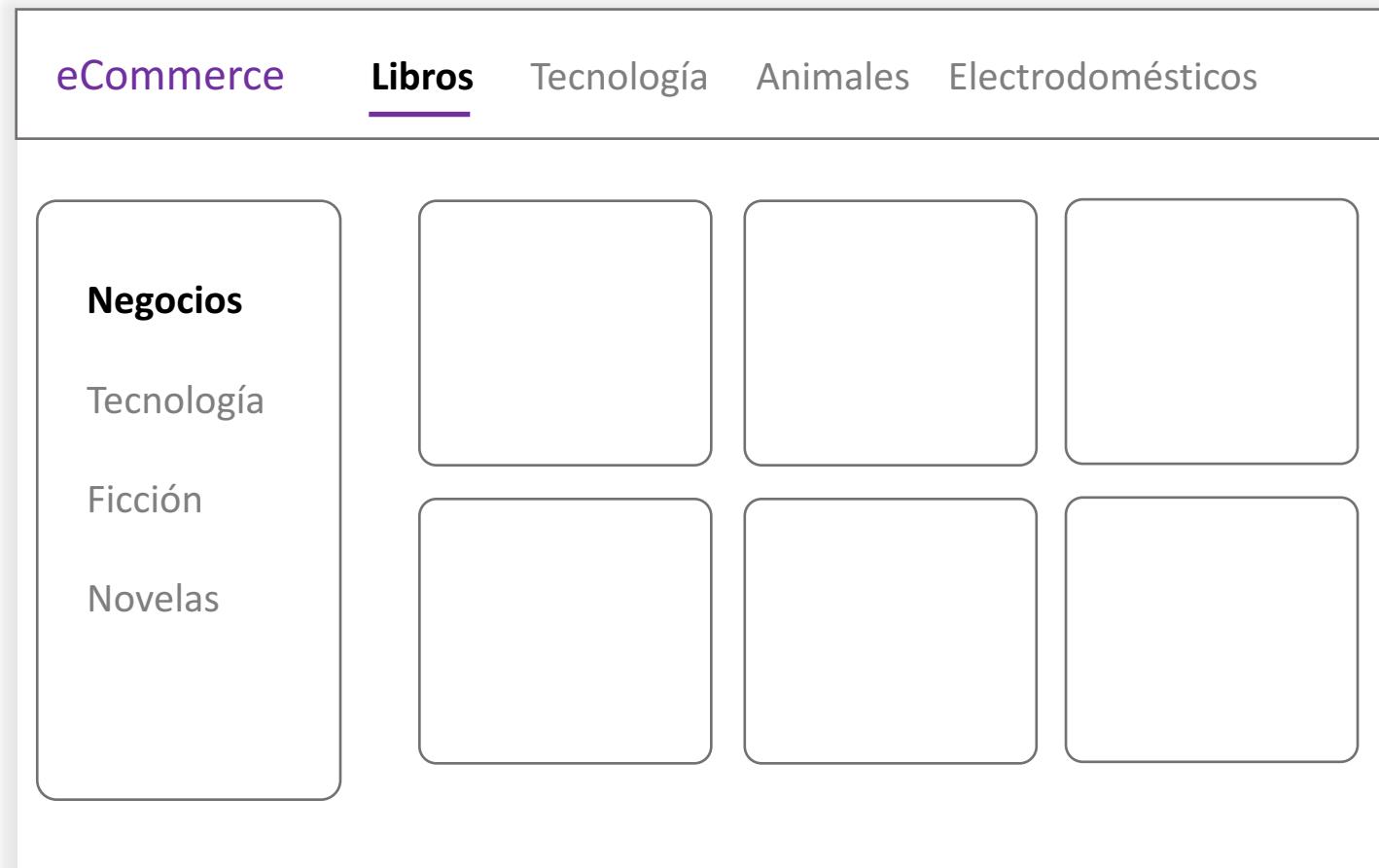
- eCommerce
- Libros**
- Tecnología
- Animales
- Electrodomésticos

**Negocios**

- Tecnología
- Ficción
- Novelas

# Actividad 4

Crear una galería de tarjetas de productos y que al dar click en una abra su detalle un modal.



# Actividad 5

Implementar un slideshow compatible con jQuery y Bootstrap

The image shows a user interface for a website. At the top, there is a navigation bar with several categories: eCommerce, Libros, Tecnología, Animales, and Electrodomésticos. The 'Libros' category is highlighted with a purple underline. Below the navigation bar, there is a slide show area. The first slide in the show contains four book titles: Negocios, Tecnología, Ficción, and Novelas. To the right of this slide, there is a large, light-gray rectangular area with a left-pointing arrow on the left side and a right-pointing arrow on the right side, indicating that more slides can be viewed by clicking these arrows.

eCommerce   **Libros**   Tecnología   Animales   Electrodomésticos

Negocios  
Tecnología  
Ficción  
Novelas

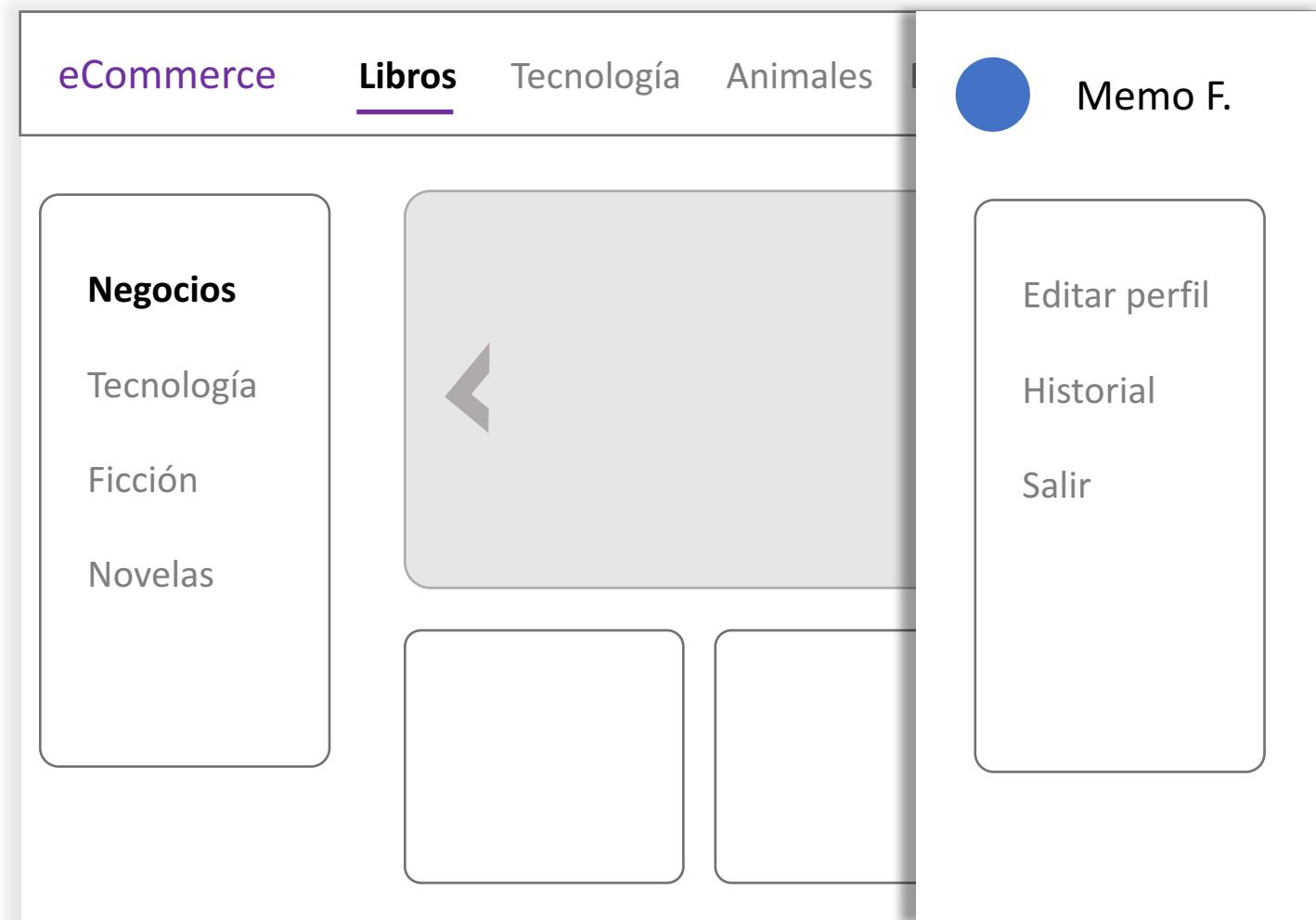
# Actividad 6

Agregar la funcionalidad para que al cargar la página despliegue un modal que contiene un texto y un campo para inscribirte a un newsletter para recibir promociones.



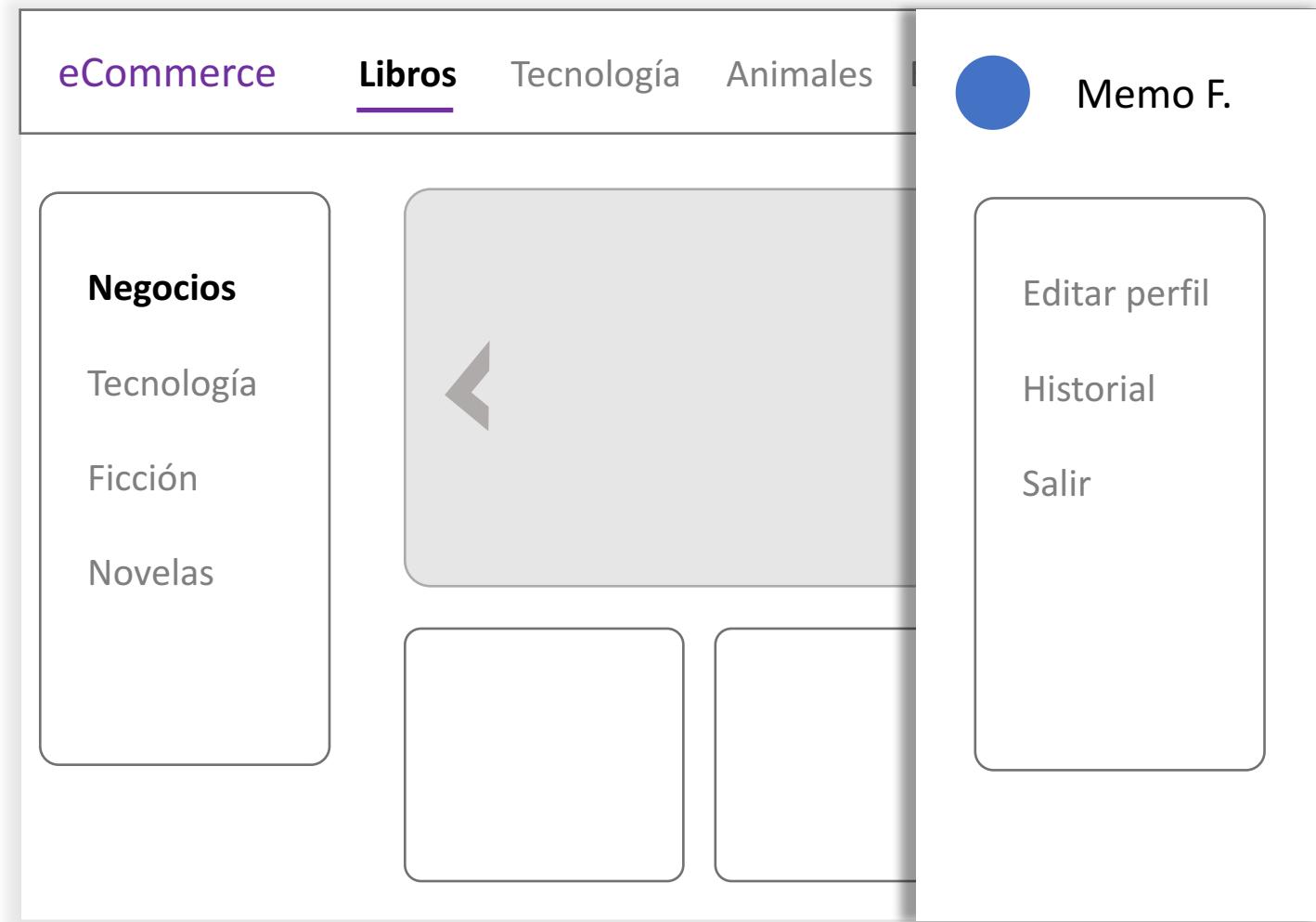
# Actividad 7

Crear un menu de navegación dentro de un cajón lateral del lado derecho de la pantalla.



# Actividad 8

Agregar un botón que abra  
y cierre el cajón.



# Actividad 9

Crear un arreglo de objetos en JSON con al menos tres atributos y al menos un atributo añadido.

Ejemplo:

Libros

Tecnología

Sub-categorías

Computadoras

Audio y Video

Móviles y Tablets

Redes



**Git**



Fast, scalable, distributed revision control system



Zona de  
Trabajo

SONRIAL  
LOS DEDOS  
FILMANDO



TWITTER

#FABRICATOR

@GCBAdata

CRONOGRAMA



CRONOGRAMA



# Git 101: Commit

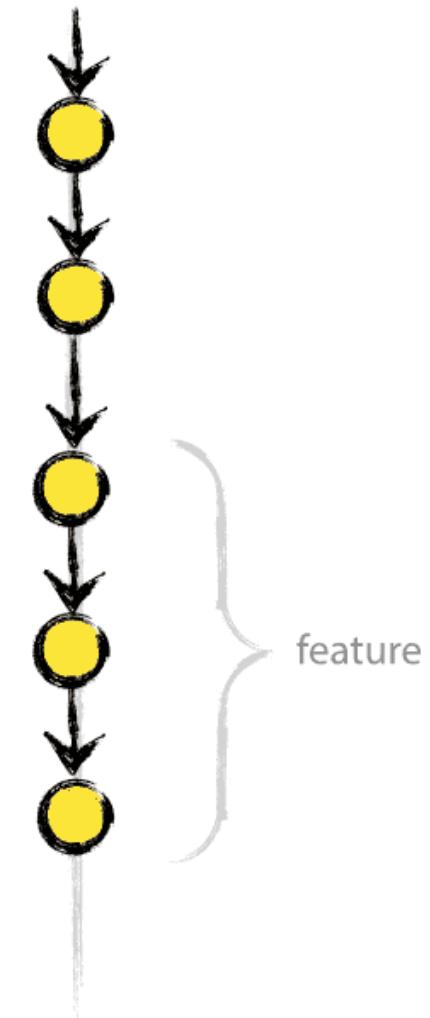
```
git commit -am "fixed typo"
```

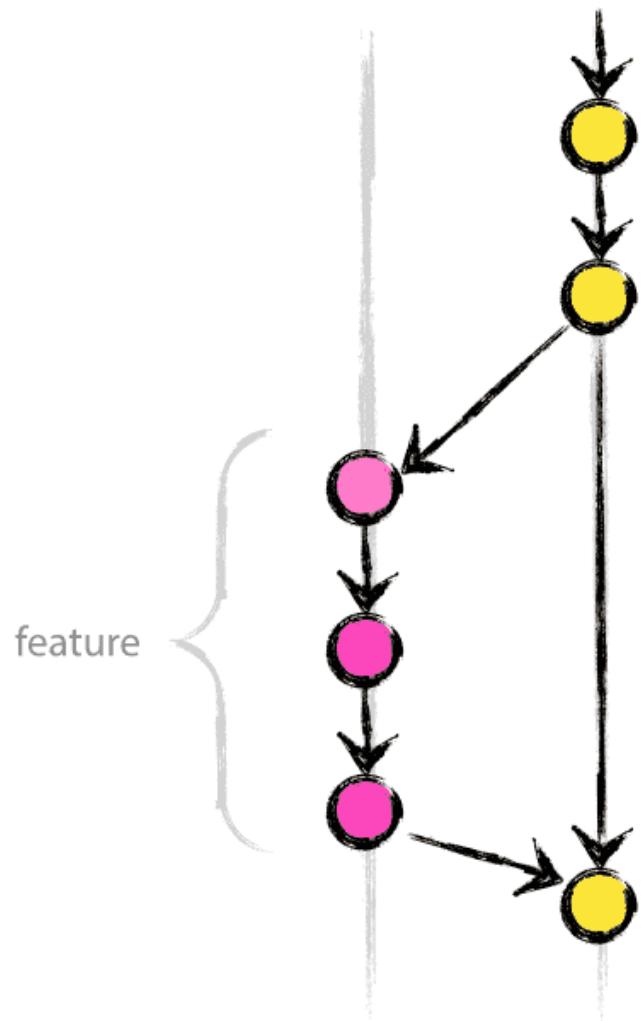
```
git commit -am "adding user's phone field"
```

```
git commit -am "allow users to join a company"
```

```
git commit -am "companies can have a logo"
```

```
git commit -am "companies url field"
```

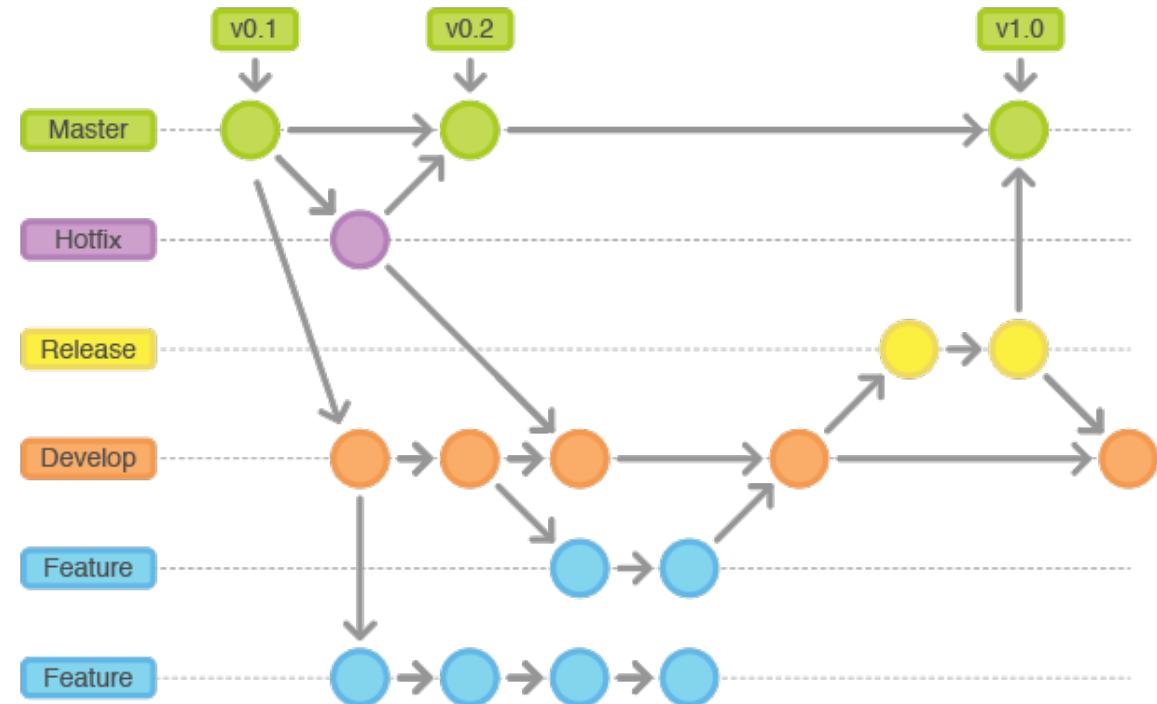




# Git 101

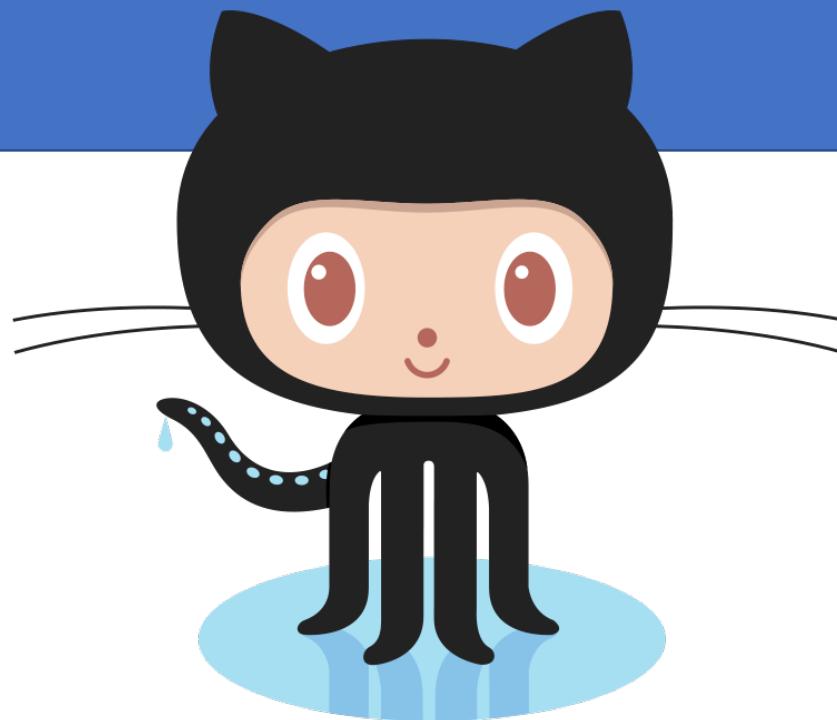
Métodos más comunes:

- init
- add
- commit
- push
- pull
- remote (origin)
- clone
- checkout
- status



## Actividad

1. Crear su cuenta en Github.com
2. Descargar el cliente de <https://desktop.github.com/>

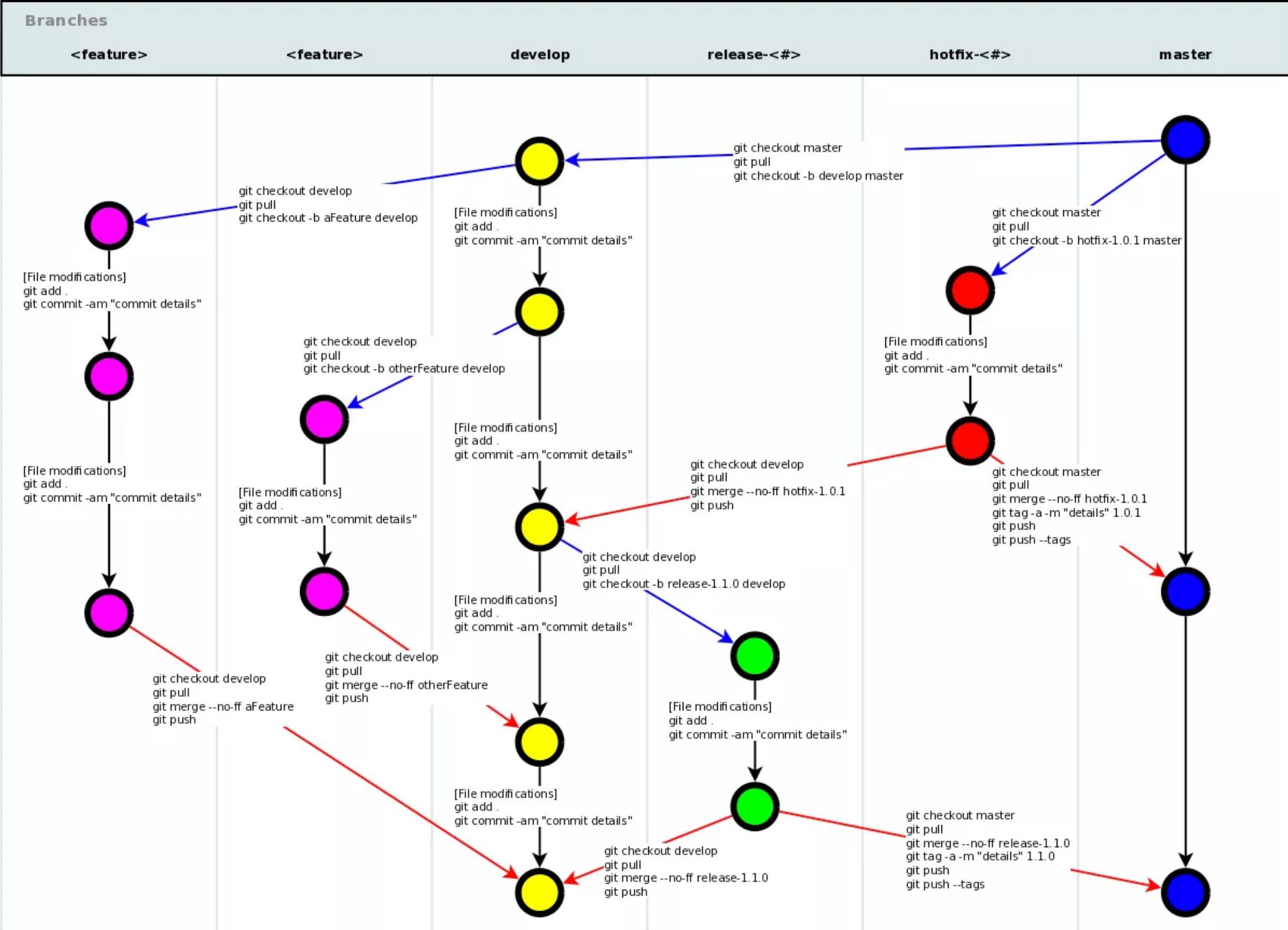


# GitHub

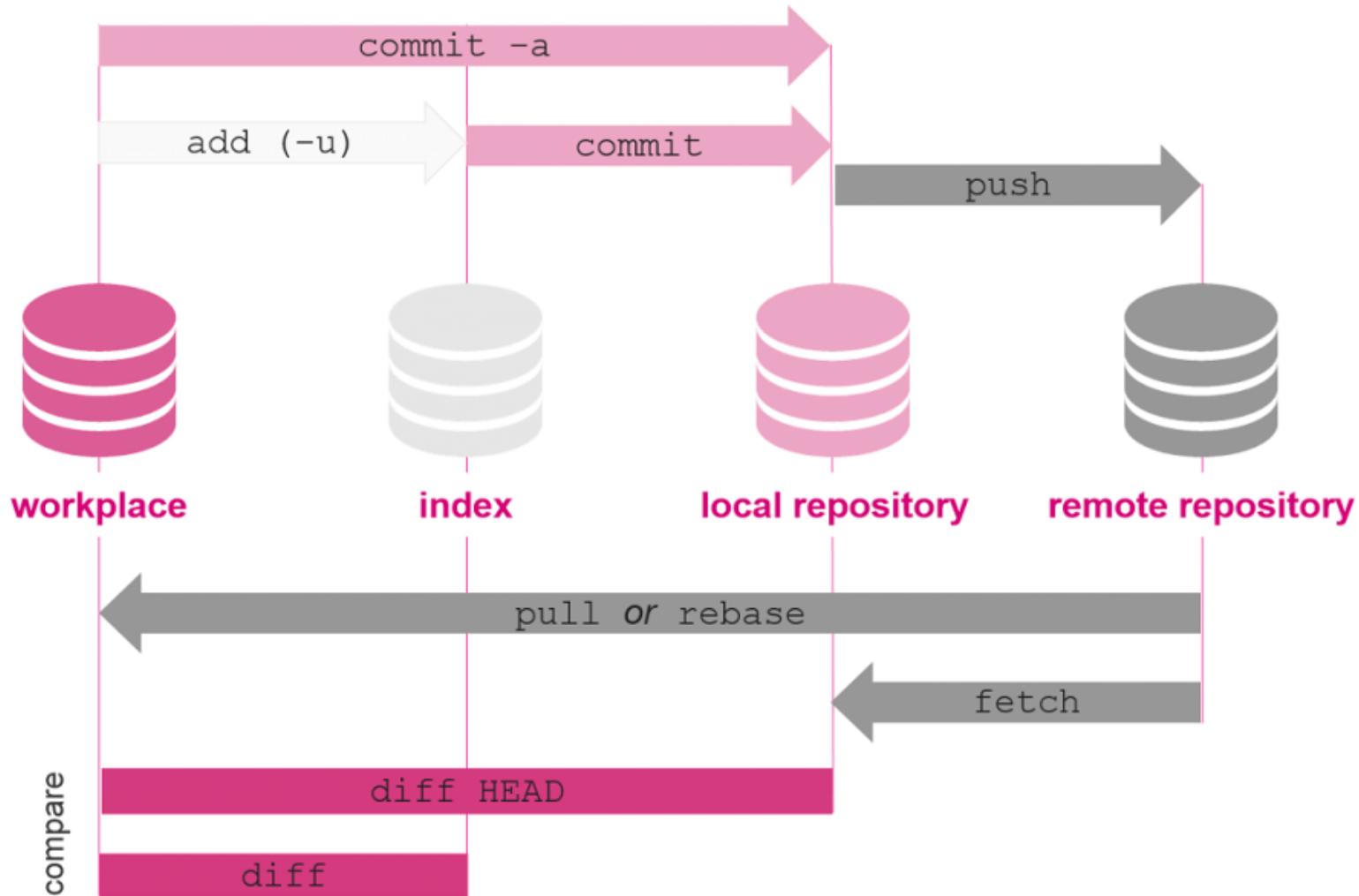


# Referencias

- Ref. <https://www.w3schools.com/jsref/default.asp>
- Ref. <http://getbootstrap.com>
- Ref. <https://jquery.com/>
- Ref. <https://git-scm.com/>
- Ref. <https://learngitbranching.js.org/>
- Ref. <https://www.phpied.com/3-ways-to-define-a-javascript-class/>
- Ref. <https://javascript.info/class>



# Git Data Transport Commands



# Git Cheat Sheet

## ★ Create

From existing data

```
cd ~/my_project_directory  
git init  
git add .
```

From existing repository

```
git clone ~/existing_repo ~/new/repo  
git clone git://host.org/project.git  
git clone ssh://user@host.org/project.git
```

## ★ Show

Files changed in working directory

```
git status
```

Changes made to tracked files

```
git diff
```

What changed between ID1 and ID2

```
git diff <ID1> <ID2>
```

History of changes

```
git log
```

History of changes for file with diffs

```
git log -p <FILE> <DIRECTORY>
```

Who changed what and when in a file

```
git blame <FILE>
```

A commit identified by ID

```
git show <ID>
```

A specific file from a specific ID

```
git show <ID>:<FILE>
```

All local branches

```
git branch  
star (*) marks the current branch
```

## ★ Revert

Return to the last committed state

```
git reset --hard  
This cannot be undone!
```

Revert the last commit

```
git revert HEAD  
Creates a new commit
```

Revert specific commit

```
git revert <ID>  
Creates a new commit
```

Fix the last commit

```
git commit -a --amend  
(after editing the broken files)
```

Checkout the ID version of a file

```
git checkout <ID> <FILE>
```

## ★ Update

Fetch latest changes from origin

```
git fetch  
(this does not merge them)
```

Pull latest changes from origin

```
git pull  
(does a fetch followed by a merge)
```

Apply a patch that someone sent you

```
git am -3 patch.mbox  
In case of conflict, resolve the conflict and  
git am --resolved
```

## ★ Publish

Commit all your local changes

```
git commit -a
```

Prepare a patch for other developers

```
git format-patch origin
```

Push changes to origin

```
git push
```

Make a version or milestone

```
git tag v1.0
```

Remember!  
`git <COMMAND> --help`

Global configuration is stored in `~/.gitconfig`.  
`git config --help`

`master` is the default development branch.  
`origin` is the default upstream repository.

## ★ Branch

Switch to a branch

```
git checkout <BRANCH>
```

Merge BRANCH1 into BRANCH2

```
git checkout <BRANCH2>  
git merge <BRANCH1>
```

Create branch BRANCH based on HEAD

```
git branch <BRANCH>
```

Create branch BRANCH based on OTHER and switch to it

```
git checkout -b <BRANCH> <OTHER>
```

Delete branch BRANCH

```
git branch -d <BRANCH>
```

## ★ Resolve merge conflicts

View merge conflicts

```
git diff
```

View merge conflicts against base file

```
git diff --base <FILE>
```

View merge conflicts against your changes

```
git diff --ours <FILE>
```

View merge conflicts against other changes

```
git diff --theirs <FILE>
```

Discard a conflicting patch

```
git reset --hard
```

```
git rebase --skip
```

After resolving conflicts, merge with

```
git add <CONFLICTING_FILE>
```

```
git rebase --continue
```

## ★ Workflow

