

# Git y Github, claves para el desarrollo en abierto

Germán Martínez Maldonado

Curso de Ciencia Abierta  
Escuela de Doctorado de Ciencias, Tecnologías e Ingenierías

# Objetivos (Qué deberíamos saber ya)

---

- ¿Qué es el software libre? ¿Cuáles son las ventajas del conocimiento abierto?
- ¿Cómo “liberar” software libre?
- ¿Cómo “liberar” conocimiento abierto?

# Objetivos (Qué queremos aprender)

---

- El estado mental correcto
- Sistemas de control de versiones
- Por qué usar Git y GitHub
- Uso básico de Git y GitHub
- Flujos de trabajo básicos
- Colaboración a través de repositorios de código

# El estado mental correcto

---

- Trabajos interminables
- Trabajos en los que vamos a equivocarnos.
- “copia”, “versión buena”, “versión buena de verdad”...

# El estado mental correcto

---

- Las máquinas están para suplir nuestras carencias.
- Las máquinas son mejores organizadores que nosotros.
- Solo necesitamos saber a dónde volver.
- Proceso muy cómodo.

# Sistema de control de versiones

---

- Colaboración
- Almacenamiento de versiones
- Restauración de versiones previas
- Restauración de versiones previas
- Comprensión de los errores
- Copias de seguridad

# Sistema de control de versiones

---

## Colaboración

- No vamos a trabajar solos
- La comunicación siempre presenta problemas
- El sistema se encarga que nos pisemos unos a otros

# Sistema de control de versiones

---

## Almacenamiento de versiones

- ¿Qué vamos a copiar?
- Política de nombres
- Conocer diferencias



# Sistema de control de versiones

---

Restauración de versiones previas

- Sin miedo a equivocarnos
- Fácil marcha atrás
- Sistemas accesible

# Sistema de control de versiones

---

## Comprensión de los errores

- Nada se pierde, todo se recupera
- Aprovechar para aprender de los errores
- Facilidad para encontrar los errores

# Sistema de control de versiones

---

## Copias de seguridad

- Sistemas preferiblemente distribuidos
- Tantas copias locales como colaboradores haya
- Siempre podemos recuperar

# Git y GitHub

---

¿Por qué usar Git?

- Es software libre
- Fue creado con la idea de contribuir al software libre
- Alternativas: SVN, Mercurial...

# Git y GitHub

---

¿Por qué usar Git?

- Si usas SVN...
- Ventajas de sistemas descentralizados
- Comodidad a la hora de subir los cambios al servidor

# Git y GitHub

---

¿Por qué usar Git?

- Si usas Mercurial...
- Software que hacen lo mismo, pero de forma distinta
- Grandes diferencias en el mezclado de cambios, etiquetado de versiones y gestión de ramas.

# Git y GitHub

---

¿Por qué usar Git?

- Comodidad
- Eficiencia
- Integridad

# Git y GitHub

---

¿Por qué usar Git?

- Flujos de trabajo para el desarrollo
- Mantenimiento de proyectos
- Despliegue



# Git y GitHub

---

¿Por qué usar Git?

- Enorme ecosistema de Git
- Gran cantidad de herramientas
- “Hooks”

# Git y GitHub

---

¿Por qué usar Git?

- Gestión de ramas
- Gestión de etiquetas
- Versiones de desarrollo, versiones de lanzamiento

# Git y GitHub

---

¿Por qué usar Git?

- Todo lo anterior
- Es el más usado
- Mayor soporte

# Git y GitHub

---

¿Por qué usar GitHub?

- Forjas
- Ya no solo orientadas a software
- La ciencia hoy en día es algo más que texto

# Git y GitHub

---

¿Por qué usar GitHub?

- Plataforma de desarrollo colaborativo
- Proyectos multidisciplinarios con varios integrantes
- Facilitan la difusión y el soporte

# Git y GitHub

---

¿Por qué usar GitHub?

- Se creó para trabajar sobre Git
- También existe BitBucket (y GitLab) pero...
- Si Git es el más extendido...
- El dilema de la licencia (y GitLab)

# Git y GitHub

---

¿Por qué usar GitHub?

- También podríamos usar GitLab
- El funcionamiento (incluso a nivel web) es prácticamente idéntico
- Solo encontramos diferencias de usabilidad

# Git y GitHub

---

¿Por qué usar GitHub?

- GitLab permite definir permisos a nivel de rol.
- GitLab permite adjuntar archivos a los issues.
- GitLab permite cambiar el estatus de diferentes issues de una vez.
- GitLab permite etiquetar proyectos como WIP.
- GitLab tiene una versión que podemos instalar en nuestro servidor.



# Git y GitHub

---

¿Por qué usar GitHub?

- Aparentemente GitLab tiene más opciones de vista al usuario
- Pero GitHub cumple perfectamente con su función
- GitHub llegó antes

# Uso básico de Git

---

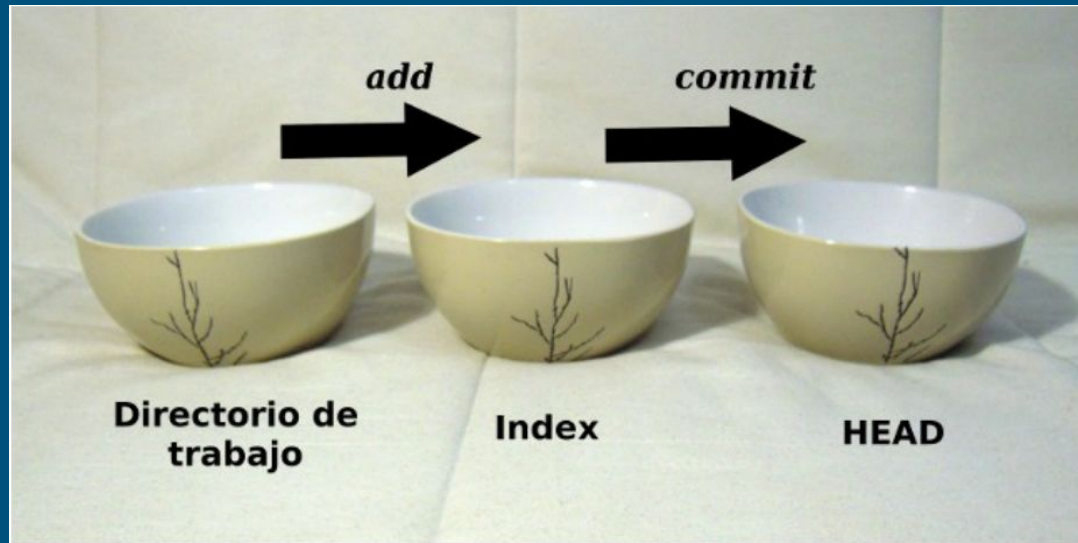
## Configuración:

- `$ git config --global user.name "German Martinez"`
- `$ git config --global user.email germaaan@protonmail.com`
- `$ git config --global core.editor nano`
- `$ git config --global merge.tool vimdiff`
  
- `$ git config --list`

# Uso básico de Git

---

Los tres cuencos (original por [Psicobyte](#))



# Uso básico de Git

---

## Configuración SSH:

- `$ ssh-keygen -t rsa -b 4096 -C germaaan@protonmail.com`
- `$ eval "$(ssh-agent -s)"`
- `$ ssh-add ~/.ssh/id_rsa`

Añadir la clave en "`~/.ssh/id_rsa.pub`" a GitHub.

- `$ ssh -T git@github.com`

# Uso básico de Git

---

Inicializando un repositorio y añadiendo origen remoto:

- `$ git init`
- `$ git remote add origin  
git@github.com:JJ/doctorado-ciencia-abierta.git`

Clonando un repositorio:

- `$ git clone git@github.com:JJ/doctorado-ciencia-abierta.git`

# Uso básico de Git

---

Añadir el archivo al índice:

- `$ git add NOMBREARCHIVO`
- `$ git add NOMBREDIRECTORIO`
- `$ git add .`

# Uso básico de Git

---

Pasar del índice al HEAD:

- `$ git commit`
- `$ git commit -m "COMENTARIO"`
- `$ git commit -am "COMENTARIO"`
- `$ git commit -amend "COMENTARIO"`

# Uso básico de Git

---

Enviar cambios al servidor remoto:

- `$ git push`
- `$ git push origin master`
- `$ git push ORIGEN RAMA`
  
- `$ git push -f (!!!)`



# Uso básico de Git

---

Traer cambios del servidor remoto:

- `$ git pull`
- `$ git pull origin master`
- `$ git pull ORIGEN RAMA`

# Uso básico de Git

---

Comprobar estado actual de nuestros archivos:

- `$ git status`

Deshacer todo los cambios temporales para restablecer las versiones en HEAD.

- `$ git stash`

# Uso básico de Git

---

Trabajando con ramas:

- `$ git branch`
- `$ git branch -a`
- `$ git branch -d NOMBRE RAMA`

# Uso básico de Git

---

Trabajando con ramas:

- `$ git checkout`
- `$ git checkout NOMBRRERAMA`
- `$ git checkout -b NOMBRRERAMANUEVA`
- `$ git checkout -t origin/NOMBRRERAMA`

# Uso básico de Git

---

Mezclando ramas:

- \$ git checkout master
  - \$ git merge dev
  - \$ git pull origin dev
  - \$ git push

# Uso básico de Git

---

## Trabajando con etiquetas

- `$ git tag`
- `$ git tag -l "v1.1.*"`
- `$ git tag -a v1.2 -m "MENSAJE"`
- `$ git show v1.2`

# Uso básico de Git

---

Eliminando archivos:

- `$ git rm NOMBREARCHIVO`
- `$ git rm --cached NOMBREARCHIVO`

Moviendo archivos:

- `$ git mv NOMBREARCHIVO`

# Uso básico de Git

---

Eliminando archivos:

- `$ git rm NOMBREARCHIVO`
- `$ git rm --cached NOMBREARCHIVO`

Moviendo archivos:

- `$ git mv NOMBREARCHIVO`



# Uso básico de Git

---

Eliminando cambios:

- Eliminar los cambios de los que aún no se ha hecho commit.
  - `$ git reset --hard HEAD`
- Eliminar los cambios de los que se ha hecho commit.
  - `$ git revert COMMIT`

# Uso básico de Git

---

- Filosofía open: “si no sabes pregunta”
  - `$ git help COMANDO`
- Muchos comandos más: blame, diff, fetch, rebase...

# Uso básico de GitHub

The screenshot shows the GitHub homepage with a dark header bar. The header includes the GitHub logo, a search bar, and navigation links for Pull requests, Issues, Marketplace, and Explore. On the right side of the header, there are icons for notifications, a plus sign, and a profile picture.

Below the header, the main content area is divided into three sections:

- Activity:** A list of recent activity on GitHub, including stars and forks. The list is filtered by the user 'germaan'. The activity includes:
  - elbault starred adriennefriend/imposter-syndrome-disclaimer (3 hours ago)
  - camosposfuentes forked camosposfuentes/react-native-counter from Kerumen/react-native-counter (5 hours ago)
  - toblerone554 forked toblorone554/starboard-lsdrv from mmuman/starboard-lsdrv (5 hours ago)
  - andreshp starred rjraya/Field-Theory (5 hours ago)
  - elbault forked elbault/scalatest from scalatest/scalatest (5 hours ago)
  - JuanjoSalvador starred Juanlu001/welcome-tg-bot (16 hours ago)
  - yabirgb forked yabirgb/Estructuras-de-Datos from Ocete/Estructuras-de-Datos (17 hours ago)
  - rafaelleru starred mlpack/mlpack (19 hours ago)
- Repositories you contribute to:** A list of repositories that the user has contributed to, including:
  - Interferencias/interferencias.git...
  - oslugr/ugr-transparente-servidor
  - oslugr/proyectosHackathonUGR
  - Interferencias/presentacionAso...
  - Interferencias/materialTalleresC...
- Your repositories:** A list of repositories owned by the user, including:
  - JJ/doctorado-ciencia-abierta
  - oslugr/sitio9jr
  - oslugr/TF-libres-UGR
  - asdfsdfas
  - JJ/SerU
  - presenciaDigital

On the right side of the page, there is a sidebar with a notification banner that says "A more connected universe" and a button to "View 6 new broadcasts".

# Uso básico de GitHub

The screenshot shows the GitHub homepage with a dark header bar. The header includes the GitHub logo, a search bar, and navigation links for Pull requests, Issues, Marketplace, and Explore. On the right side of the header, there are icons for notifications, a plus sign, and a profile picture.

Below the header, the main content area is divided into three sections:

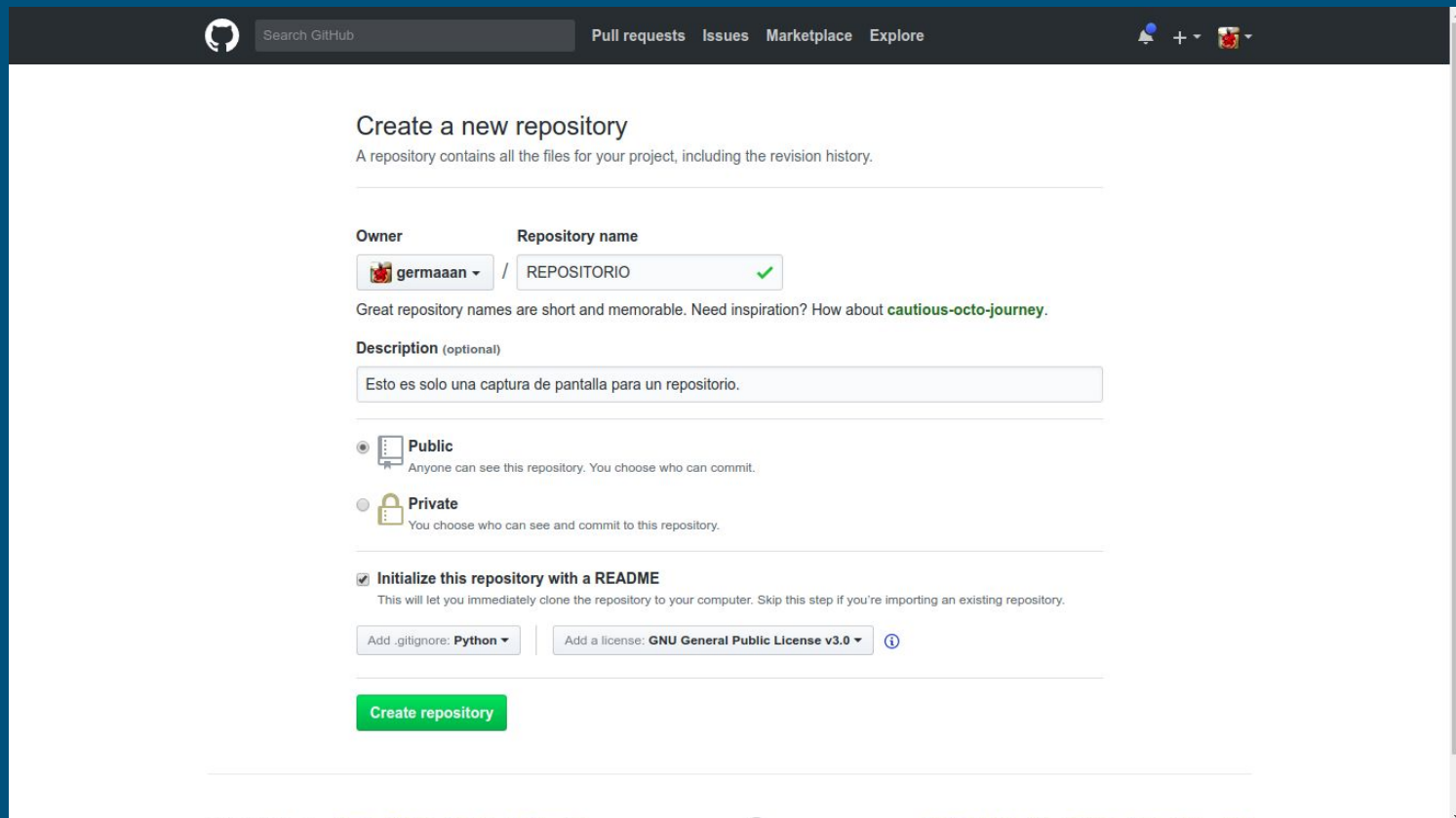
- Activity:** A list of recent activity on GitHub, including stars and forks. The list is filtered by the user 'germaan'. The activity includes:
  - elbault starred adriennefriend/imposter-syndrome-disclaimer (3 hours ago)
  - camosposfuentes forked camosposfuentes/react-native-counter from Kerumen/react-native-counter (5 hours ago)
  - toblerone554 forked toblorone554/starboard-lsdrv from mmuman/starboard-lsdrv (5 hours ago)
  - andreshp starred rjraya/Field-Theory (5 hours ago)
  - elbault forked elbault/scalatest from scalatest/scalatest (5 hours ago)
  - JuanjoSalvador starred Juanlu001/welcome-tg-bot (16 hours ago)
  - yabirgb forked yabirgb/Estructuras-de-Datos from Ocete/Estructuras-de-Datos (17 hours ago)
  - rafaelleru starred mlpack/mlpack (19 hours ago)
- Repos you contribute to:** A list of repositories that the user has contributed to, including:
  - Interferencias/interferencias.git...
  - oslugr/ugr-transparente-servidor
  - oslugr/proyectosHackathonUGR
  - Interferencias/presentacionAso...
  - Interferencias/materialTalleresC...
- Your repositories:** A list of repositories owned by the user, including:
  - JJ/doctorado-ciencia-abierta
  - oslugr/sitio9jr
  - oslugr/TF-libres-UGR
  - asdfsdfas
  - JJ/SerU
  - presenciaDigital

On the right side of the page, there is a sidebar with a notification banner that says "A more connected universe" and a button to "View 6 new broadcasts".

\_\_\_\_\_

[illegible]

# Uso básico de GitHub



The screenshot shows the GitHub interface for creating a new repository. At the top is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. On the right of the bar are notification, user, and organization icons. The main content area is titled 'Create a new repository' with a subtitle explaining that a repository contains all project files and revision history. Below this, there are two input fields: 'Owner' (set to 'germaan') and 'Repository name' (set to 'REPOSITORIO' with a green checkmark). A tip suggests repository names should be short and memorable, with an example 'cautious-octo-journey'. The 'Description' field is optional and contains the text 'Esto es solo una captura de pantalla para un repositorio.' Below the description, there are two radio button options for visibility: 'Public' (selected) and 'Private'. The 'Public' option includes a subtext: 'Anyone can see this repository. You choose who can commit.' The 'Private' option includes a subtext: 'You choose who can see and commit to this repository.' There is a checked checkbox for 'Initialize this repository with a README', with a subtext: 'This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.' At the bottom of the form, there are two dropdown menus: 'Add .gitignore: Python' and 'Add a license: GNU General Public License v3.0', followed by an information icon. A large green 'Create repository' button is at the bottom of the form.

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner / Repository name

germaan / REPOSITORIO ✓

Great repository names are short and memorable. Need inspiration? How about [cautious-octo-journey](#).

Description (optional)

Esto es solo una captura de pantalla para un repositorio.

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

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

☒ Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: Python Add a license: GNU General Public License v3.0 ⓘ

Create repository

# Uso básico de GitHub

The screenshot shows a GitHub repository page for 'germaaan / REPOSITORIO'. The repository is in the 'master' branch and has 1 commit, 1 branch, 0 releases, and 1 contributor. The repository contains three files: .gitignore, LICENSE, and README.md, all committed 'just now'. The README.md file is displayed below the file list, showing the title 'REPOSITORIO' and the placeholder text 'Esto es solo una captura de pantalla para un repositorio.'.

germaaan / REPOSITORIO

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Esto es solo una captura de pantalla para un repositorio. [Edit](#)

[Add topics](#)

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request

Create new file Upload files Find file Clone or download

germaaan Initial commit	Latest commit 7540a05 just now
<a href="#">.gitignore</a>	Initial commit just now
<a href="#">LICENSE</a>	Initial commit just now
<a href="#">README.md</a>	Initial commit just now

README.md

## REPOSITORIO

Esto es solo una captura de pantalla para un repositorio.

# Uso básico de GitHub

The screenshot shows the GitHub interface for the repository 'angrykoala/gaucha'. The top navigation bar includes links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The repository name is 'angrykoala / gaucha', with 3 Watchers, 35 Unstars, and 22 Forks. The 'Issues' tab is selected, showing 37 issues. A banner prompts users to submit issues, advising them to read the contributing guidelines. Below the banner, a search bar shows the filter 'is:issue is:open'. A table of issues is displayed, with columns for Author, Labels, Projects, Milestones, Assignee, and Sort. The issues listed are:

- Forcing reflow warning message when opening modal** (bug, Hacktoberfest) - #202, opened 5 days ago by angrykoala.
- Overall architecture refactoring** (enhancement, maintenance) - #198, opened 5 days ago by angrykoala. Progress: 0 of 1, 0.4.
- Sortablejs warning** (bug) - #196, opened 6 days ago by angrykoala. 1 comment.
- After "Clear Tasks" trying to add a new task won't work** (bug, critical) - #193, opened 6 days ago by angrykoala. 0.4.
- Tasks data appear to delete randomly** (bug) - #189, opened 8 days ago by angrykoala. 2 comments.
- Consider replacing chai assert for built-in assert module in tests** (discussion) - #183, opened 10 days ago by angrykoala.



# Uso básico de GitHub

The screenshot shows a GitHub repository page for 'angrykoala / gaucho'. The repository has 37 issues, 4 pull requests, 0 projects, and 0 wiki pages. The current issue is titled 'Forcing reflow warning message when opening modal #202' and is marked as 'Open'. It was opened by 'angrykoala' 5 days ago and has 0 comments. The issue description includes a code snippet: `[Violation] Forced reflow while executing JavaScript took 89ms`, followed by the text: 'This warning message appears when a modal (using sweetalert2) is opened. Investigate why this message appears'. The issue has two labels: 'bug' and 'Hacktoberfest'. The right sidebar shows 'Assignees' (No one assigned), 'Labels' (bug, Hacktoberfest), 'Projects' (None yet), 'Milestone' (No milestone), and 'Notifications' (You're not receiving notifications from this thread). The bottom of the page shows a 'Write' button and a 'Preview' button, with a text area for leaving a comment.

angrykoala / gaucho

Watch 3 Unstar 35 Fork 22

Code Issues 37 Pull requests 4 Projects 0 Wiki Insights

## Forcing reflow warning message when opening modal #202

**Open** angrykoala opened this issue 5 days ago · 0 comments

angrykoala commented 5 days ago **Owner**

```
[Violation] Forced reflow while executing
JavaScript took 89ms
```

This warning message appears when a modal (using sweetalert2) is opened.

Investigate why this message appears

angrykoala added **bug** **Hacktoberfest** labels 3 days ago

angrykoala referenced this issue 3 days ago

### Sortablejs warning #196

**Open**

**Assignees**  
No one assigned

**Labels**  
**bug**  
**Hacktoberfest**

**Projects**  
None yet

**Milestone**  
No milestone

**Notifications**  
**Subscribe**  
You're not receiving notifications from this thread.

Write Preview

AA B i " < > : : : : @

Leave a comment

# Uso básico de GitHub

The screenshot shows the GitHub interface for the repository `angrykoala / gaucho`. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. The repository header shows 3 Watchers, 35 Unstars, and 22 Forks. The main content area displays a message for first-time contributors and a list of pull requests. The pull requests list includes details such as the title, issue number, author, and status.

**First time contributing to angrykoala/gaucho?** [Dismiss](#)

If you know how to fix an [issue](#), consider opening a pull request for it.  
You can read this repository's [contributing guidelines](#) to learn how to open a good pull request.

Filters  Labels Milestones [New pull request](#)

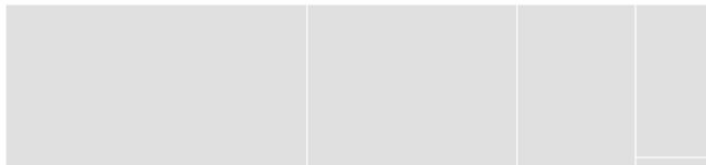
4 Open	✓ 40 Closed	Author	Labels	Projects	Milestones	Reviews	Assignee	Sort
<b>log should auto scroll</b> ✓								5
#207 opened 2 days ago by ssteuteville  4 of 5								
<b>Import tasks</b>								
#206 opened 4 days ago by grg121								
<b>Moving tasks between suits.</b> ✓								8
#199 opened 5 days ago by sergeir82 • Changes requested								
<b>Context menu</b> ✓								12
#197 opened 5 days ago by ssteuteville  4 of 4								

**ProTip!** `no:milestone` will show everything without a milestone.

# Uso básico de GitHub

## Codecov Report

Merging #205 into dev will **not change** coverage.  
The diff coverage is `n/a`.



@@	Coverage Diff		@@
##	dev	#205	+/ -
=====			
Coverage	94.69%	94.69%	
=====			
Files	5	5	
Lines	132	132	
Branches	21	21	
=====			
Hits	125	125	
Misses	7	7	

[Continue to review full report at Codecov.](#)

**Legend** - [Click here to learn more](#)

$\Delta$  = absolute <relative> (impact),  $\emptyset$  = not affected,  $?$  = missing data

Powered by [Codecov](#). Last update `3cbbf22...bc20de9`. Read the [comment docs](#).



**angrykoala** merged commit `1b8f58c` into `angrykoala:dev` 4 days ago

3 checks passed

[View details](#)

# Uso básico de GitHub



# Flujos de trabajo básicos

---

- [Crear repositorio | Hacer fork de un repositorio existente]
- [Crear una nueva rama]
- Crear milestone
- Crear issues
- Añadir commits que resuelvan los issues.
- [Mezclar ramas]
- [Hacer Pull Request]

# Flujos de trabajo básicos

---

Ejemplo: Vamos a añadir el capítulo de introducción de nuestro paper.

- `$ git checkout -b introduccion`
- `$ git add introduccion.tex`
- `$ git commit -am "Closes #1"`
- `$ git push`
- `$ git checkout master`
- `$ git merge`

# Flujos de trabajo básicos

---

Ejemplo: Vamos a añadir el capítulo de introducción de nuestro paper.

- `$ git checkout -b introduccion`
- `$ git add introduccion.tex`
- `$ git commit -am "Closes #1"`
- `$ git push`
- `$ git checkout master`
- `$ git merge`

# Colaboración a través de repositorios de código

- README explicativo sobre qué trata el proyecto
- Crear un issue para todo lo que tengamos en mente desarrollar en un futuro
- Todos los issues debidamente etiquetados y agrupados en sus correspondientes milestones
- CONTRIBUTING con indicaciones sobre cómo contribuir al proyecto
- Plantillas para Pull Request
- Cuando se rechaza un Pull Request indicar el motivo



Pues ahora vamos a poner  
en práctica todo lo que  
acabamos de aprender

¿PREGUNTAS?