



ULPGC
Universidad de
Las Palmas de
Gran Canaria

Iniciación a GitHub

Idafen Santana Pérez, PhD
Grupo de Tecnología Médica y
Audiovisual

04/03/2022

email

idafen.santana@ulpgc.es

Name

Last name

Work Spain

About me

Idafen Santana-Perez, PhD

- UPM
 - PhD 2016
 - Postdoc 2018
- ISTAC
 - Air Quality Data
- ULPGC, Research Fellow
 - GTMA, EELabs Project
- Lecturer: UPM, IE, ESCP, UNED



Grupo de Tecnología Médica y Audiovisual

ULPGC

- Grupo Propio ULPGC
 - Telecomunicaciones + Medicina
 - IUIBS
- Centro de simulación
- Procesamiento imagen médica
- **Realidad Virtual/Aumentada, Videojuegos, hologramas, procesamiento y visualización de datos**
- NAMIC (usando GitHub)
 - [https://www.na-mic.org/wiki/Project Events](https://www.na-mic.org/wiki/Project_Events)



Sistemas de entrenamiento

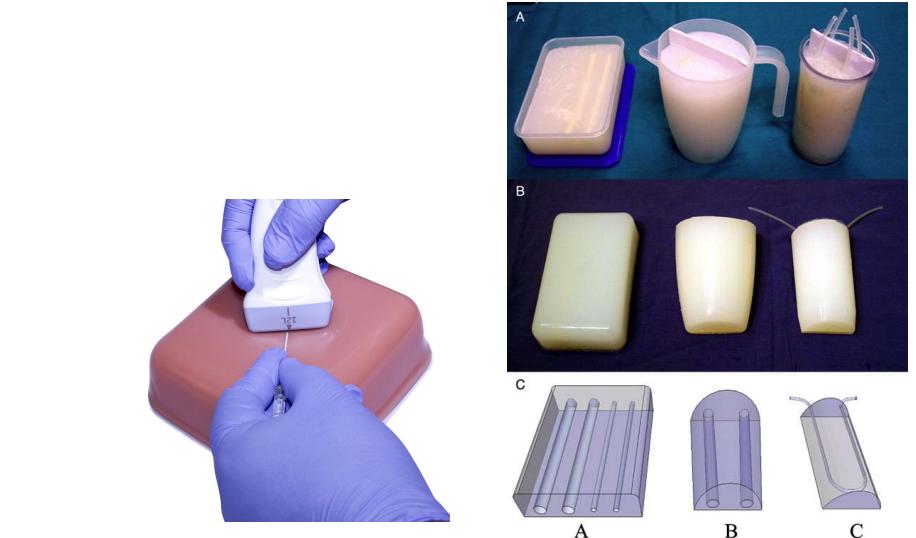
- Desarrollo de sistemas de entrenamiento para médicos residentes o estudiantes de medicina
 - Construcción de maniquís (fantomas)
 - Uso de nuevas tecnologías



Display holográfico



Dispositivo háptico



Fantomas de acceso vascular

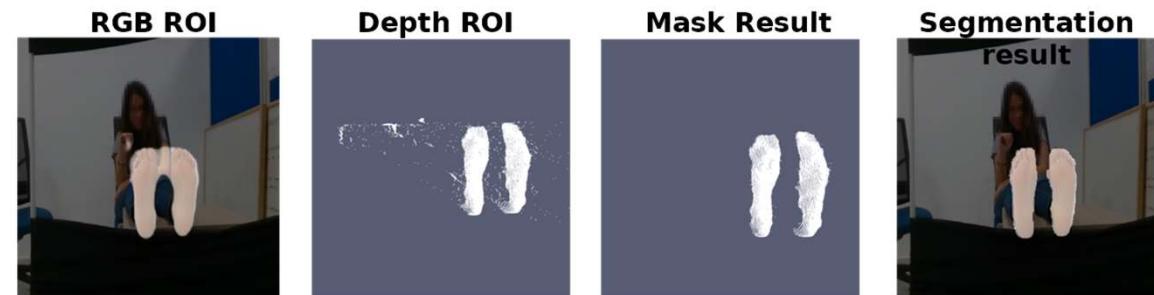
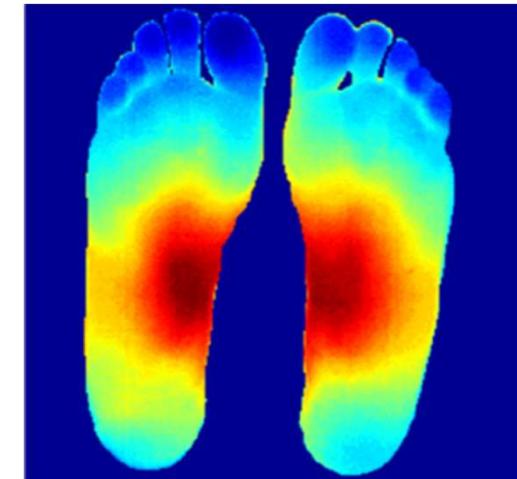
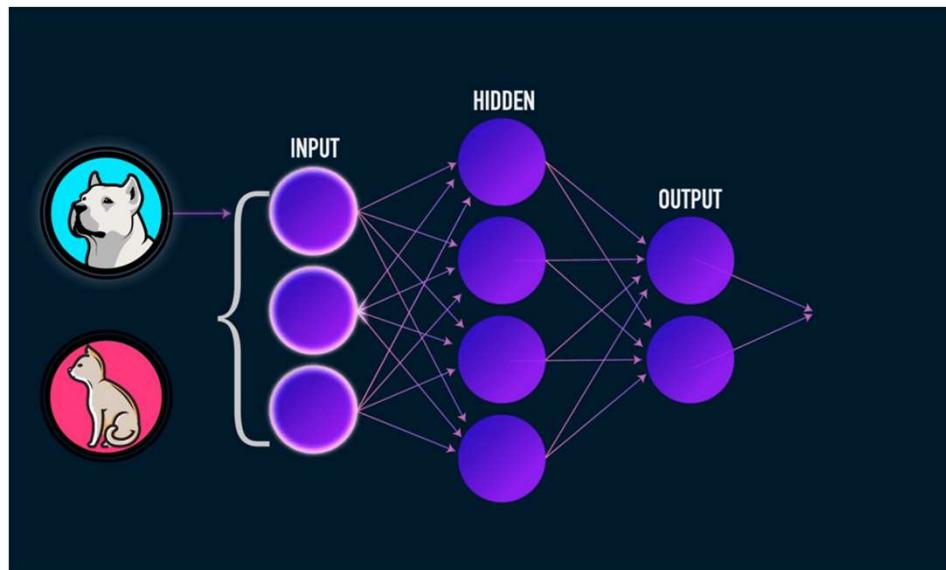


Realidad Virtual y Aumentada



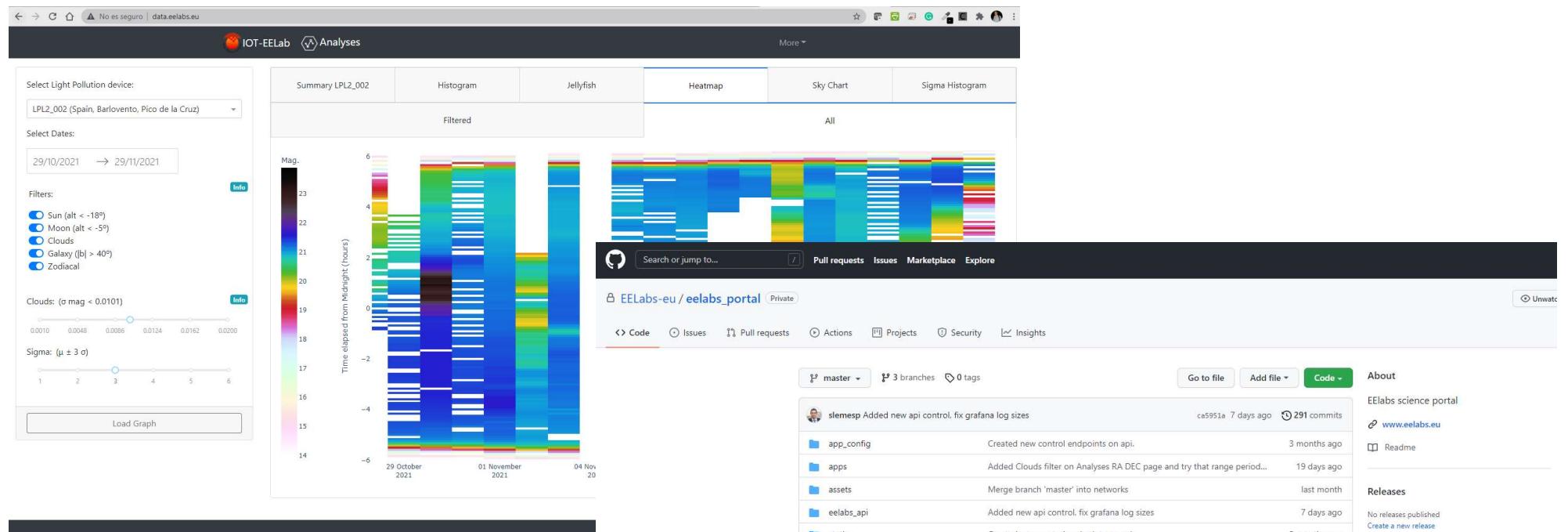
Procesamiento Imagen Médica

- Inteligencia Artificial
 - Deep Learning
 - Segmentación y clasificación de imágenes



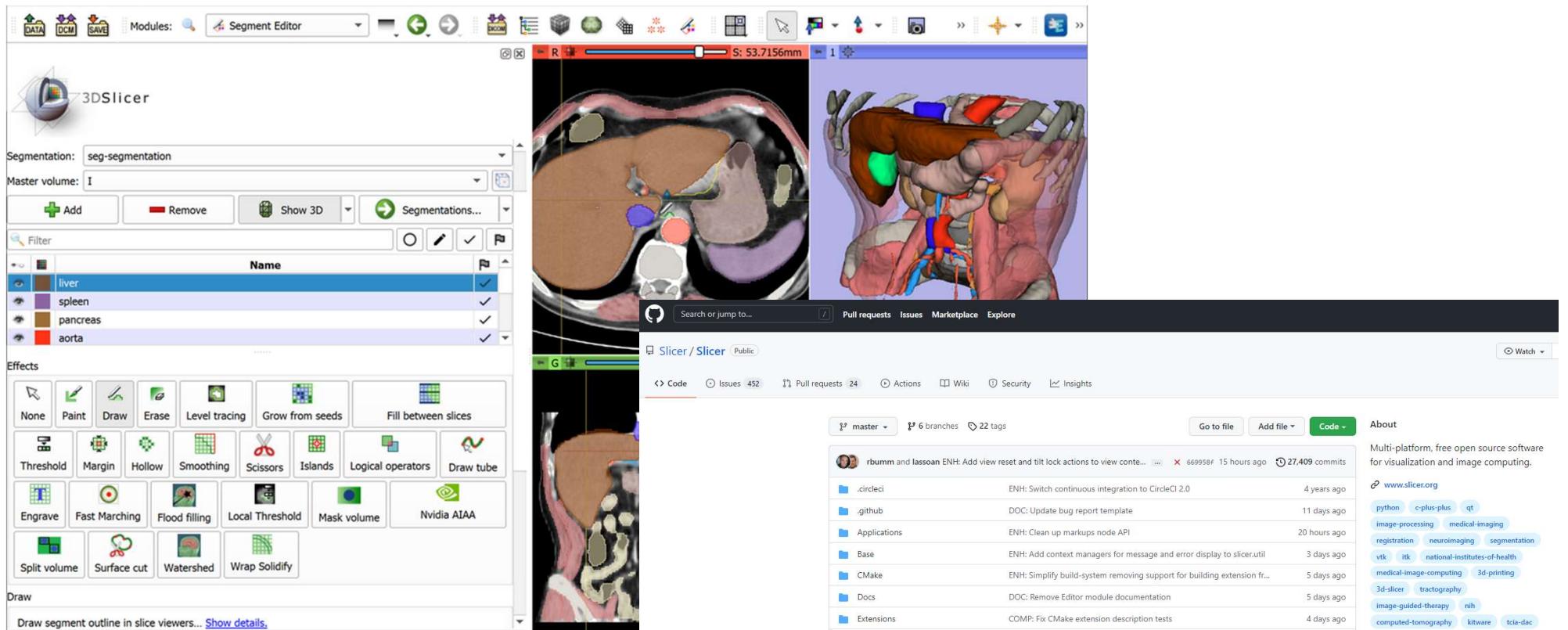
Grupo de Tecnología Médica y Audiovisual

ULPGC



Grupo de Tecnología Médica y Audiovisual

ULPGC



ESTRUCTURA

Iniciación a GitHub

Día 1

- Introducción al curso
- Git
 - Conceptos y aplicaciones
 - Repositorios
 - Commit
 - Revisión de historial



Iniciación a GitHub

Día 1

- Plataforma GitHub
 - Introducción a GitHub
 - Creación de cuenta GitHub
 - Creación de un repositorio
 - Notación markdown
 - Push & Pull
- Propuesta de ejercicio



Iniciación a GitHub

Objetivo Día 1

- Saber que es Git
 - Comandos básicos
 - Saber que es GitHub como plataforma
 - Conceptos
 - Repositorio
 - Edición y actualizar código en GitHub
- Práctico



Iniciación a GitHub

Día 2

- Corrección de ejercicios
- Creación de un Proyecto colaborativo
 - Ramas y Forks
 - Pull Requests y Merges
 - Gestión de conflictos en GitHub
 - Gestión de Issues
- Conclusiones y resumen del curso



Iniciación a GitHub

Objetivo Día 2

- Concepto de Branch
 - Branch vs Fork
 - Crear y cambiar de rama
 - Integrar código
- Concepto de Issue
 - Documentación y gestión de issues
- Práctico
 - Individual y colectivo



Materiales del curso

<https://idafensp.github.io/github/>

Mi usuario GH

GH pages

Directorio

Repositorio GH



¿PREGUNTAS?



¿GIT?
¿GITHUB?

The background features a minimalist design with abstract shapes. On the left side, there are three overlapping circles in shades of orange and yellow. The top circle is a light cream color, the middle is a bright orange, and the bottom is a darker orange. The right side of the image is a solid, vibrant orange color.

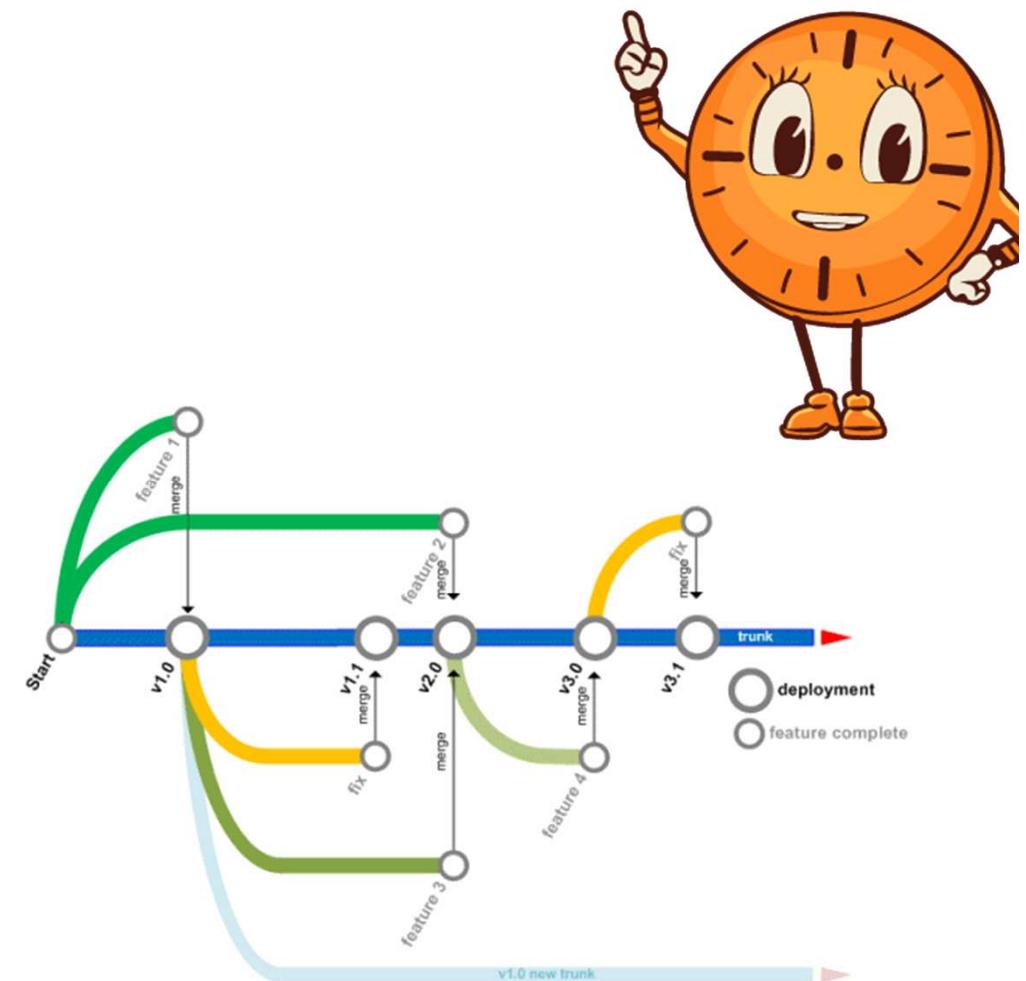
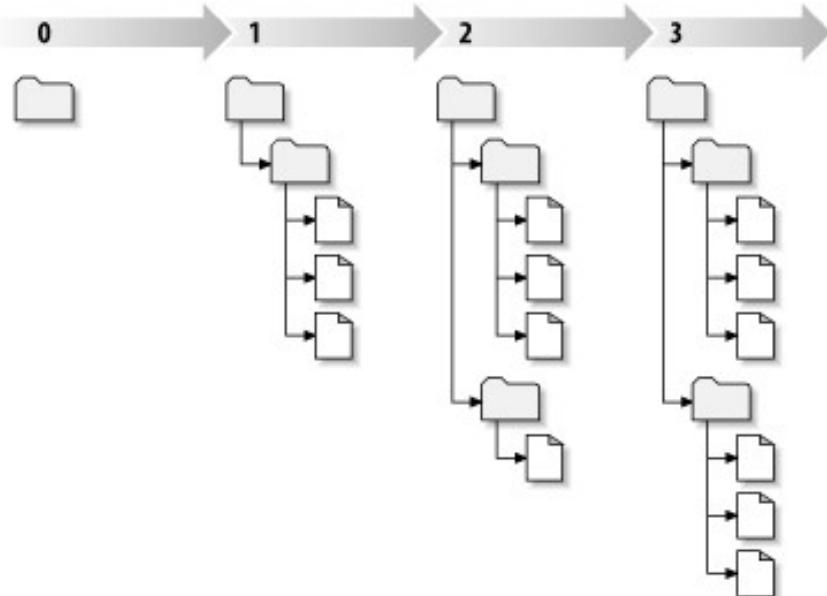
GIT

GIT es un Sistema de Control de Versiones

El **control de versiones** es un sistema que registra los **cambios** realizados sobre un archivo o conjunto de archivos a lo **largo del tiempo**, de forma que se puedan recuperar **versiones** anteriores en cualquier momento.



Control de versiones



Git

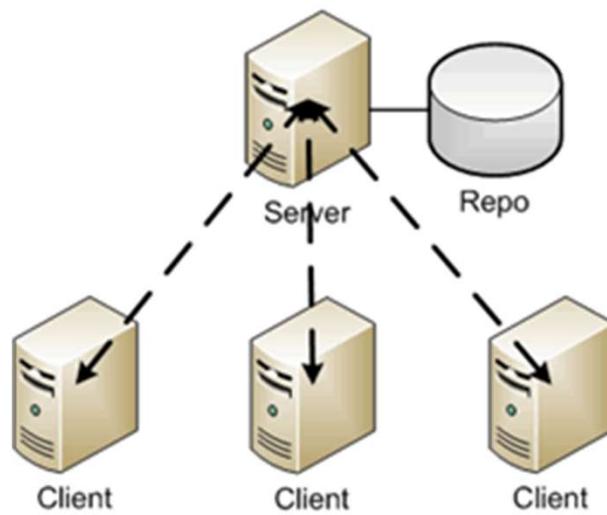
- <http://git-scm.com>
- Inicio en 2005, por Linus Torvalds y parte de la comunidad de Linux (reemplazo a BitKeeper).
- Una gestión distribuida de versiones.
- Herramienta de línea de comandos
- Cross-platform y Open Source
- Es **tremendamente rápido, muy eficiente con grandes proyectos, y tiene un increíble sistema de ramificación (branching)**



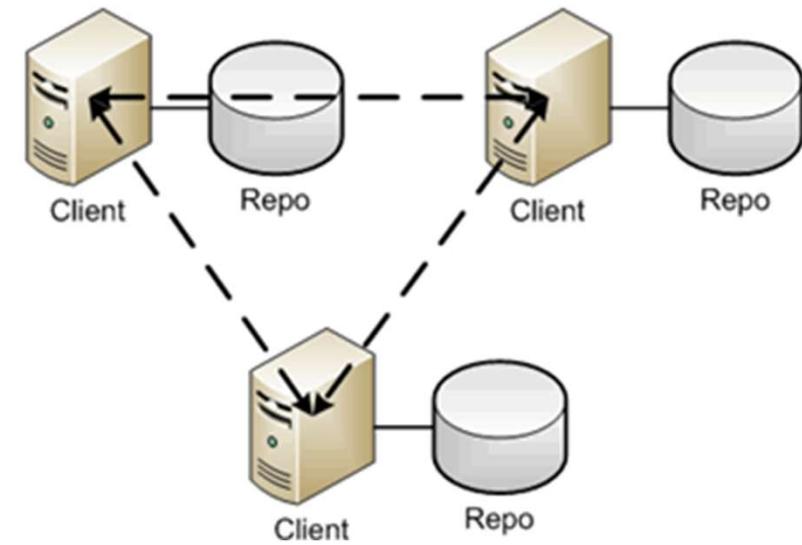
Git

- Sistema de control de versiones distribuido

Traditional



Distributed



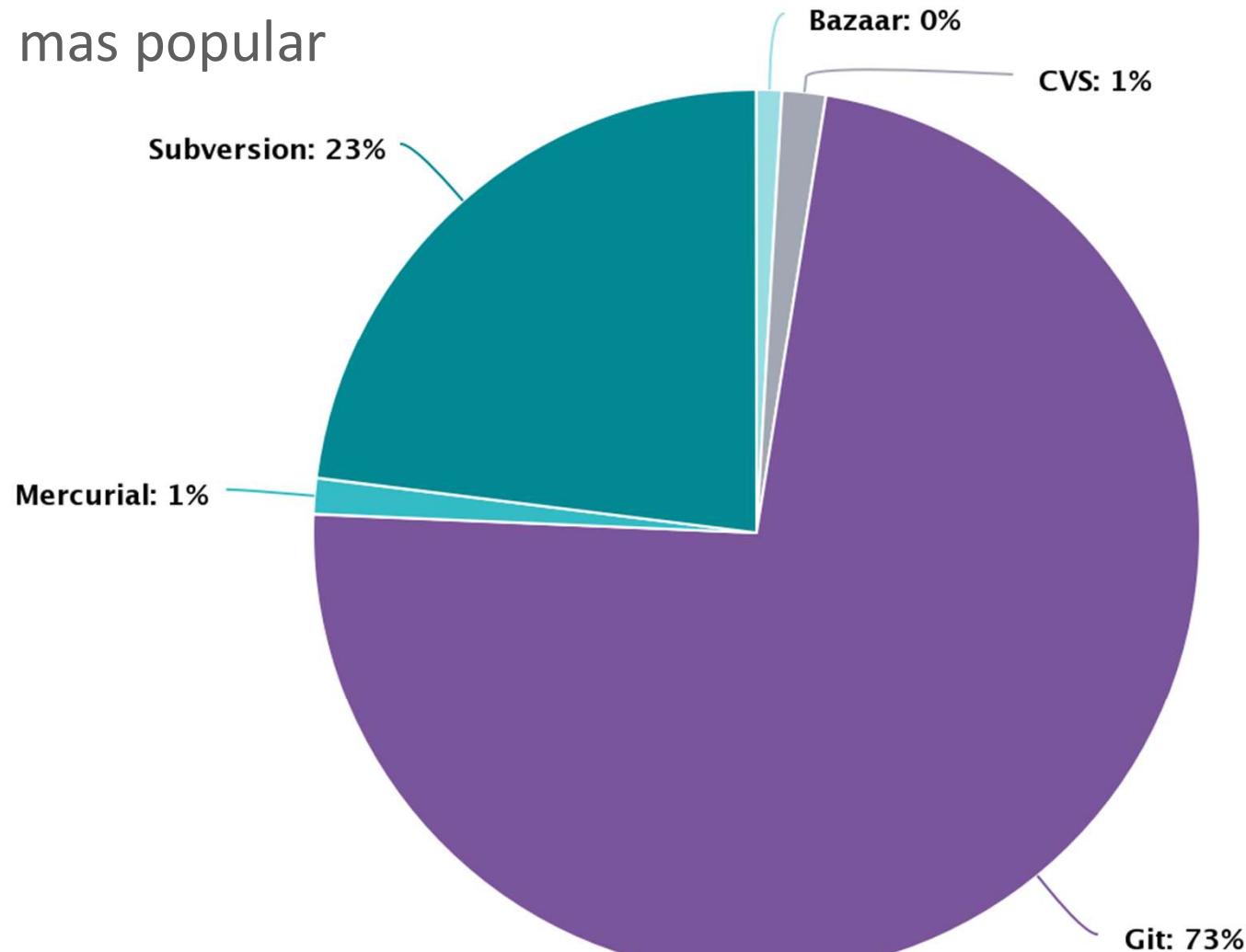
Git

- Sistema de control de versiones distribuido
 - Sin necesidad servidor central
 - Trabajo offline
 - Trabajo independiente y coordinación posterior
 - Todas las copias de un proyecto tienen el historial completo
 - **Git registra cambios, no versiones**



Git

- Sistema mas popular



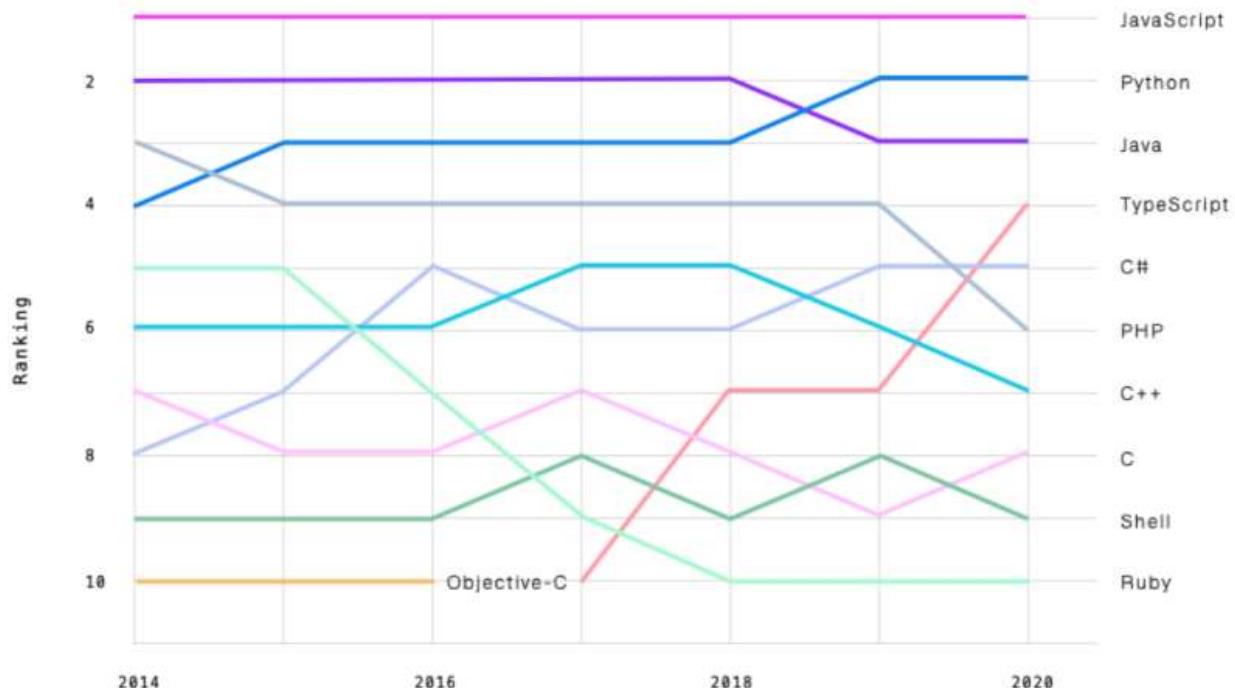
Source: <https://www.openhub.net/repositories/compare>



Git

¿Para que se usa?

- ¡Código!
 - JavaScript
 - Python
 - Java
 - Typescript
 - C#
 - PHP
 - C++
 - ...

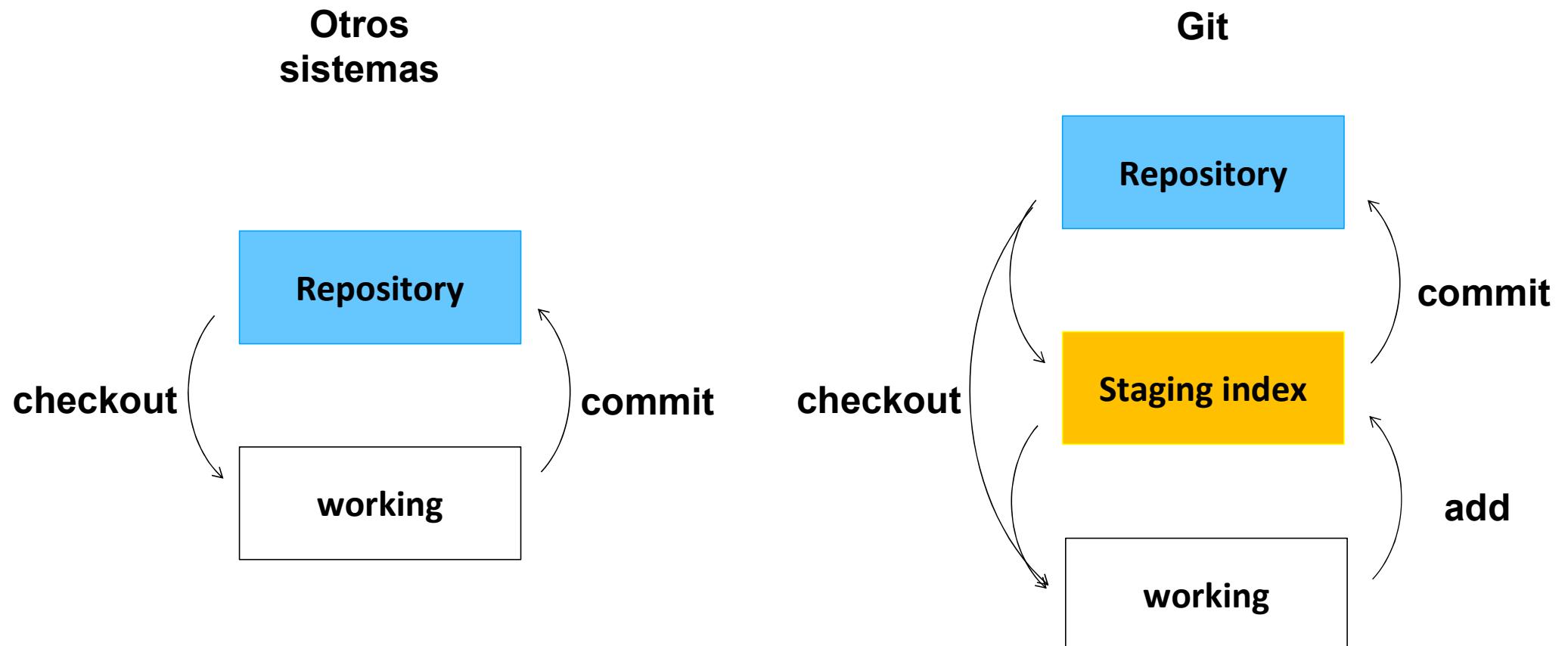


No es lo mas
idóneo para
imágenes, vídeos,
PDF, etc.



Git

Conceptos básicos



Git

Conceptos básicos

- **Working directory**

Solo puede haber una versión activa

- **Repository**

Contiene todas las versiones y las relaciones entre ellas.



Git

Conceptos básicos

- **HEAD**

Puntero a la versión con la que estamos trabajando. La que tenemos en nuestro directorio de trabajo.

- **Master / Main**

Es la rama principal sobre la que debe ir creciendo el desarrollo.

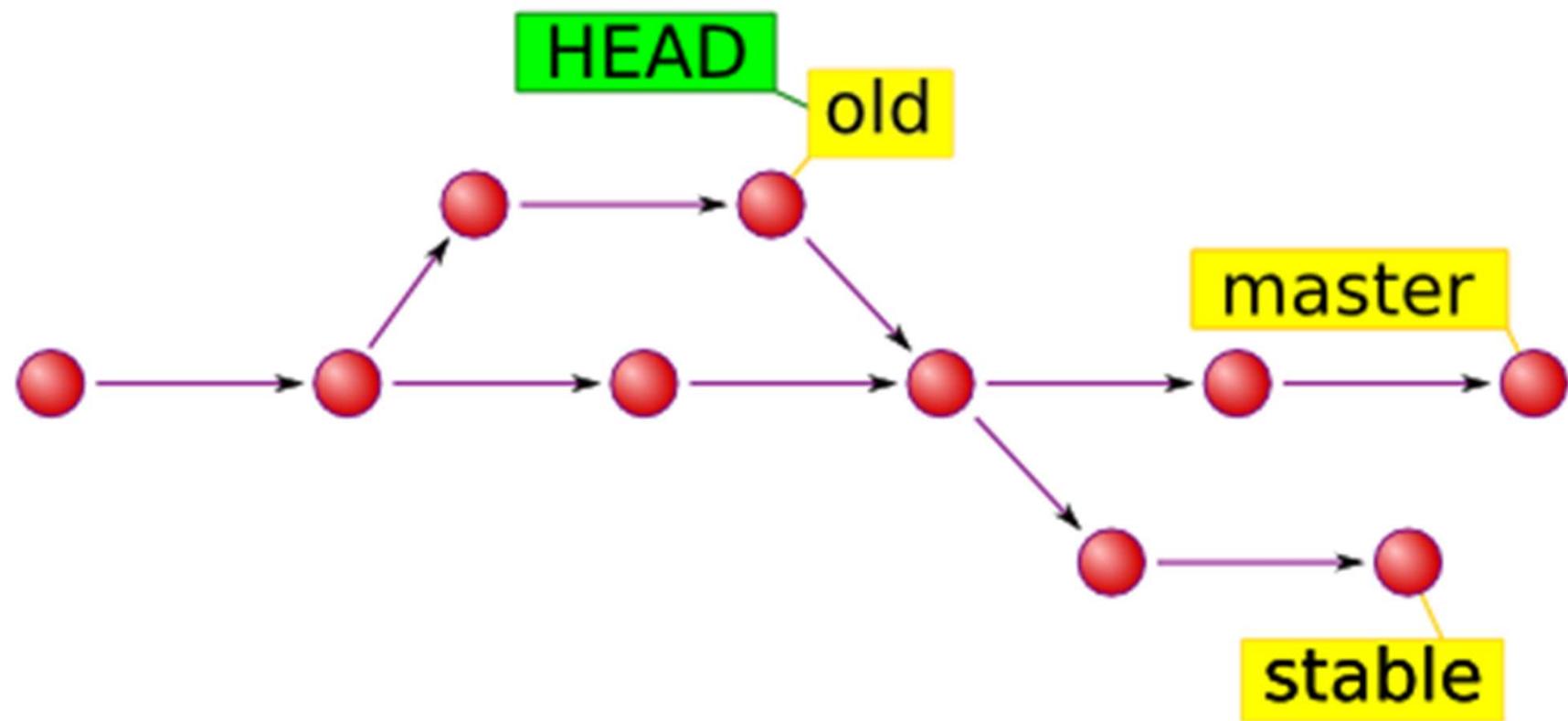
- **Branch**

Son ramas secundarias sobre las que se crean nuevas funcionalidades



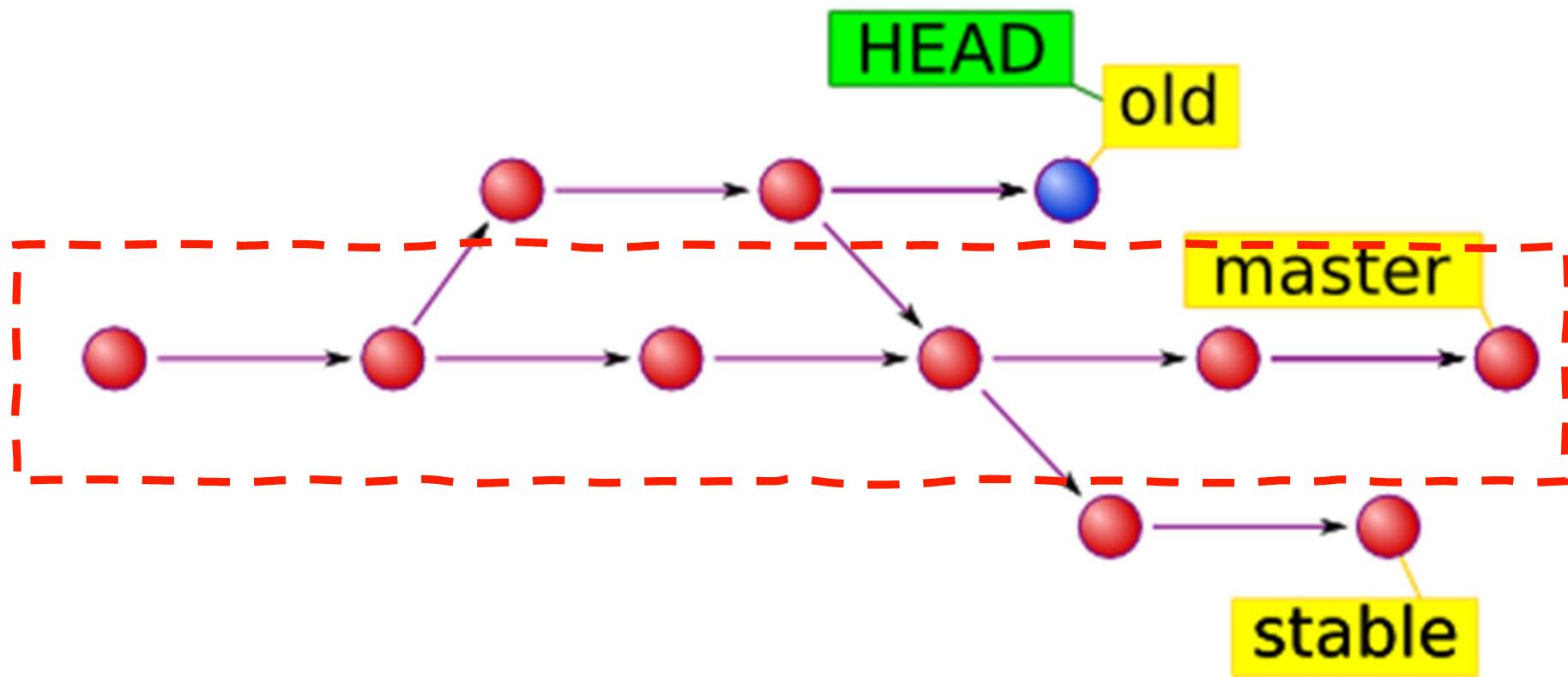
Git

HEAD y Ramas



Git

HEAD y Ramas

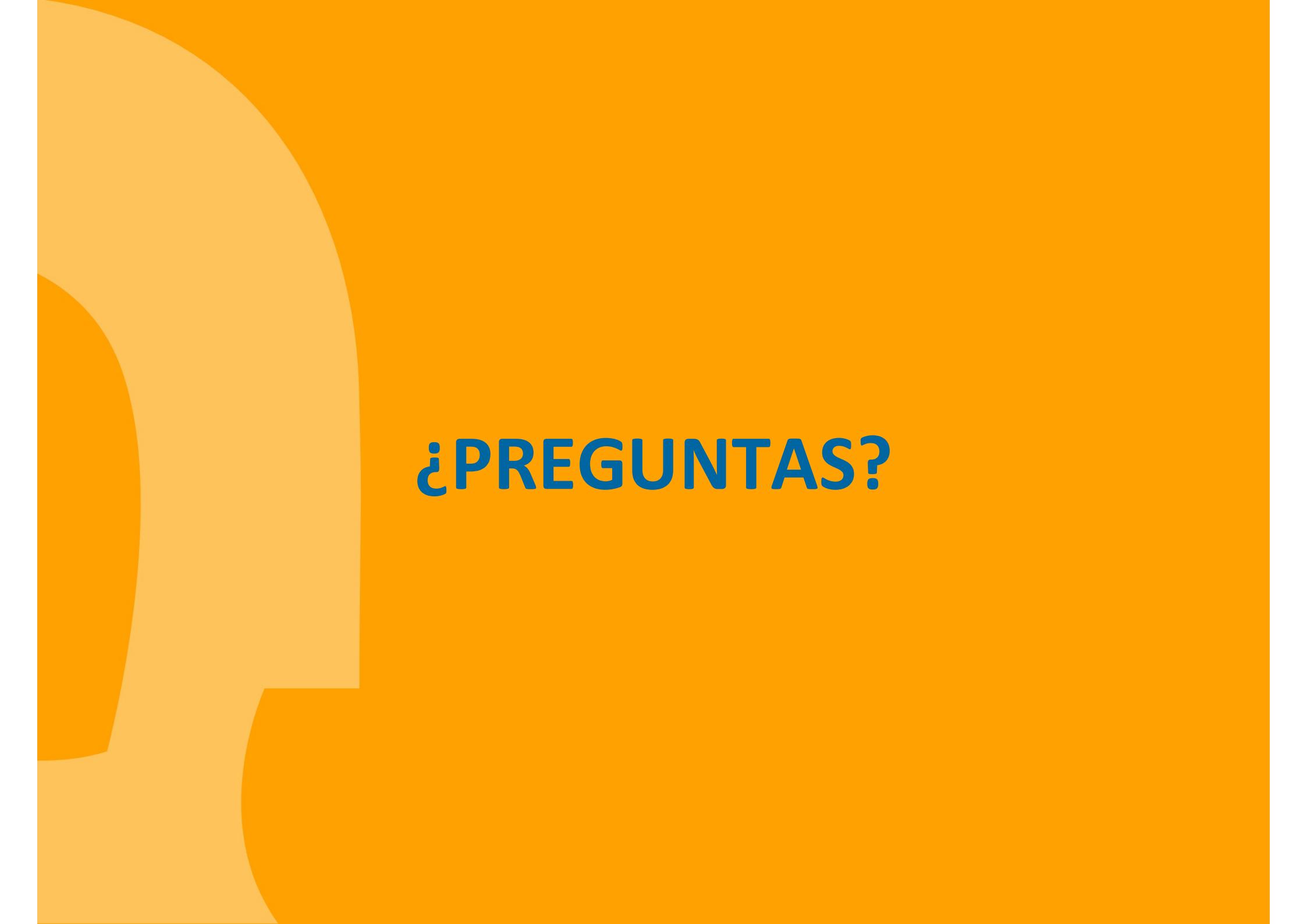


Git

Comandos mas usados (de 164 posibles)

- <https://git-scm.com/docs/git# git commands>
- ***git init***: iniciar un repositorio git
- ***git status***: comprobar el estado
- ***git log***: revisar historial de cambios
- ***git add***: añadir ficheros para un commit
- ***git commit***: confirmar un cambio
- ***git push***: subir cambios a remoto
- ***git pull***: recuperar cambios de remoto a local
- ***git rm***: eliminar fichero del sistema git





¿PREGUNTAS?



GITHUB

GitHub

- <https://github.com>



github.com/microsoft

 Microsoft
Open source projects and samples from Microsoft
📍 Redmond, WA | <https://opensource.microsoft.com> | [@OpenAtMicrosoft](#) | opensource@microsoft.com | Verified

[Overview](#) | [Repositories](#) 4.5k | [Packages](#) | [People](#) 4.4k | [Projects](#) 12 | [Sponsoring](#) 24

README.md

Open.
Collaborative.
Flexible.

Open Source enables Microsoft products and services to bring choice, technology and community to our customers.

Get Involved

Right now, open communities are building amazing software together, and there are excellent "good first issue" opportunities, if you're looking to get involved.

- [Explore featured projects](#)
- [Explore open source jobs at Microsoft](#)

Visit opensource.microsoft.com to learn more!

People



[View all](#)

Sponsoring



[View all](#)

Top languages

C# TypeScript Python JavaScript
C++

Most used topics

azure microsoft python

github.com/microsoft/vscode

main ▾ 469 branches 209 tags Go to file Add file ▾ Code ▾

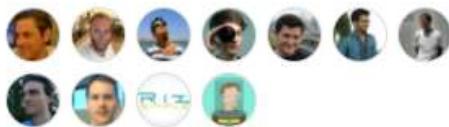
 sandy081 fix tests	94ab8f1 15 minutes ago	90,291 commits
.devcontainer	Minor edits	last month
.github	Run unit tests against node.js too (#137790)	7 days ago
.vscode	update endgame notebooks	2 days ago
build	build: 🚧 give names to the api proposal tasks	2 days ago
extensions	Fix regression related to git.scanRepositories	5 hours ago
remote	➡️ @parcel/watcher (fix #136460)	2 days ago
resources	➡️ web playground	9 days ago
scripts	tests - enable crash reporter for electron based unit tests	12 days ago
src	fix tests	15 minutes ago
test	smoke - ➡️ retry timeout	3 hours ago
.editorconfig	No forcing tabsizes on users	3 years ago
.eslintignore	Switch to dompurify for sanitizing markdown content (#131950)	3 months ago
.eslintrc.json	Copy and use IWindowDriver in PlaywrightDriver	19 days ago
.git-blame-ignore	Add organize to ignore list	4 months ago
.gitattributes	Whitelist comments in all JSON files. For #129206	2 months ago
.gitignore	Add server folder	last month
.lsifrc.json	Add lsifrc file	5 months ago
mailmap	Adds another of my email addresses	10 months ago

About Visual Studio Code code.visualstudio.com electron microsoft editor typescript visual-studio-code

Readme MIT License Code of conduct

Releases 54 [October 2021 Recovery 3](#) Latest 13 days ago + 53 releases

Used by 4 

Contributors 1,549 

GitHub

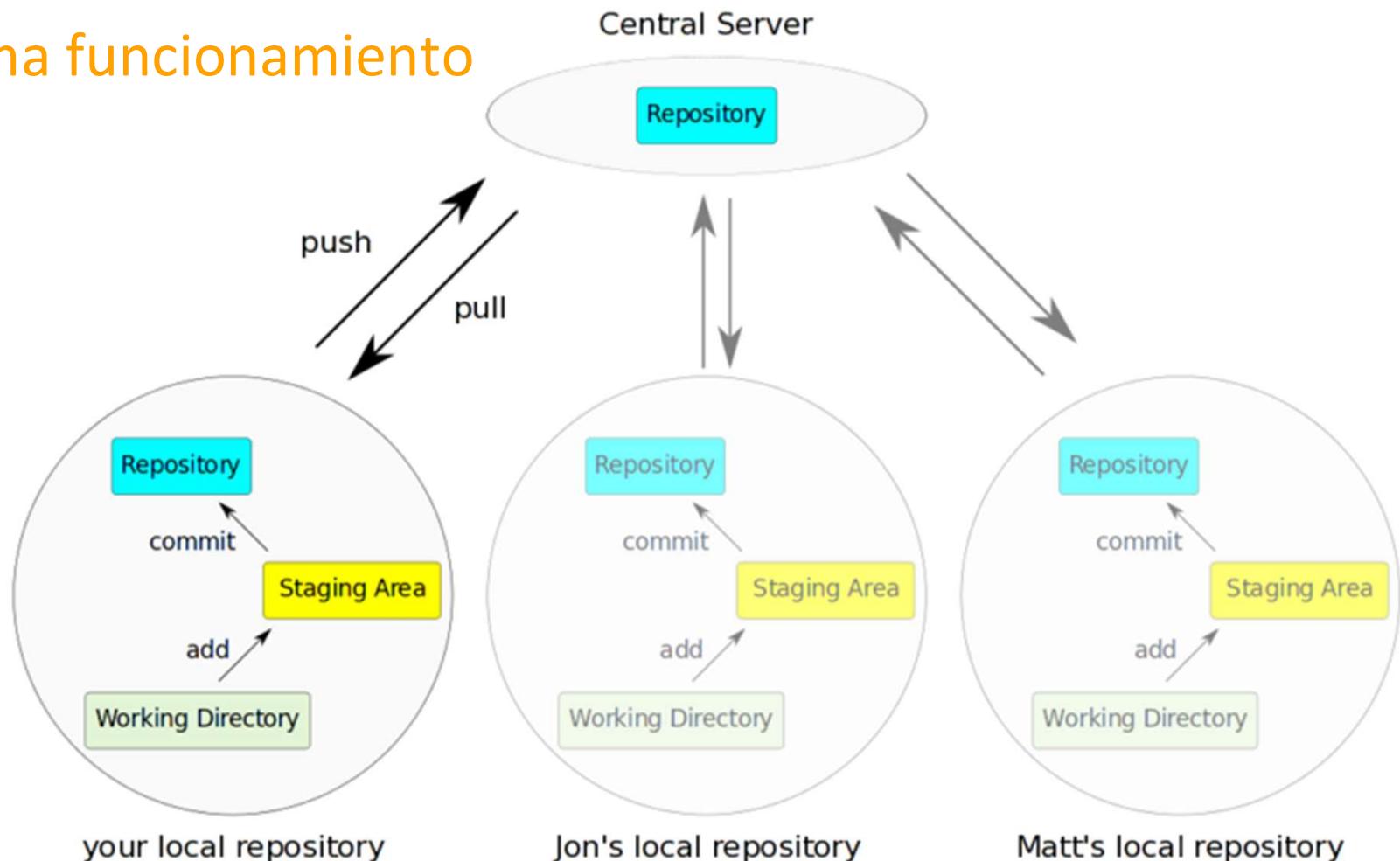
Historia

- <http://github.com>
- Desde 2008
- Plataforma online de gestión de repositories Git
- Proveedor mas popular de Git
- Git como “red social”
- Gratis y de pago
- Microsoft 2018 (\$7.500 mill.)



GitHub

Esquema funcionamiento



GitHub

Terminología

- <https://docs.github.com/es/get-started/quickstart/github-glossary>
- **Rama (Branch):** Una rama es una ***versión paralela de un repositorio.*** El repositorio la contiene, pero no afecta la rama primaria o principal, lo que te permite trabajar libremente sin interrumpir la versión "activa".
 - Una vez que haz hecho las modificaciones que deseabas, puedes fusionar tu rama de nuevo en la rama principal para publicar tus modificaciones.



GitHub

Terminología

- **Clon:** Un clon es una *copia de un repositorio que vive en tu equipo* en vez de en algún lugar del servidor del sitio web o el acto de realizar dicha copia.
 - Cuando haces un clon, puedes editar los archivos en tu editor preferido y usar Git para hacer un seguimiento de tus modificaciones sin tener que estar en línea.
 - El repositorio que clonas sigue conectado a la versión remota para que puedas subir tus modificaciones locales al remoto para mantenerlos sincronizados cuando estás en línea.



GitHub

Terminología

- **Commit (Confirmación):** Una *confirmación de cambios* o "revisión", es una modificación individual a un archivo (o conjunto de archivos).
 - Cuando realizas un commit para guardar tu trabajo, Git crea un ID único (también conocido como, "SHA" o "hash") que te permite mantener un registro de los cambios específicos confirmados junto con quién los realizó y cuándo.
 - Los commits generalmente contienen un mensaje de confirmación que es una descripción breve de las modificaciones que fueron realizadas.



GitHub

Terminología

- **Pull:** se refiere a cuando se *recuperan cambios y se fusionan*. Por ejemplo, si alguien ha editado el archivo remoto en el que ambos están trabajando, querrás extraer esos cambios para tu copia local para que esté actualizado. También consulta recuperar.
- **Fetch:** Cuando usas *git fetch*, estás *agregando cambios desde el repositorio remoto a tu rama de trabajo local sin confirmarlos*. A diferencia de *git pull*, la recuperación te permite revisar los cambios antes de confirmarlos en tu rama local.

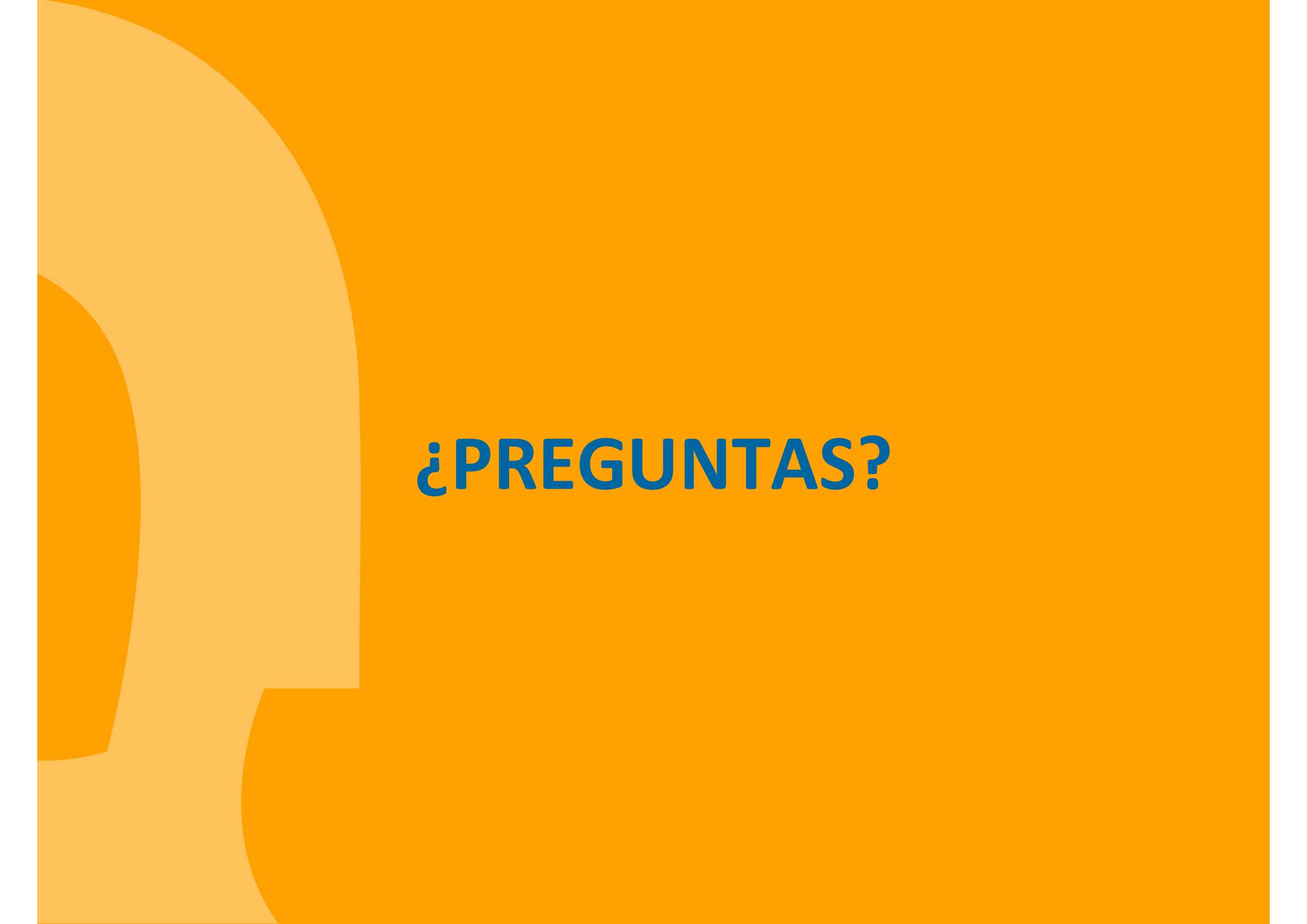


GitHub

Terminología

- Push: significa *enviar tus cambios confirmados a un repositorio remoto* en GitHub.com. Por ejemplo, si cambias algo de forma local, puedes subir esos cambios para que otros puedan acceder a ellos.
- Markdown: *Lenguaje de marcado* específico de GitHub usado para dar formato a la prosa y al código en GitHub.
 - Introducción a GitHub Markdown
 - <https://github.github.com/gfm/>





¿PREGUNTAS?

USUARIO GITHUB

GitHub

Cuenta de usuario

Choose the plan that's right for you.

How often do you want to pay?

Monthly Yearly

The screenshot shows the GitHub pricing page with three main plans: Free, Team, and Enterprise. The Team plan is highlighted with a blue border.

Plan	Cost	Collaboration	Deployment	Support
Free	\$0 per user/month	Individual and organization	No	Community support
Team	\$4 per user/month	Advanced collaboration for individuals and organizations	Yes	Everything included in Free, plus... (Access to GitHub Codespaces, Protected branches, Multiple reviewers in pull requests, Draft pull requests, Code owners, Required reviewers, Pages and Wikis)
Enterprise	\$21 per user/month	Security, compliance, and flexible deployment	Yes	Everything included in Team, plus... (Automatic security and version updates, SAML single sign-on, Advanced auditing, GitHub Connect, 50,000 automation minutes/month Free for public repositories, 50GB of Packages storage Free for public repositories)

Free
The basics for individuals and organizations
\$ 0 per user/month

Create a free organization

- Unlimited public/private repositories
- 2,000 automation minutes/month
Free for public repositories
- 500MB of Packages storage
Free for public repositories
- New Issues & Projects (in limited beta)
- Community support

MOST POPULAR

Team
Advanced collaboration for individuals and organizations
\$ 4 per user/month

Continue with Team ▾

- Everything included in Free, plus...
- Access to GitHub Codespaces
- Protected branches
- Multiple reviewers in pull requests
- Draft pull requests
- Code owners
- Required reviewers
- Pages and Wikis

Enterprise
Security, compliance, and flexible deployment
\$ 21 per user/month

Contact Sales **Start a free trial**

- Everything included in Team, plus...
- Automatic security and version updates
- SAML single sign-on
- Advanced auditing
- Github Connect
- 50,000 automation minutes/month
Free for public repositories
- 50GB of Packages storage
Free for public repositories



GitHub

Cuenta de usuario

- <https://docs.github.com/es/get-started/signing-up-for-github>
- Crear cuenta usuario
 - Ir a <https://github.com/signup>
 - Email y Password
 - Nombre de usuario
 - Rompecabezas
 - Verificar cuenta de correo
 - Opciones de equipo y Plan (free)





¿PREGUNTAS?

GitHub

Cuenta de usuario

Popular repositories

- ar2dtool**: Another RDF to diagram tool. Public. Java. ★ 14. 8 commits.
- NLP4Types**: HTML. ★ 1.
- WicusMontageIS**: Public.
- WicusSADemos**: Demos of the WICUS Infrastructure Specification Algorithm. Public. Java. ★ 1.
- WicusAnnotationTool**: Public.
- WicusWmsIS**: Infrastructure Specification Algorithm demo for REPPAR workshop. Public. Java.

120 contributions in the last year

Contribution activity

December 2021

Created 5 commits in 2 repositories
idafensp/idafensp.github.io 4 commits

Contribution settings ▾

Popular repositories

- ar2dtool**: Another RDF to diagram tool. Public. Java. ★ 14. 8 commits.
- NLP4Types**: HTML. ★ 1.
- WicusMontageIS**: Public.
- WicusSADemos**: Demos of the WICUS Infrastructure Specification Algorithm. Public. Java. ★ 1.
- WicusAnnotationTool**: Public.
- WicusWmsIS**: Infrastructure Specification Algorithm demo for REPPAR workshop. Public. Java.

120 contributions in the last year

Contribution activity

December 2021

Created 5 commits in 2 repositories
idafensp/idafensp.github.io 4 commits

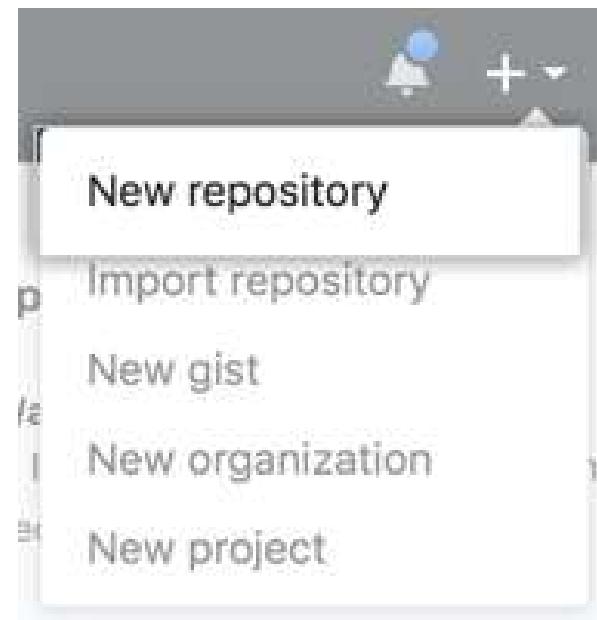


REPOSITORIO GITHUB

GitHub

Repositorio

- <https://docs.github.com/es/get-started/quickstart/create-a-repo>
- Vamos a crear un nuevo proyecto con nuestro usuario
 - Github.com
 - Iniciamos sesión
 - Nuevo repositorio





Repositorio

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *



idafensp

Repository name *

github-eii



Great repository names are short and memorable. Need inspiration? How about [super-robot?](#)

Description (optional)

Repositorio de ejemplo curso iniciación a Github EII 2021

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a [list of templates](#). [Learn more.](#)

.gitignore template: [Python](#) ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

License: [Apache License 2.0](#) ▾

[Create repository](#)



GitHub

Repositorio

- Add README
- Add gitignore
 - Python
- Choose a license
 - Apache License 2.0



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *



idafensp

Repository name *

github-eii



Great repository names are short and memorable. Need inspiration? How about [super-robot](#)?

Description (optional)

Repositorio de ejemplo curso iniciación a Github EII 2021

Public

Anyone on the internet can see this repository. You choose who can commit.

Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file

This is where you can write a long description for your project. [Learn more](#).

Add .gitignore

Choose which files not to track from a [list of templates](#). [Learn more](#).

.gitignore template: Python ▾

Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

License: Apache License 2.0 ▾

Create repository

GitHub

Repositorio

The image displays three identical screenshots of a GitHub repository page for the user 'idafensp' named 'github-eii'. The repository is described as 'Public'.

Header: The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. It also shows 1 branch, 0 tags, and 1 commit. On the right, there are buttons for Unwatch, Star (0), and Fork (0).

Code Section: This section lists the repository's contents. It shows a single commit by 'idafensp' titled 'Initial commit' made at '357ab6' 1 minute ago. The commit includes files: '.gitignore', 'LICENSE', and 'README.md', all of which are listed as 'Initial commit' made 1 minute ago.

README.md Content: The README file contains the following text:

```
github-eii
Repository de ejemplo curso iniciación a Github EII 2021
```

About Section: This section provides a brief description of the repository: 'Repositorio de ejemplo curso iniciación a Github EII 2021'. It includes a 'Readme' link and a 'Create a new release' button.

Releases Section: This section indicates 'No releases published' and 'Create a new release'.

Packages Section: This section indicates 'No packages published' and 'Publish your first package'.

Footer: The footer contains links for GitHub, Inc., Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About. It also features the same Unwatch, Star (0), and Fork (0) buttons.



GitHub

Repositorio

- Licencia

The screenshot shows two main sections of a GitHub repository page.

LICENSE Page:

- Header:** main · [github-eii / LICENSE](#) · Go to file · ...
- License Information:** idafensp/github-eii is licensed under the Apache License 2.0. A permissive license whose main conditions require preservation of copyright and license notices. Contributors provide an express grant of patent rights. Licensed works, modifications, and larger works may be distributed under different terms and without source code.
- Permissions:**
 - ✓ Commercial use
 - ✓ Modification
 - ✓ Distribution
 - ✓ Patent use
 - ✓ Private use
- Limitations:**
 - ✗ Trademark use
 - ✗ Liability
 - ✗ Warranty
- Conditions:**
 - ① License and copyright notice
 - ① State changes
- Note:** This is not legal advice. Learn more about repository licenses.

Repository Overview:

- Contributor:** idafensp Initial commit · Latest commit 357ab6 2 minutes ago · History
- Contributors:** 1 contributor
- File Details:** 201 lines (169 sloc) | 11.1 KB
- Code Preview:**

```

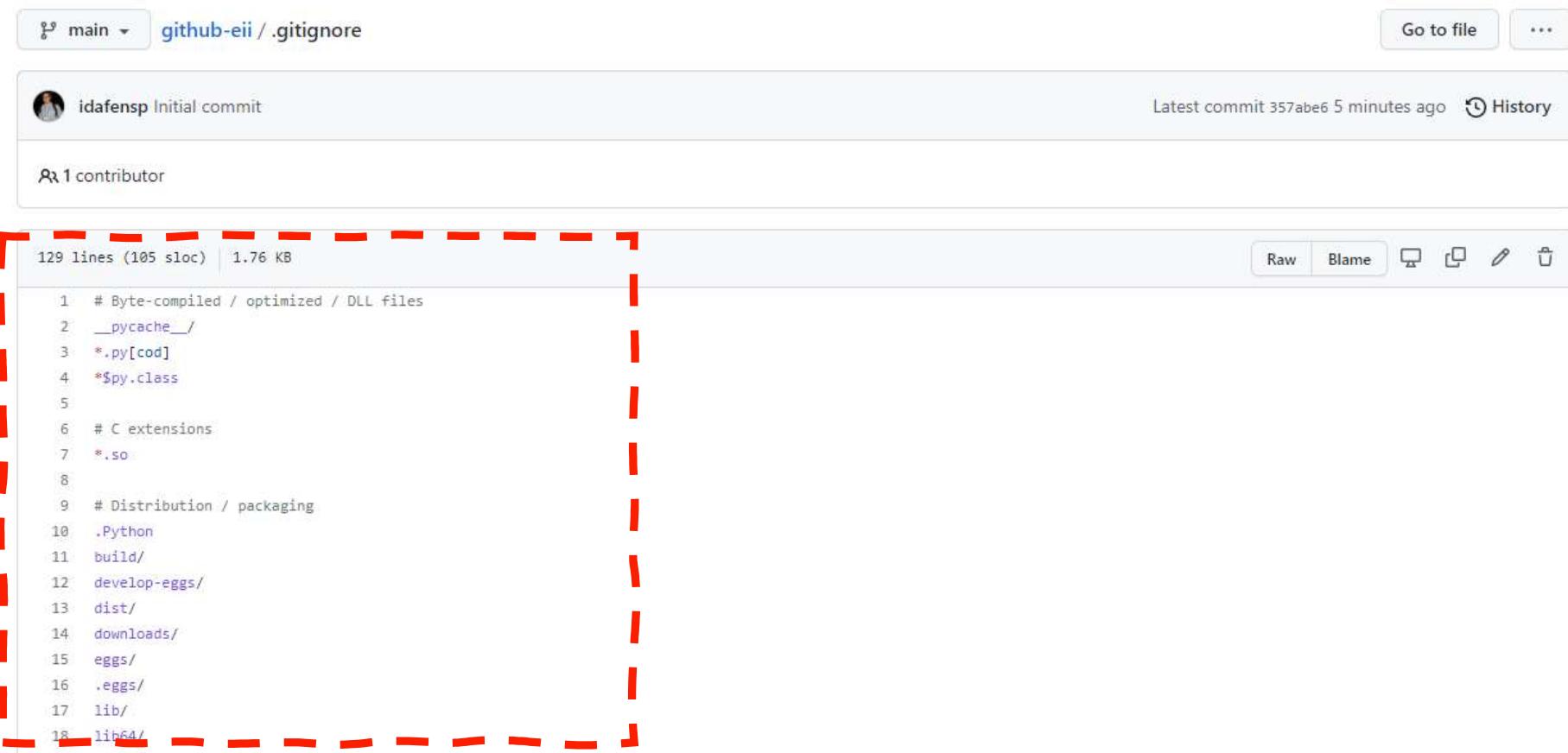
1      Apache License
2      Version 2.0, January 2004
3      http://www.apache.org/licenses/
4
5      TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION
6
7      1. Definitions.
8
9      "License" shall mean the terms and conditions for use, reproduction,
10     and distribution as defined by Sections 1 through 9 of this document.

```
- Actions:** Raw, Blame, View, Edit, Delete

GitHub

Repositorio

- .gitignore



The screenshot shows a GitHub repository page for 'github-eii' with the file '.gitignore' selected. The page includes a navigation bar with 'main' and 'github-eii / .gitignore', and buttons for 'Go to file' and '...'. Below the navigation is a commit history section with one commit by 'idafensp' labeled 'Initial commit'. It shows 1 contributor and a latest commit at 357abe6 5 minutes ago. A 'History' button is also present. The main content area displays the .gitignore file with 129 lines (105 sloc) and a size of 1.76 KB. The file content is as follows:

```
1 # Byte-compiled / optimized / DLL files
2 __pycache__/
3 *.py[cod]
4 *$py.class
5
6 # C extensions
7 *.so
8
9 # Distribution / packaging
10 .Python
11 build/
12 develop-eggs/
13 dist/
14 downloads/
15 eggs/
16 .eggs/
17 lib/
18 lib64/
```

At the bottom right of the code editor, there are buttons for 'Raw', 'Blame', and various file operations like copy, paste, and delete.

GitHub

Repositorio

- README

The screenshot shows a GitHub repository page for 'github-eii'. The top navigation bar includes a dropdown for branches ('main'), the repository name ('github-eii / README.md'), and buttons for 'Go to file' and '...'. Below the header, there's a commit history section showing one commit from 'idafensp' labeled 'Initial commit' and a 'Latest commit' from 357abe6 6 minutes ago. It also shows '1 contributor'. At the bottom, it displays '2 lines (2 sloc) | 72 Bytes' and a row of icons for comparison, raw view, blame, copy, and edit. A red circle highlights the edit icon (pencil). The main content area contains the text 'github-eii' and the description 'Repository de ejemplo curso iniciación a Github EII 2021'.

main ➔ [github-eii / README.md](#) Go to file ...

idafensp Initial commit Latest commit 357abe6 6 minutes ago History

1 contributor

2 lines (2 sloc) | 72 Bytes

Raw Blame

github-eii

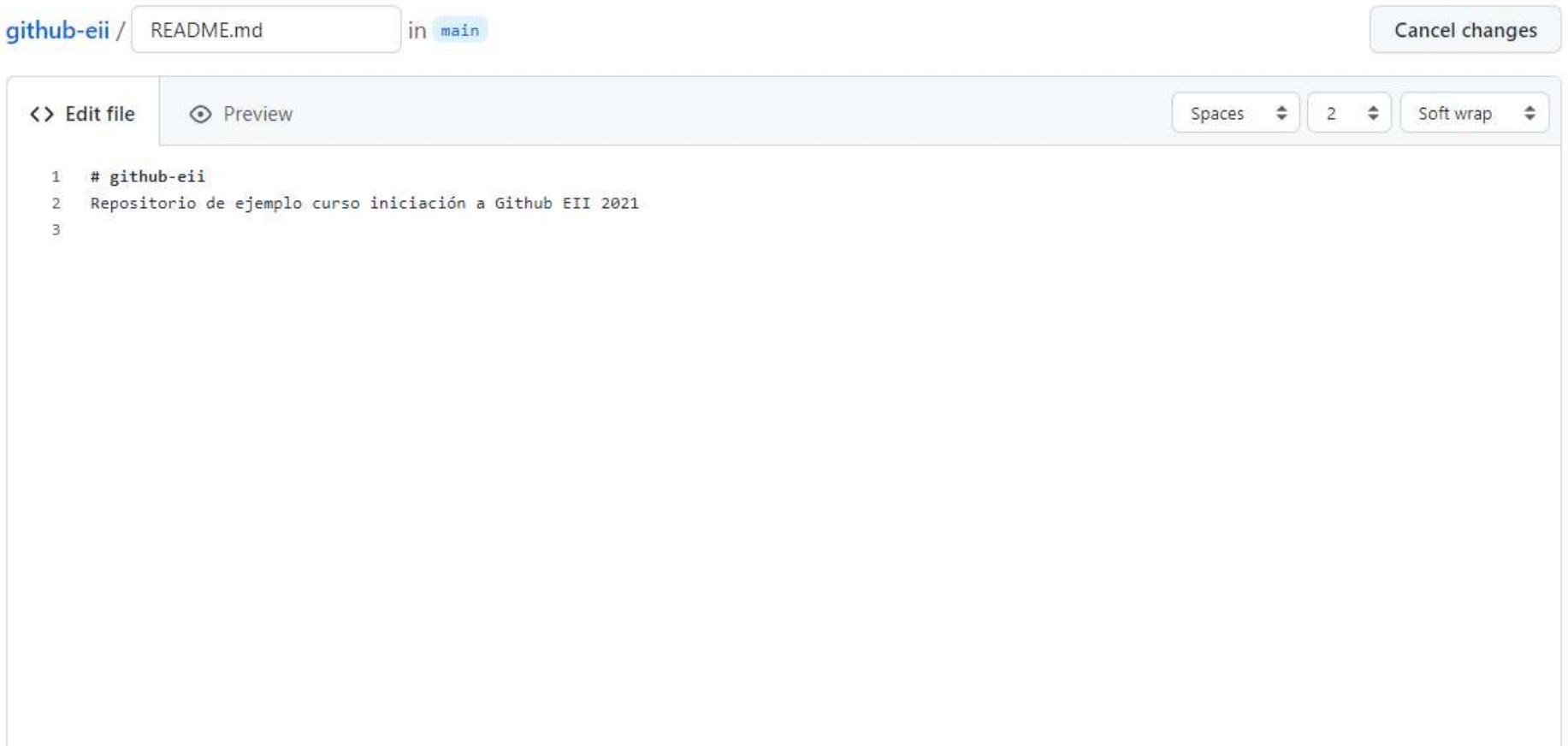
Repository de ejemplo curso iniciación a Github EII 2021



GitHub

Repositorio

- README



The screenshot shows the GitHub interface for editing a README file. At the top, it displays the repository path "github-eii / README.md" and the branch "in main". On the right, there is a "Cancel changes" button. Below the header, there are two tabs: "Edit file" (selected) and "Preview". To the right of these tabs are buttons for "Spaces" (dropdown), "2" (dropdown), and "Soft wrap" (dropdown). The main area contains the following text:

```
1 # github-eii
2 Repositorio de ejemplo curso iniciación a Github EII 2021
3
```



GitHub

Repositorio

- Markdown
 - <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>

 **Markdown Cheatsheet** 

A lightweight markup language with plain text formatting syntax.

Headers	Lists	Backslash Escapes	Images
<pre># This is Heading 1 - <h1> ## This is Heading 2 - <h2> ### This is Heading 3 - <h3> #### This is Heading 4 - <h4> ##### This is Heading 5 - <h5> ##### This is Heading 6 - <h6></pre>	<p>Unordered List</p> <ul style="list-style-type: none"> * Main Item 1 * Main Item 2 * Subitem 2a * Subitem 2b <p>Unordered Lists Can Be:</p> <ul style="list-style-type: none"> * Asterisks - Minus + Plus <p>Ordered List</p> <ol style="list-style-type: none"> 1. Main Item 1 2. Main Item 2 3. Main Item 3 3.1 Subitem 3a 3.2 Subitem 3b 	<pre>*literal asterisks*</pre> <ul style="list-style-type: none"> \backslash `backtick * asterisk _ underscore { curly braces [] square brackets () parentheses # hashmark + plus sign - minus sign (hyphen) . dot ! exclamation mark 	<pre>![Logo](/images/logo.png) ![Logo](/images/logo.png "Image Title") ![Logo](https://www.images.com/logo.png) ![Logo](https://www.images.com/logo.png "Title") ![Logo][image_logo]</pre> <p>[image_logo]: https://www.images.com/logo.png</p>
Emphasis	Task List	Code Blocks	Links
<pre>*This text have an italic font style* _This text have an italic font style_ **This text have an bold font style** __This text have an bold font style__ *italics **bold and italics** italics* **bold _bold and italic _bold**</pre> <p>~~Strikethrough this text~~</p>	<p>Horizontal Rules</p> <p>Three or more:</p> <pre>*** (asterisks) ____ (underscores) --- (hyphens)</pre> <p>Blockquotes</p> <p>This indicates that the enclosed text is an extended quotation and is rendered visually by indentation. (HTML <blockquote> Element)</p>	<pre>```java public class MyClass { } ``` Inline `code`.</pre>	<p>My homepage is at the following [link].</p> <pre>[URL of Code2Bits]: http://www.code2bits.com [:] https://www.code2bits.com [link]: https://www.code2bits.com</pre>
Horizontal Rules	Emoji	Tables	Tables
	<p>Emojis on GitHub: www.emoji-cheat-sheet.com</p> <pre>:+1: :sparkles: :camel: :tada: :rocket: :metal: :octocat:</pre>	<p>Header 1 Header 2 Header 3 ----- ----- ----- left-aligned centered right-aligned </p> <p>Header 1 Header 2 ----- ----- Content Content Content Content</p>	<p>Header 1 Header 2 Header 3 ----- ----- ----- left-aligned centered right-aligned </p> <p>Header 1 Header 2 ----- ----- Content Content Content Content</p>



Version 0.2 <https://www.code2bits.com>

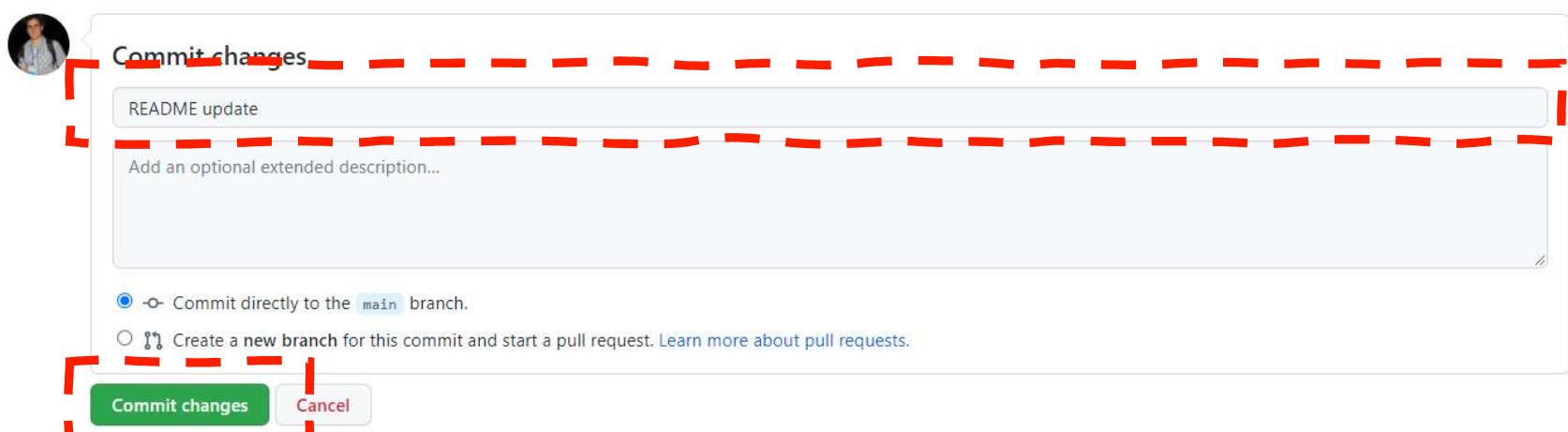
Source: <https://studylib.es/doc/8825178/cheatsheet-markdown>



GitHub

Repositorio

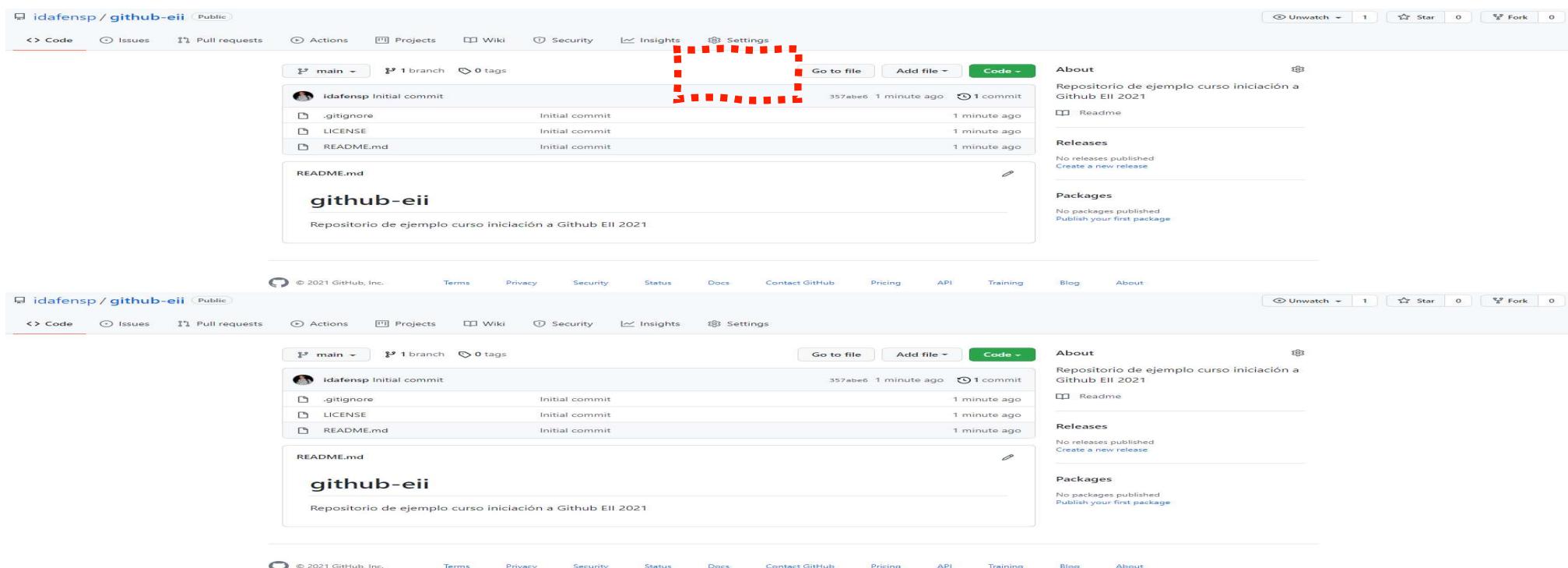
- README.md
 - En nuestro repositorio
 - Actualizamos, modificamos el texto como queramos
 - Hacemos un commit (rama principal)
 - Cada commit lleva un mensaje



TRABAJAR EN GITHUB

GitHub

Settings



The screenshot shows three identical views of a GitHub repository page for 'idafenesp / github-eii'. The top view has a red dashed box highlighting the 'Settings' tab in the top navigation bar. The middle and bottom views show the repository's code, commit history, README, and various GitHub features like About, Releases, Packages, and Insights.

Repository Overview:

- Code:** main branch, 1 branch, 0 tags
- Commits:** 357ab6, 1 minute ago, 1 commit
- Files:** .gitignore, LICENSE, README.md
- README.md:** **github-eii**, Repository de ejemplo curso iniciación a Github EII 2021

About: Repositorio de ejemplo curso iniciación a Github EII 2021

Releases: No releases published. Create a new release

Packages: No packages published. Publish your first package

Bottom Navigation: © 2021 GitHub, Inc., Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, About



GitHub

Settings

The screenshot shows the GitHub repository settings page for 'idafensp / github-eii'. The page has a light gray header with the repository name, a star count of 0, a fork count of 0, and a watch count of 1. Below the header is a navigation bar with links: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The Settings link is underlined, indicating it is the active tab.

A vertical sidebar on the left contains a list of settings options, each preceded by a red dashed border:

- Options
- Manage access
- Security & analysis
- Branches
- Webhooks
- Notifications
- Integrations
- Deploy keys
- Actions
- Environments
- Secrets
- Pages
- Moderation settings

Below the sidebar is an 'Edit' button.

The main content area is titled 'Settings' and includes the following sections:

- Repository name:** 'github-eii' with a 'Rename' button.
- Template repository:** A note explaining template repositories and a 'Learn more' link.
- Social preview:** A note about customizing the repository's social media preview and a 'Download template' link.
- Danger Zone:** A section containing four buttons with red dashed borders:
 - Change repository visibility:** A note stating 'For security reasons, you cannot change the visibility of a fork.' with a 'Change visibility' button.
 - Transfer ownership:** A note about transferring the repository to another user or organization with a 'Transfer' button.
 - Archive this repository:** A note about marking the repository as archived and read-only with an 'Archive this repository' button.
 - Delete this repository:** A note about deleting the repository with a 'Delete this repository' button.

GitHub

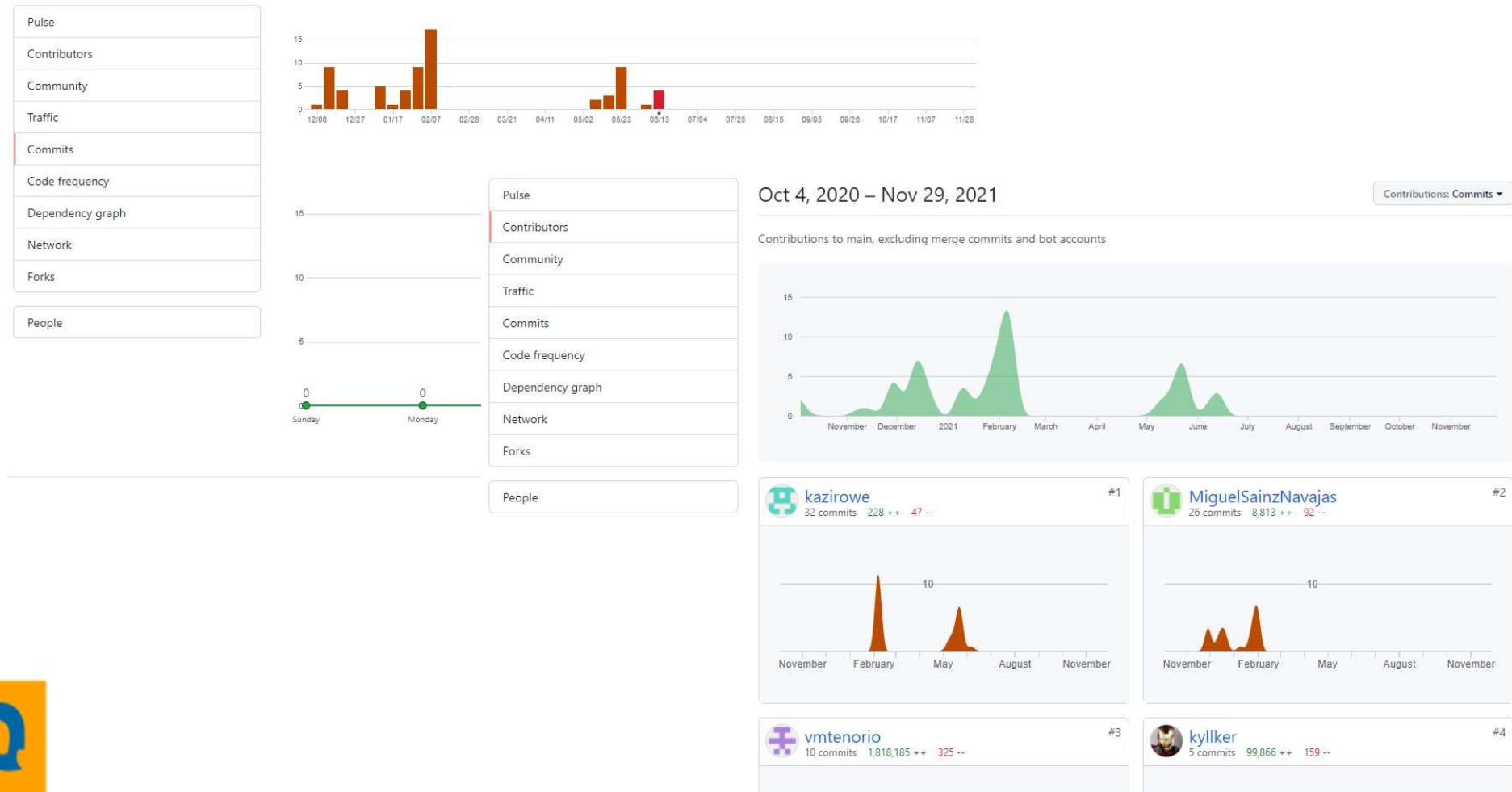
Insights

The image displays three identical screenshots of a GitHub repository page for 'idafenesp / github-eii'. The repository is public and contains one branch ('main'), one commit ('357ab6'), and three files: '.gitignore', 'LICENSE', and 'README.md'. The 'Code' tab is selected. On the right side of the page, the 'Insights' section is visible, featuring a red heatmap visualization. Below the heatmap, there is a summary of the commit: 'Initial commit' made by 'idafenesp' at '357ab6' 1 minute ago, with 1 commit. The 'About' section indicates it's a 'Repository de ejemplo curso iniciación a Github EII 2021'. The 'Releases' and 'Packages' sections show no activity.



GitHub

Insights





TRABAJAR EN LOCAL

GitHub

Línea de comandos

- GitHub es un repositorio online
- Git es un sistema basado en línea de commandos
 - <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>
- Windows
 - <https://gitforwindows.org/>



GitHub

Línea de comandos

- Personal Access Token (PAT)
 - <https://docs.github.com/es/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>
- Creamos un PAT
 - Para repositorios
 - Duración que creamos conveniente (> 2 semanas)
 - Copiamos el token y lo tememos a mano



Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your new personal access token now. You won't be able to see it again!

✓ ghp_IqIMNOZH6z0wIEB4T9A2g4EHMy8Ji42q4HA5

Enable SSO Delete

GitHub

Clonando un repositorio

- **Clon:** Un clon es una *copia de un repositorio que vive en tu equipo* en vez de en algún lugar del servidor del sitio web o el acto de realizar dicha copia.
 - Cuando haces un clon, puedes editar los archivos en tu editor preferido y usar Git para hacer un seguimiento de tus modificaciones sin tener que estar en línea.
 - El repositorio que clonas sigue conectado a la versión remota para que puedas subir tus modificaciones locales al remoto para mantenerlos sincronizados cuando estás en línea.





¿CLI?

Bash (CLI)

Comandos útiles

- **pwd** en que directorio estoy, muestra la ruta
- **cd** cambia el directorio
 - *cd gh-repo*
 - *cd .. (sube un directorio)*
- **ls** lista de elementos en un directorio
 - *ls -l (en formato lista)*
 - *ls -a (muestra los archivos ocultos)*



Bash (CLI)

Comandos útiles

- ***mv*** mover un fichero
 - *mv hello.py.txt hello.py* (renombrar un fichero)
- ***cat*** ver contenido de un fichero
 - *cat README.md*



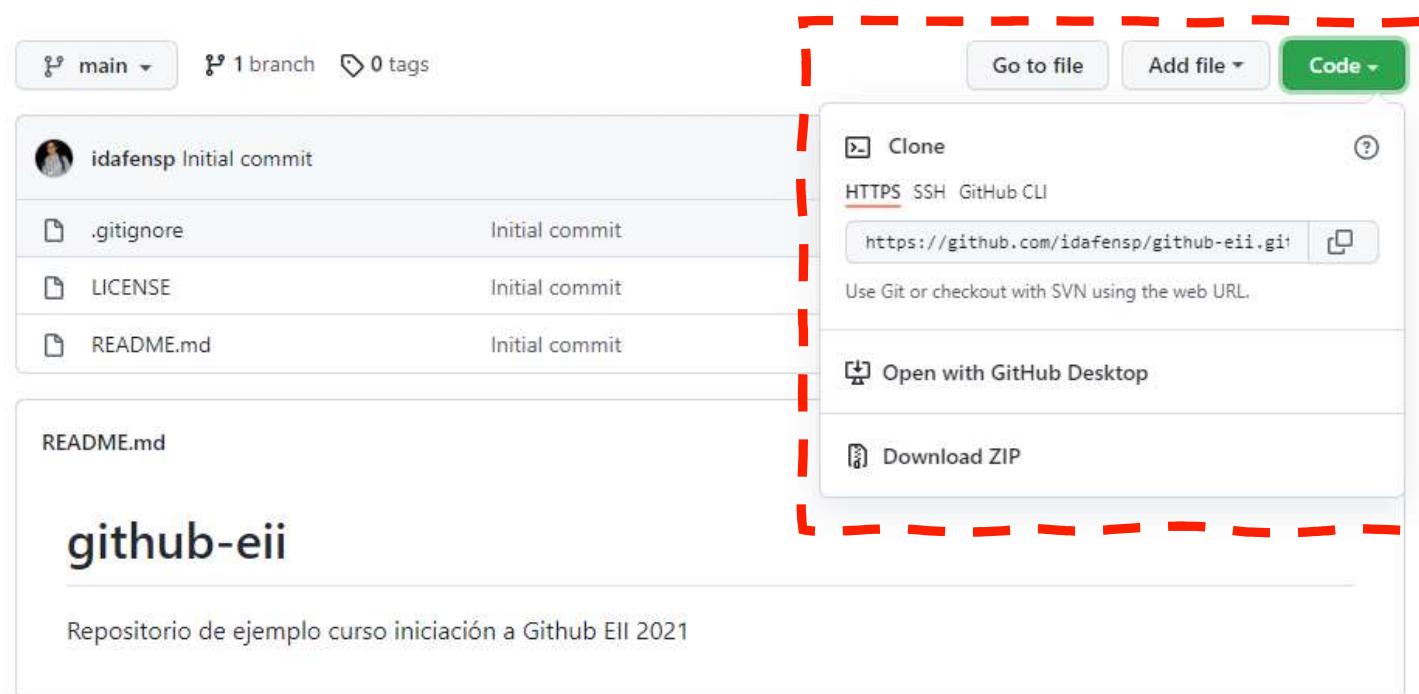


CLONANDO REPOS

GitHub

Clonando un repositorio

- *git clone <URL>*
- Usamos la URL git del repositorio que hemos creado



GitHub

Clonando un repositorio

- *git clone <https://github.com/idafensp/github-eii-2022.git>*

```
Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~
$ git clone https://github.com/idafensp/github-eii.git
Cloning into 'github-eii'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), done.
```

```
Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~
$ |
```



GitHub

Clonando un repositorio

- Crear PAT
- Abrir Gitbash
 - Elegimos el directorio de trabajo (e.g. *cd Desktop*)
 - *git clone <URL>*
 - Usamos la URL del repositorio que hemos creado
 - Revisamos los ficheros que se han descargado



```
$ git clone https://github.com/username/repo.git  
Username: your_username  
Password: your_token
```



GitHub

Ficheros del proyecto

- Ficheros ocultos

```
Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii (main)
$ ls -l
total 13
-rw-r--r-- 1 Idafen Santana Perez 197121 11558 Nov 29 12:48 LICENSE
-rw-r--r-- 1 Idafen Santana Perez 197121      74 Nov 29 12:48 README.md

Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii (main)
$ ls -la
total 65
drwxr-xr-x 1 Idafen Santana Perez 197121      0 Dec  2 10:10 .
drwxr-xr-x 1 Idafen Santana Perez 197121      0 Dec  1 13:01 ..
drwxr-xr-x 1 Idafen Santana Perez 197121      0 Nov 29 13:19 .git/
-rw-r--r-- 1 Idafen Santana Perez 197121    1928 Nov 29 12:48 .gitignore
-rw-r--r-- 1 Idafen Santana Perez 197121 11558 Nov 29 12:48 LICENSE
-rw-r--r-- 1 Idafen Santana Perez 197121      74 Nov 29 12:48 README.md
```



GitHub

Ficheros del proyecto

- .git

```
Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii (main)
$ cd .git

Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii/.git (GIT_DIR!)
$ ls -lath
total 24K
drwxr-xr-x 1 Idafen Santana Perez 197121  0 Nov 29 13:19 .
-rw-r--r-- 1 Idafen Santana Perez 197121 361 Nov 29 13:19 index
drwxr-xr-x 1 Idafen Santana Perez 197121  0 Nov 29 13:19 ..
-rw-r--r-- 1 Idafen Santana Perez 197121  41 Nov 29 13:19 ORIG_HEAD
-rw-r--r-- 1 Idafen Santana Perez 197121  98 Nov 29 13:19 FETCH_HEAD
drwxr-xr-x 1 Idafen Santana Perez 197121  0 Nov 29 13:19 objects/
-rw-r--r-- 1 Idafen Santana Perez 197121 20 Nov 29 13:15 COMMIT_EDITMSG
-rw-r--r-- 1 Idafen Santana Perez 197121 301 Nov 29 12:48 config
drwxr-xr-x 1 Idaten Santana Perez 197121  0 Nov 29 12:48 Togs/
-rw-r--r-- 1 Idafen Santana Perez 197121  21 Nov 29 12:48 HEAD
drwxr-xr-x 1 Idafen Santana Perez 197121  0 Nov 29 12:48 refs/
-rw-r--r-- 1 Idafen Santana Perez 197121 112 Nov 29 12:48 packed-refs
drwxr-xr-x 1 Idafen Santana Perez 197121  0 Nov 29 12:48 info/
drwxr-xr-x 1 Idafen Santana Perez 197121  0 Nov 29 12:48 hooks/
-rw-r--r-- 1 Idafen Santana Perez 197121  73 Nov 29 12:48 description
```



GitHub

Ficheros del proyecto

- .git/config

```
Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii/.git (GIT_DIR!)
$ cat config
[core]
    repositoryformatversion = 0
    filemode = false
    bare = false
    logallrefupdates = true
    symlinks = false
    ignorecase = true
[remote "origin"]
    url = https://github.com/idafensp/github-eii.git
    fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
    remote = origin
    merge = refs/heads/main
```

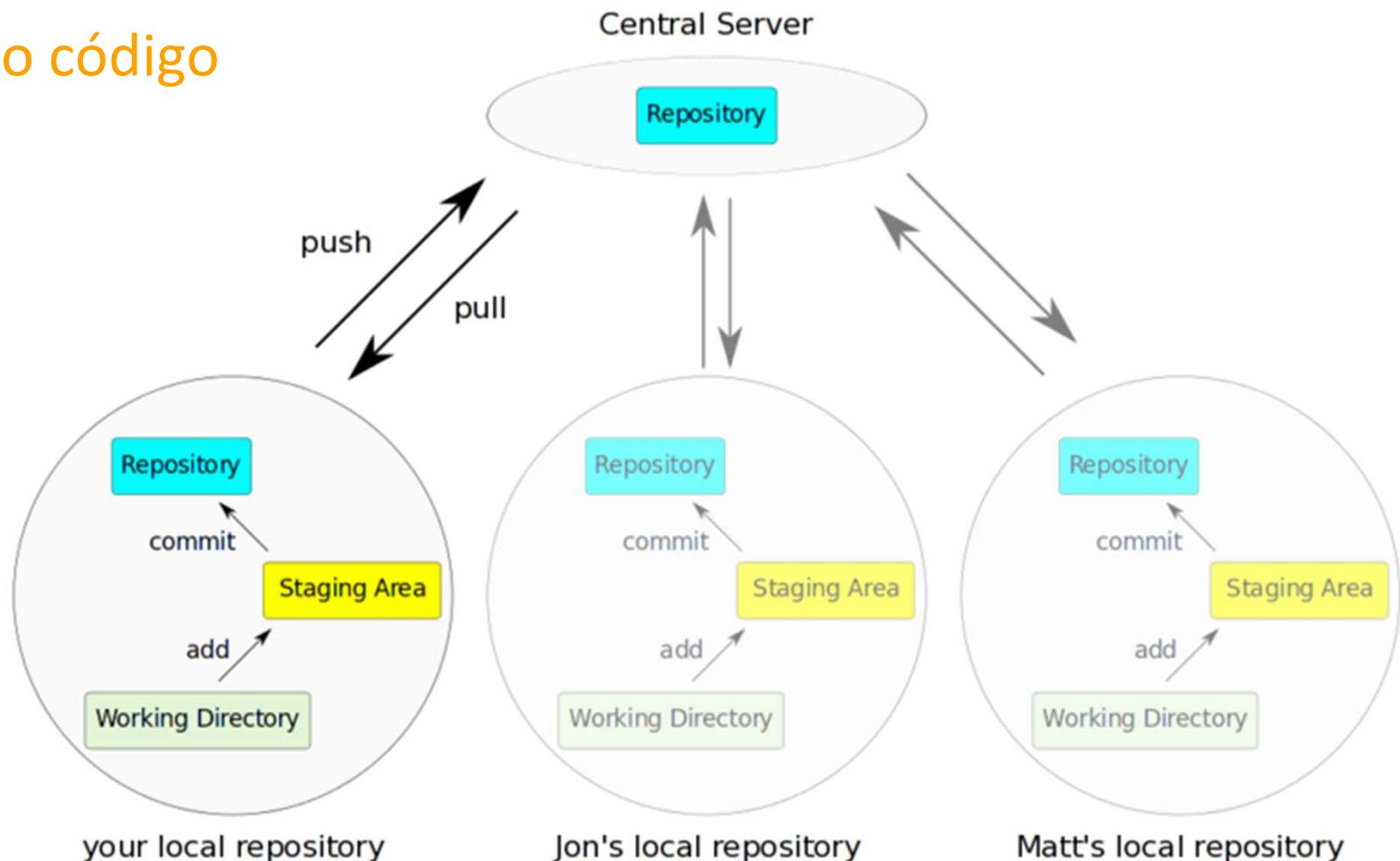




¿PREGUNTAS?

GitHub

Creando código



GitHub

Terminología

- <https://github.com/git-guides/git-add>
- Add (añadir): **agrega archivos nuevos o modificados en el directorio de trabajo** al área de preparación de Git.
 - *git add* es un comando importante; sin él, ningún *git commit* haría nada.
 - *git add <PATH>*
 - *git add **
 - *git add .*



GitHub

Terminología

- **Commit (Confirmación):** Una **confirmación de cambios** o "revisión", es una modificación individual a un archivo (o conjunto de archivos).
 - Cuando realizas un commit para guardar tu trabajo, Git crea un ID único (también conocido como, "SHA" o "hash") que te permite mantener un **registro de los cambios** específicos confirmados junto con quién los realizó y cuándo.
 - Los commits generalmente contienen un **mensaje de confirmación** que es una descripción breve de las modificaciones que fueron realizadas.



GitHub

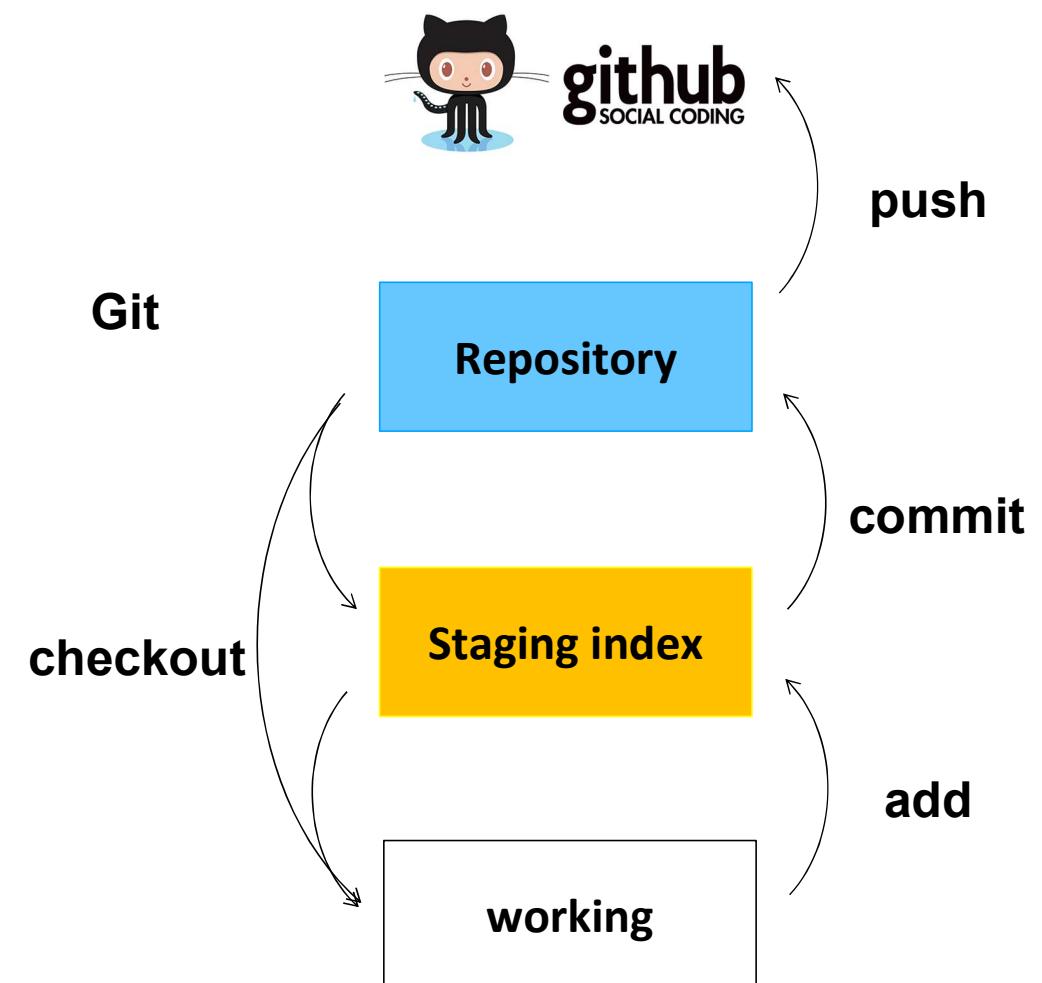
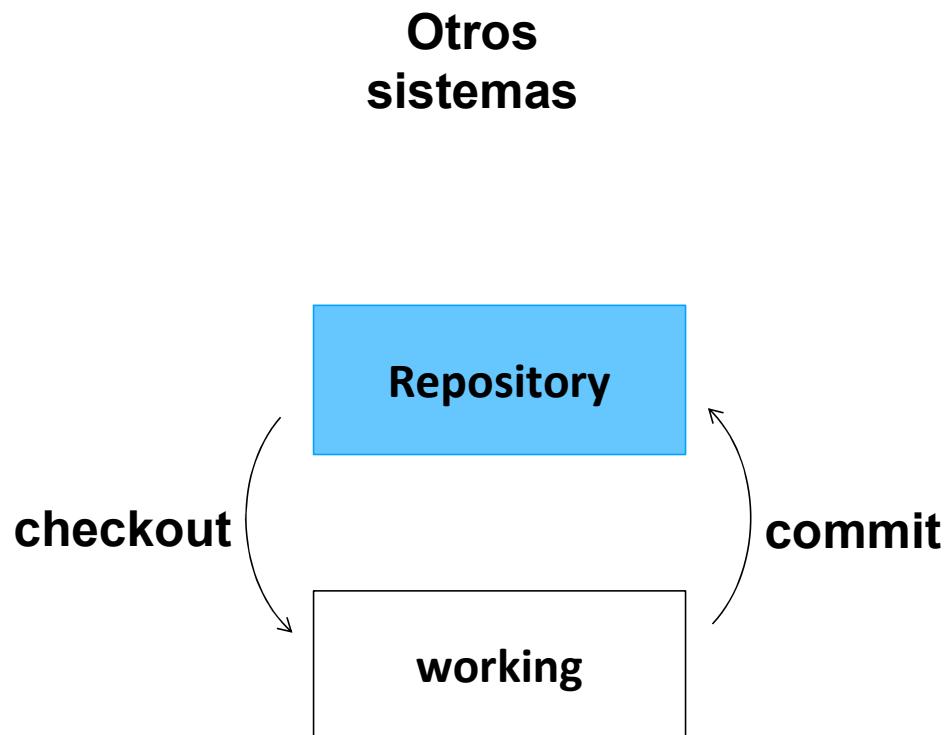
Terminología

- Push: significa *enviar tus cambios confirmados a un repositorio remoto* en GitHub.com. Por ejemplo, si cambias algo de forma local, puedes subir esos cambios para que otros puedan acceder a ellos.



Git

Conceptos básicos



GitHub

Creando código

- Vamos al directorio local del repositorio
- Creamos un nuevo fichero de texto
 - Lo nombramos “hello.py”
 - Lo editamos para añadir: `print("Hello world!")`
- Subimos los cambios a GitHub
 - `git add hello.py`
 - `git commit -m "Creamos el hello.py"`
 - `git push`
- Comprobamos los cambios en nuestro repo en GitHub (web)



Corto e
informativo





¿PREGUNTAS?

GitHub

Creando código

```
Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~
$ cd github-eii/

Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii (main)
$ git add hello.py

Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii (main)
$ git commit -m "Creamos el hello.py"
[main a4d5237] creamos el hello.py
 1 file changed, 1 insertion(+)
 create mode 100644 hello.py

Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 293 bytes | 293.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/idafensp/github-eii.git
 357abe6..a4d5237 main -> main
```





**ACTUALIZANDO
CÓDIGO**

GitHub

Terminología

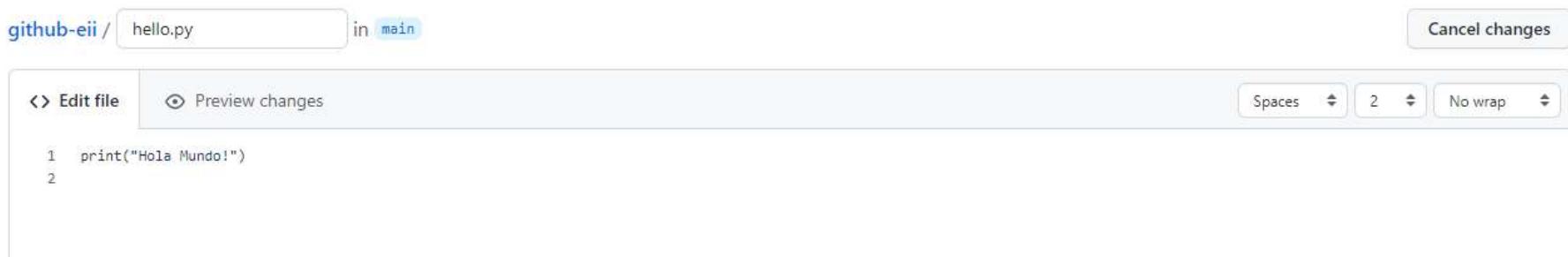
- **Pull:** se refiere a cuando se *recuperan cambios y se fusionan*. Por ejemplo, si alguien ha editado el archivo remoto en el que ambos están trabajando, querrás extraer esos cambios para tu copia local para que esté actualizado. También consulta recuperar.



GitHub

Actualizando código desde GitHub

- Ya estamos sincronizados con GitHub
 - Vamos a simular cambios de otro desarrollador
 - Vamos a nuestro repositorio en **GitHub (web)**
 - Editamos el documento hello.py
 - Cambiamos el mensaje a “Hola Mundo!”
 - Realizamos un commit (desde la propia web)



The screenshot shows a GitHub commit interface. At the top, it says "github-eii / hello.py in main". On the right, there are buttons for "Cancel changes" and "Edit file". Below that, there are buttons for "Edit file" and "Preview changes", along with settings for "Spaces", "2", and "No wrap". The code editor shows the following Python code:

```
1 print("Hola Mundo!")
```

In the bottom left corner of the slide, there is a small yellow square containing a blue stylized letter 'Q' logo.

GitHub

Actualizando código desde GitHub

- Vamos al directorio local del repositorio en Gitbash
- Actualizamos el código desde GitHub a nuestro working
 - *git pull*
- Comprobamos que se ha actualizado el contenido del fichero en local





¿PREGUNTAS?

GitHub

Comprobando el estado e historial



- Modificamos el fichero *hello.py*
 - Cambiamos el mensaje a lo que queramos
 - Añadimos el fichero: *git add hello.py* (*o git add **, *o git add .*)
 - Verificamos el estado ***git status***
 - Commit and Push
 - ***git log***
 - Comprobamos que los cambios que vemos corresponden con los que hemos realizado



GitHub

Comprobando el estado

```
Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii (main)
$ git add hello.py

Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii (main)
$ git status
on branch main
Your branch is up-to-date with 'origin/main'.
Changes to be committed:
  (use "git rese
    Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii (main)
Changes to be co$ git add hello.py
  (use "git rese
    Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii (main)
modified: $ git commit -m "Actualizamos mensaje"
[main 4fe3409] Actualizamos mensaje
  1 file changed, 1 insertion(+), 1 deletion(-)

    Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii (main)
$ git status
on branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```



GitHub

Comprobando el historial

```
Idafen Santana Perez@DESKTOP-P6T8IPR MINGW64 ~/github-eii (main)
$ git log
commit 7831f2ee6ce4a4fce3acc95b43a26a86e9be230b (HEAD -> main, origin/main, origin/HEAD)
Author: Idafen Santana Pérez <idafensp@gmail.com>
Date:   Mon Nov 29 13:19:04 2021 +0000

    Cambiamos a español

commit a4d5237fb5bef4c0eec34e2642ff82249e77ebf6
Author: unknown <idafensp@gmail.com>
Date:   Mon Nov 29 13:15:30 2021 +0000

    creamos el hello.py

commit 357abe627c9f98cd955b5373ad5ffb79bfa57f32
Author: Idafen Santana Pérez <idafensp@gmail.com>
Date:   Fri Nov 26 13:04:56 2021 +0000

    Initial commit
```





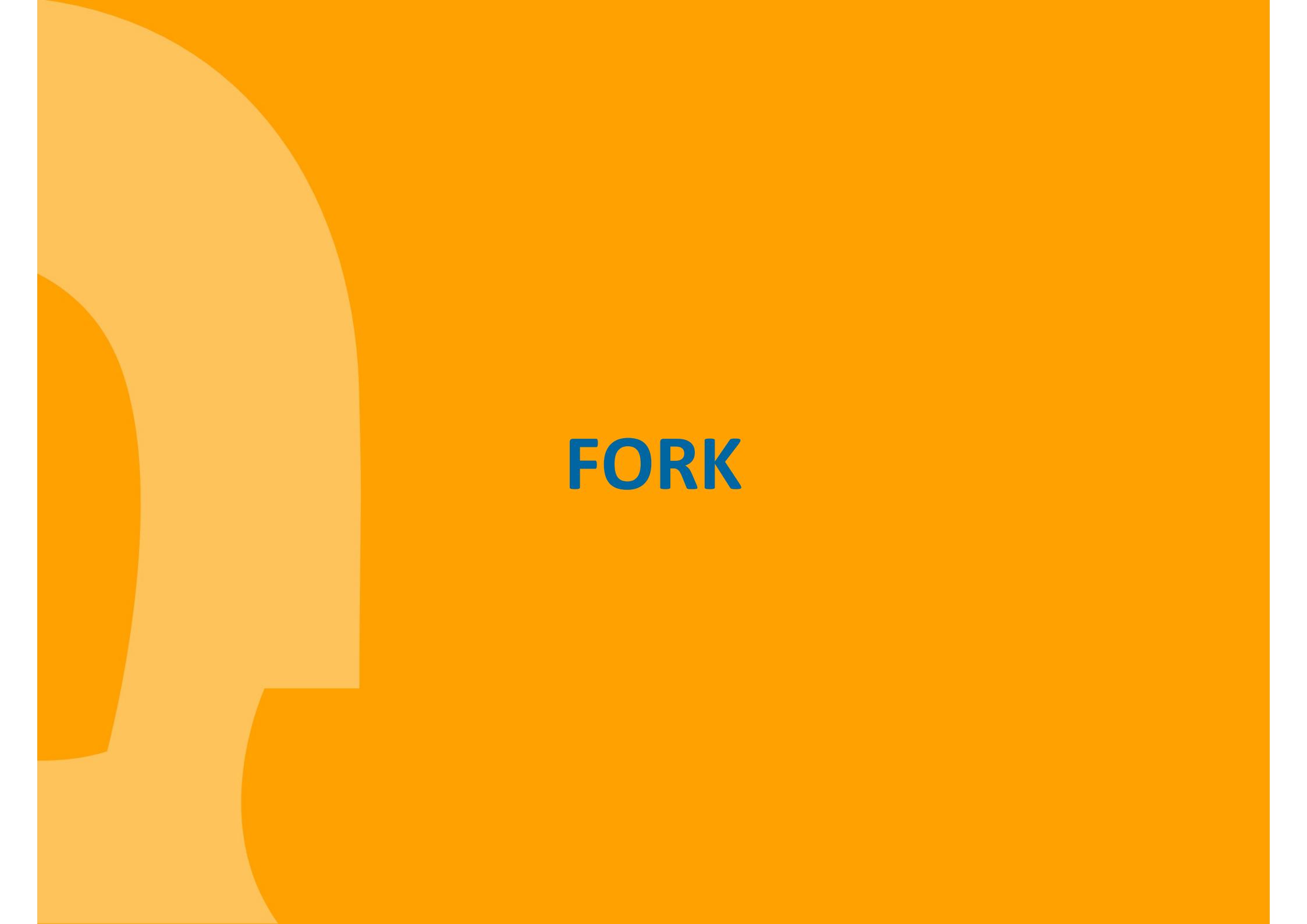
¿PREGUNTAS?

Iniciación a Github

Objetivo Día 1

- Saber que es Git
 - Comandos básicos
- Saber que es GitHub como plataforma
 - Cuenta en GitHub
 - Primer repositorio en GitHub
- Conceptos
 - Repositorio
 - Actualizar código en GitHub





FORK

GitHub

Usando un repositorio común

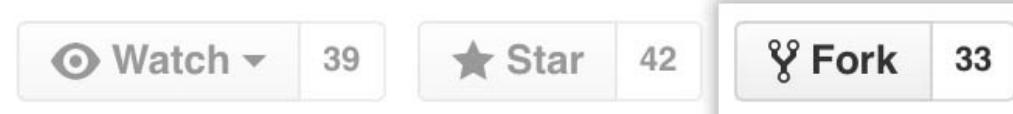
- <https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/working-with-forks/about-forks>
- Fork: es una copia de un repositorio. *Permiten realizar cambios en un proyecto sin afectar el repositorio original.*



GitHub

Usando un repositorio común

- Fork: vamos a crear una copia del repositorio
 - <https://docs.github.com/es/get-started/quickstart/fork-a-repo>
 - Vamos al repositorio
 - <https://github.com/idafensp/gh-eii-repos-2022>
 - Hacemos un fork



GitHub

Ejercicio final



- Fork del repo: <https://github.com/idafensp/gh-eii-repos-2022>
- Esto nos crea nuestra copia <https://github.com/<USUARIO>/gh-eii-repos-2022>
- Clone, add, commit, push (a tu fork)
 - O en la misma interfaz de Github
- Modificar el fichero lista.md (formato markdown)
 - Tu usuario Github y link a **tu repo** (en una línea)
 - [corsario85](<https://github.com/corsario85/prueba-gh-eii>)
- Hacemos un ***pull request***



GitHub

Ejercicio final



Two screenshots of GitHub repository pages are shown side-by-side. Both pages belong to the user 'corsario85' and are titled 'gh-eii-repos'. The top repository is a fork from 'idafensp/gh-eii-repos'. The bottom repository is also a fork from 'idafensp/gh-eii-repos'. Both repositories have a single branch named 'main' and no tags. The commit history shows four commits from the user 'idafensp otro'. The README.md file contains the following text:

```
gh-eii-repos
Lista de repositorios de alumnos del curso de GitHub EII 2021
```

The top screenshot has a red box highlighting the repository title 'corsario85 / gh-eii-repos' and the top navigation bar. The bottom screenshot has a red box highlighting the top navigation bar.





GitHub

Ejercicio final

The screenshot shows a GitHub repository page for 'corsario85 / gh-eii-repos'. The 'Pull requests' tab is highlighted with a red box. The main content area displays a commit from 'corsario85' titled 'Añadiendo lista corsario85'. This commit has '2 contributors'. A red box highlights the contributor list. On the right, there's a 'History' link and a 'Latest commit' message. Below the commit, there's a file statistics section ('1 lines (1 sloc) | 58 Bytes') and a toolbar with various icons. A red circle highlights the edit icon (pencil) in the toolbar.



GitHub

Ejercicio final



A screenshot of the GitHub pull request interface. At the top, there are filters, a search bar containing "is:pr is:open", and buttons for "Labels 9", "Milestone 0", and "New pull request". The "New pull request" button is highlighted with a red box. Below this, a message says "Welcome to pull requests!" followed by instructions: "Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should [create a pull request](#)".

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

A screenshot of the GitHub comparison interface. It shows a comparison between "base repository: idafensp/gh-eii-repos" and "head repository: corsario85/gh-eii-repos", both set to "base: main" and "compare: main". A red box highlights the message "✓ Able to merge. These branches can be automatically merged.". Below this, a light blue box contains the instruction "Discuss and review the changes in this comparison with others. [Learn about pull requests](#)". On the right, a green "Create pull request" button is highlighted with a red box. At the bottom, it shows "1 commit", "1 file changed", and "1 contributor". The commit details show "Commits on Dec 1, 2021" with a message "Añadiendo lista corsario85" and a timestamp "corsario85 committed 8 minutes ago".



GitHub

Ejercicio final



Añadiendo lista corsario85 #1

[Open corsario85 wants to merge 1 commit into `idafensp:main` from `corsario85:main`](#)

Conversation 0 Commits 1 Checks 0 Files changed 1

 corsario85 commented now

No description provided.

 Añadiendo lista corsario85 Verified 0455a1d

Add more commits by pushing to the `main` branch on [corsario85/gh-eii-repos](#).

 This branch has no conflicts with the base branch
Only those with write access to this repository can merge pull requests.

 Write Preview

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

[Close pull request](#) [Comment](#)

Remember, contributions to this repository should follow our GitHub Community Guidelines.



GitHub

Ejercicio final



Screenshot of a GitHub repository page for `idafensp / gh-eii-repos`. The repository is public and has 1 pull request.

The pull request list shows one open pull request:

- Añadiendo lista corsario85** #1 opened 1 minute ago by corsario85

Filters applied: `is:pr is:open`

Labels: 9

Milestones: 0

New pull request

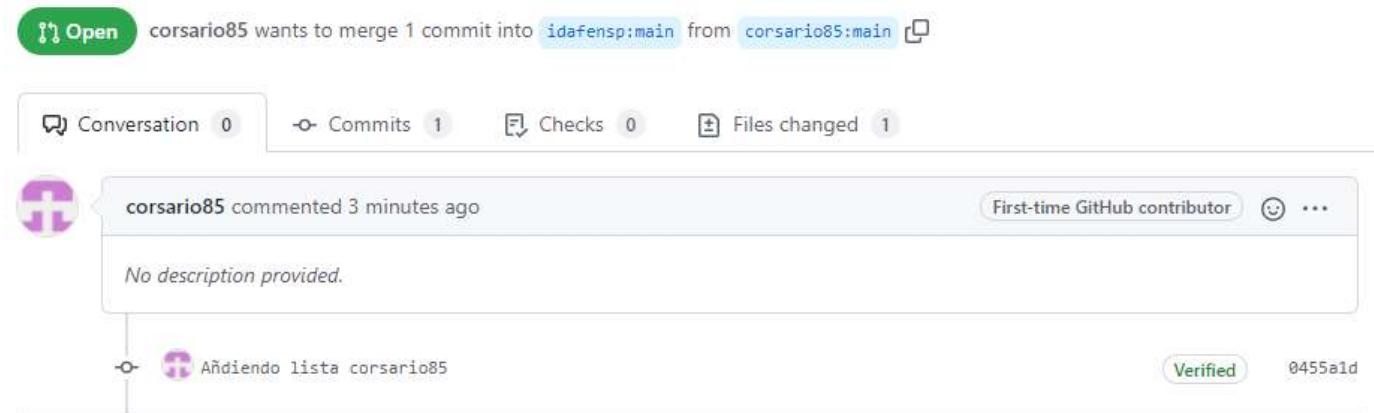
ProTip! Exclude your own issues with `-author:idafensp`.



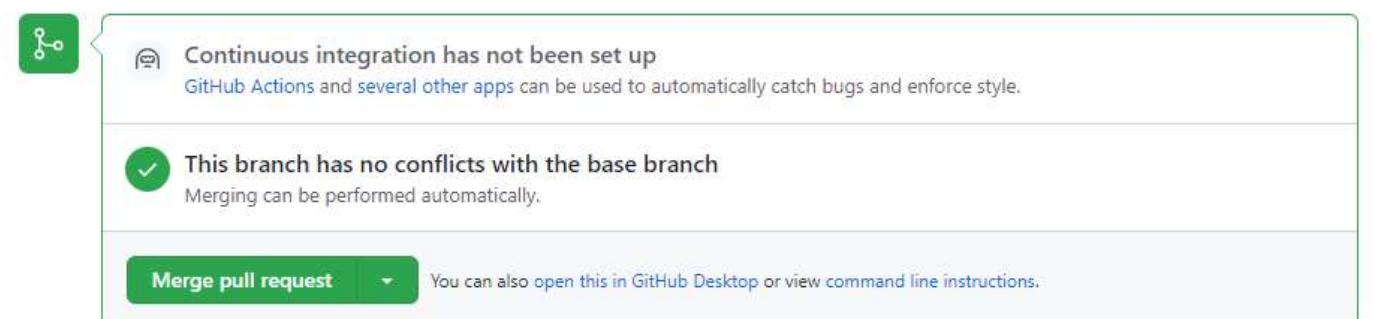
GitHub

Ejercicio final

Añdiendo lista corsario85 #1



Add more commits by pushing to the `main` branch on [corsario85/gh-eii-repos](#).





¿PREGUNTAS?



¡Gracias!



ULPGC
Universidad de
Las Palmas de
Gran Canaria

Iniciación a GitHub

Idafen Santana Pérez, PhD
Grupo de Tecnología Médica y
Audiovisual

04/03/2022