



ARQUITECTURA DEL SOFTWARE

2024-25

Jose Emilio Labra Gayo
Pablo González
Irene Cid Rico

Laboratorio 1

Introducción a la práctica
Organización de equipos
Git
Github



Escuela de
Ingeniería
Informática



Universidad de Oviedo

● Introducción a la práctica

¿Qué vamos a hacer?

Diseñar y desarrollar un juego online de preguntas y respuestas que se generen automáticamente utilizando Wikidata.

¿Qué **recursos** debemos utilizar?

- El [website](#) de Arquitectura del Software donde se encontrará toda la documentación de la asignatura.
- [Campus virtual](#).
- La [especificación](#) de WIQ.
- Github del proyecto [repositorios](#) .

¿Cómo se va a **evaluar** la práctica?

70% - Trabajo en grupo 

30% - Trabajo individual 

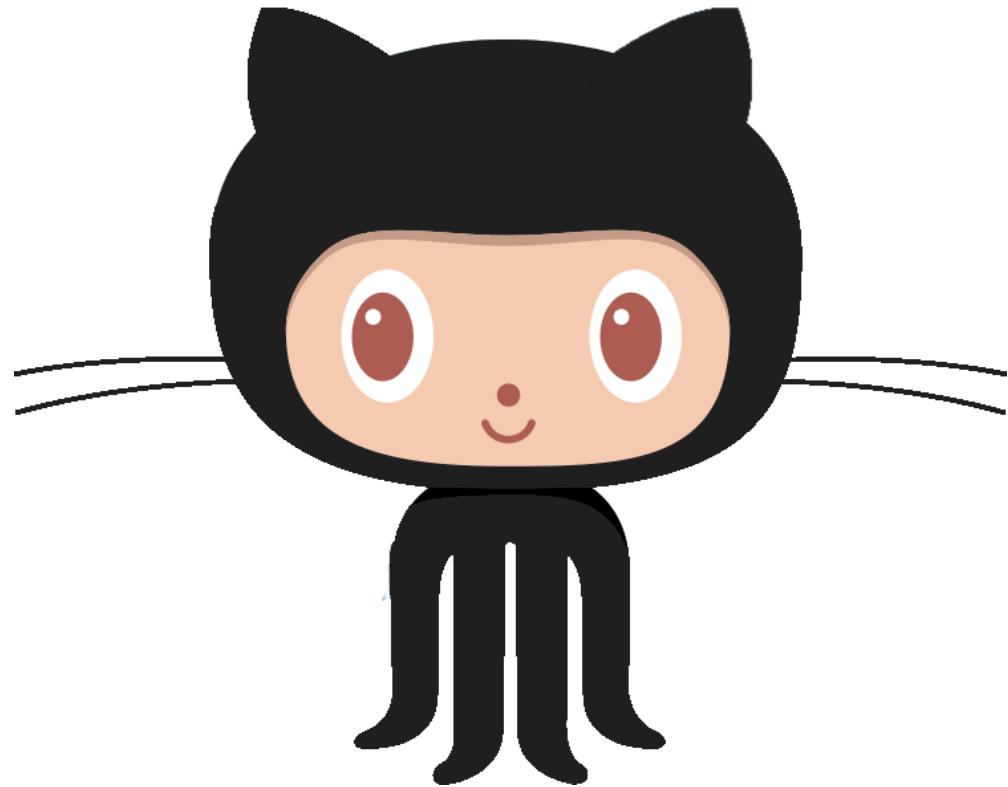
● Organización de equipos

Actas de reuniones

- Clases de práctica == reunión.
 - Se pueden hacer otras reuniones aparte de las clases de laboratorio
- **Obligatorio** tomar acta de las reuniones.
- Se utilizará la sección “wiki” del repositorio para las actas.
- Formato mínimo **obligatorio**:
 - Fecha 
 - Lista de participantes 
 - Acuerdo en el Reparto de trabajo adoptados para la próxima sesión (issues abiertos) 
 - Revisión de estado de tareas en reuniones anteriores
 - Enlaces a **Issues** y **Pull requests** 
 - Breve descripción decisiones tomadas
 - Preferible enlazar a registros de decisiones arquitectónicos (<https://adr.github.io/>)

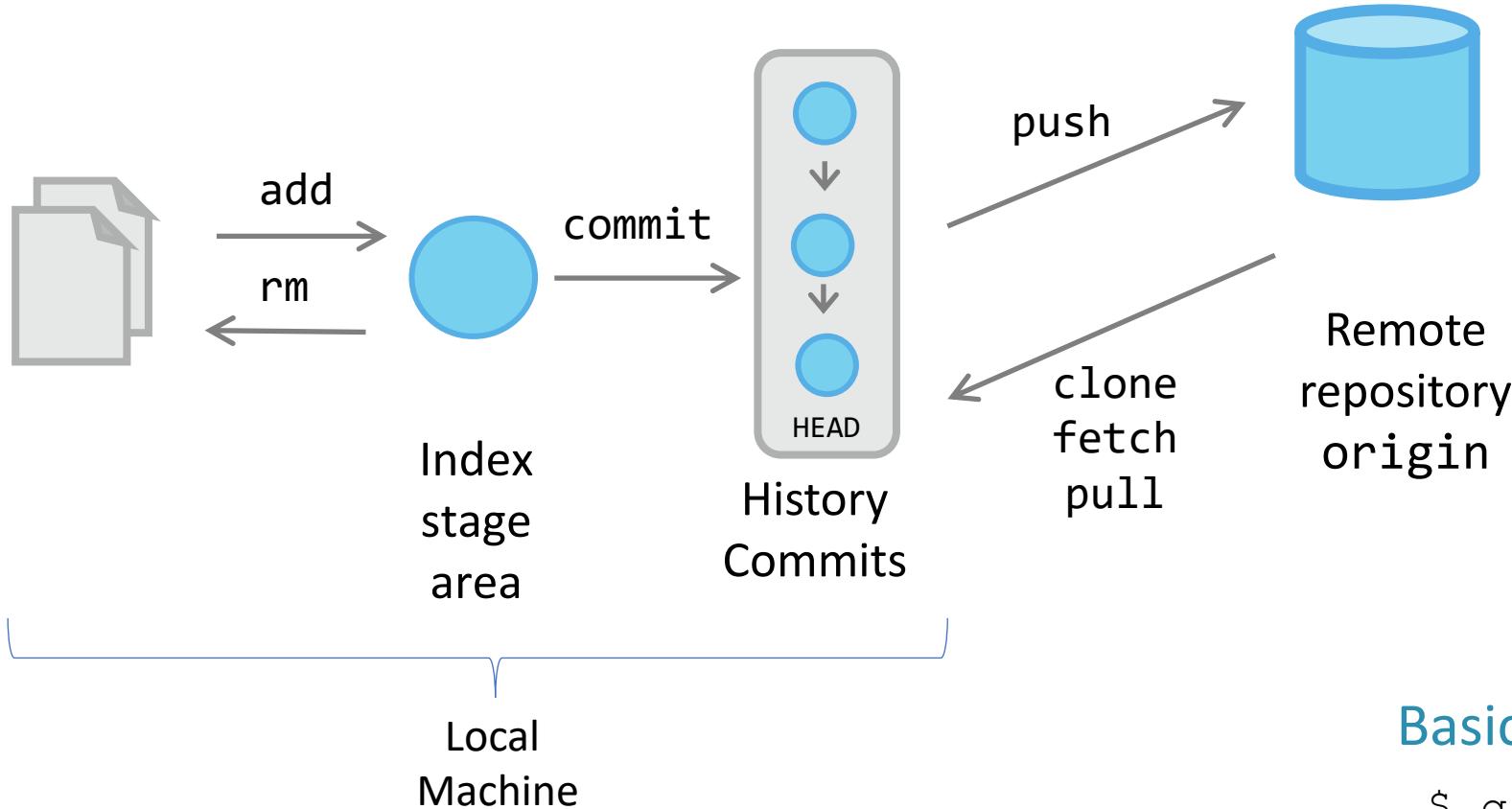


Git





Git



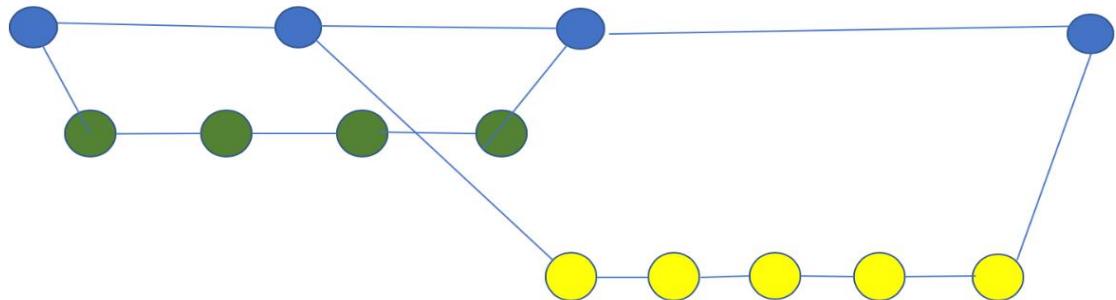
Basic workflow

```
$ git init  
$ git clone urlRepository  
$ git add .  
$ git commit -m "message"  
$ git push origin master
```



Git

Trabajar con ramas



- Crear una rama:
`$ git checkout -b rama1`
- Ver en qué rama estamos
`$ git branch`
- Cambiar de rama
`$ git checkout master`
- Ver los cambios entre ramas
`$ git diff --stat master rama1`
- Fusionar ramas
`$ git checkout master`
`$ git merge --no-ff rama1`
- Eliminar la rama
`$ git branch -d rama1`
- Crear rama **develop**
`$ git checkout -b develop`
- Añadir la rama a nuestro repositorio
`$ git push origin develop`



Git

Estrategias de ramificación

- Todas las estrategias se basan esencialmente en la forma en la que van a crear o no ramas y fusionarlas a la rama principal.
- **No existe** una estrategia perfecta 😢. Depende de muchos factores desde la veteranía del equipo hasta las tecnologías utilizadas o las políticas de la organización.
 - <https://martinfowler.com/articles/branching-patterns.html#Trunk-basedDevelopment>
- Varias estrategias:
 - Git Flow. Vincent Driessen en 2010: [A successful Git branching model](#)” (Tiene código comandos git).
 - [GitHub Flow](#).
 - Trunk-based development: <https://trunkbaseddevelopment.com/>



Pull Request

Pull Request es la aportación más importante que ha hecho GitHub:
Poder integrar a un repositorio código hecho por otros usuarios.

Pasos

- Crear la rama

```
$ git flow feature start RE1 develop
$ git checkout -b feature-RE1 develop
```
- Añade tu nombre en **README.md** en el apartado *Colaboradores*
- Subir los cambios en local

```
$ git add .
$ git commit
```
- Subir los cambios

```
$ git push --set-upstream origin feature-RE1
```
- Ir a github y **solicitar una pull request**

The screenshot shows a GitHub repository summary page. At the top, there are four metrics: 5 commits (red bar), 3 branches (blue bar), 0 packages (grey bar), and 0 releases (yellow bar). Below these, a section titled "Your recently pushed branches:" lists two branches: "develop (about 1 hour ago)" and "feature/RE1 (1 minute ago)". To the right of the branches, there are two green buttons labeled "Compare & pull request". A red oval and a red arrow point to the second button, indicating the action to take.



Pull request

The screenshot shows the GitHub interface for creating a pull request. The top navigation bar includes 'Code', 'Issues 4', 'Pull requests 2', 'Actions', 'Projects', 'Wiki', 'Security 1', 'Insights', and 'Settings'. The main section is titled 'Open a pull request' with the sub-instruction: 'Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#)'. The pull request creation form has the following steps:

1. 'base: master' dropdown.
2. 'compare: feature-new-cool-thing' dropdown, with a note: 'Able to merge. These branches can be automatically merged.'
3. User profile picture and commit message: 'Fix problem when creating an user'.
4. 'Write' tab selected, showing the commit message content:

```
# Changes  
Use uuid v4 instead a manual generated id with Math.random to avoid problems when using the id in some places.  
|  
# Fixed issues  
Fixed #30 on generating new users
```
5. 'Create pull request' button.

Annotations with arrows point to specific elements:

- A blue arrow points from the text 'Rama destino.' to the 'base: master' dropdown.
- A blue arrow points from the text 'Rama que queremos fusionar' to the 'compare: feature-new-cool-thing' dropdown.
- A blue arrow points from the text 'Añadir contexto adicional.' to the 'Write' tab area.
- A blue arrow points from the text 'Crear pull request' to the 'Create pull request' button.



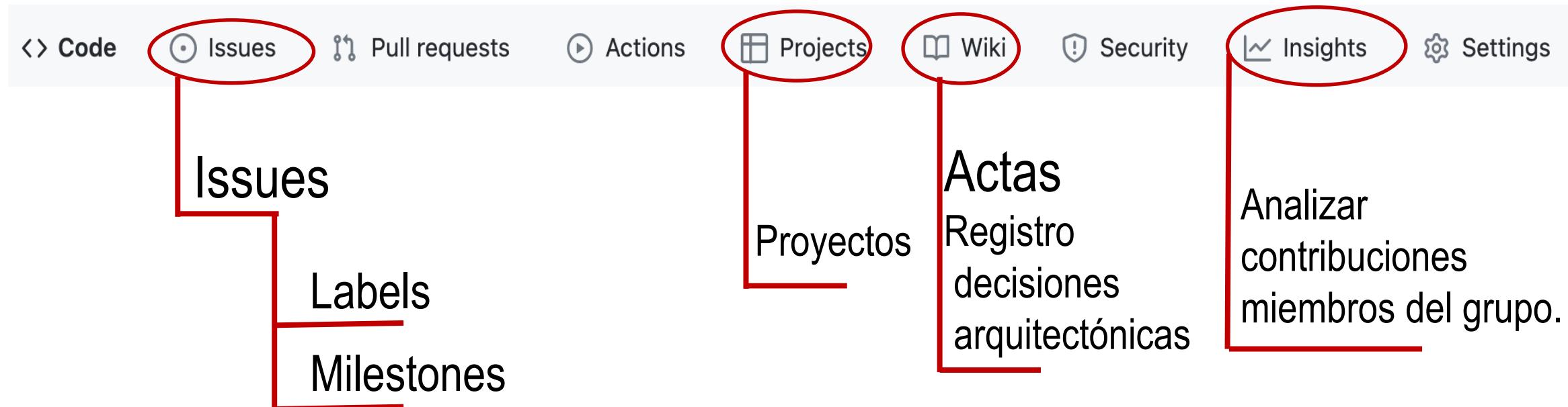
GitHub

Ventajas de la gestión de proyectos

- Planificación del proyecto (**Futuro**).
- Controlar el estado actual del proyecto. (**Presente**)
 - Detectar cuellos de botella.
 - Carga de trabajo de los miembros del equipo.
 - Problemas actuales.
- Reporte del desempeño (**Pasado**).
 - Evaluación de la contribución de los miembros del equipo.



Herramientas de Github para gestión de proyectos





GitHub

Sprint: Projects > MagiDoc - Development

+ Add column X Edit columns More

Backlog

- ① Create Docker image for project magidoc-admin-server#10 opened by jflory? **enhancement priority:low**
- ② Throw errors for missing dependencies magidoc-admin-server#11 opened by jflory? **enhancement high priority**
- ③ Curling results in empty response, server crashes magidoc-admin-server#12 opened by jflory? **bug priority:high priority:critical**
- ④ Serve aggregations by admin level magidoc-admin#13 opened by mikelab **enhancement**
- ⑤ Base API in Swagger Spec Has Extra / magidoc-openapi#14 opened by shentao **enhancement**
- ⑥ Use arguments to overwrite default configurations magidoc-aggregates-mobility#24 opened by carlosbodas **enhancement**
- ⑦ Return hasProbData in API responses for schools magidoc-openapi#24 opened by ayashot **enhancement**
- ⑧ expired token warning gets overridden by SseHandler library and return null instead of just magidoc-openapi#18 opened by mikelab **enhancement**

To Do

- ⑨ Colombo shapefile (impl.json) should be fetched from api rather than be required in react app. magidoc-react#10 opened by mikelab **enhancement**
- ⑩ Fix "the_lines" issue in mock-apps public/lib.js magidoc-react#11 opened by jflory? **bug**
- ⑪ Convert and refactor code in action-select-country.js magidoc-react#12 opened by mikelab **enhancement**
- ⑫ Add disclaimer popup on load magidoc-react#13 opened by mikelab **enhancement**
- ⑬ Rearrange login code to work in production magidoc-react#14 opened by mikelab **enhancement**

In progress

- ⑭ Add contributing guidelines placeholder (closed #20) magidoc-admin#22 opened by jflory? **enhancement**
- ⑮ Determine where to begin writing tests and refactoring magidoc-react#15 opened by mikelab **enhancement**
- ⑯ Map relationships between apps, servers, and azure resources magidoc#15 opened by mikelab **in progress internal organization priority:critic**
- ⑰ Mobility: Map should display date for mobility being displayed magidoc-react#12 opened by mikelab **enhancement**

Done

- ⑱ Rename test suite (run-test.sh), add comments magidoc-admin#17 opened by jflory? **enhancement**
- ⑲ Refactor Web-GLayers.js magidoc-map#11 opened by jflory? **bug**
- ⑳ Create production/staging/ide deployment slots for Mobility Mapping magidoc#16 opened by mikelab **enhancement internal organization priority:critic**
- ㉑ Set up automatic CI testing for new pull requests magidoc#10 opened by jflory? **enhancement priority:high**
- ㉒ Add loading spinner to mobility map magidoc-map#9 opened by mikelab **enhancement**
- ㉓ Add spinner mobility magidoc-map#10 opened by carlosbodas **enhancement**
- ㉔ Add CodeClimate maintainability + test coverage checks to CI magidoc-openapi#12 opened by jflory? **enhancement**
- ㉕ Add TravisCI and CodeClimate tests magidoc-openapi#13 opened by jflory? **enhancement**

+ Add column

Automated in Backlog Manage Automated in In progress Manage Automated in Done Manage



GitHub

Tablero de proyecto

- Podemos crear un tablero KanBan.
- Muy Flexible. Podemos hacer un tablero muy simple (y de gestión manual)
- GitHub permite automatizar el flujo de trabajo a través de la gestión de Issues y pull requests.
- Podemos crear **diferentes** tableros (documentación, backend,...)

The screenshot shows the GitHub interface for managing projects. At the top, there are navigation links: Code, Issues, Pull requests, Actions, Projects (which is highlighted with a red oval), and Below this, there's a search bar with the query 'is:open'. A dropdown menu labeled 'Projects' is open. On the right side of the dashboard, there's a large green button labeled 'New project' with a red oval around it. At the bottom, it says 'There aren't any projects yet'.

The screenshot shows the 'Create a new project' dialog box. It has a title 'Create a new project' and a subtitle 'Coordinate, track, and update your work in one place, so projects stay transparent and on schedule.' There are two main input fields: 'Project board name' (with a red oval around it) and 'Description (optional)'. Below these is a section for 'Project template' with a note 'Save yourself time with a pre-configured project board template' and a dropdown menu set to 'Template: None' (with a red oval around it). At the bottom is a large green 'Create project' button (with a red oval around it).

Opciones de automatización



Issues



Puede entenderse como un post-it en el tablero **Kanban**.

👉 Cualquier problema o tarea que nos encontramos con el proyecto debe tener un Issue.

Utilizan **Markdown**.

Escribir solo lo necesario para que se entienda el problema. Se puede acompañar imágenes o hiperenlaces en la explicación.

Es el eje la gestión de un proyecto. Cuantos más Issues, mayor información acerca del estado del proyecto.

Un Issue puede relacionarse con otras entidades del repositorio de GitHub. Esta interacción potencia enormemente la información.

- Responsables
- Etiquetas
- Milestones



SOLO SE EVALUA LA COMUNICACIÓN QUE ESTÉ EN EL REPOSITORIO DE GITHUB



GitHub



Investigate to what extent HTTPS should be mandatory #1091

[New issue](#)

Estado Issue ←

Open

RubenVerborgh opened this issue on 9 Dec 2021 · 2 comments

Explicación
del
problema

Comentarios



RubenVerborgh commented
on 9 Dec 2021

Member



...

In some previous testing, I have come across preliminary evidence that some Solid-related functionality only works over HTTPS. In particular, when running Mashlib as the on-server UI, authentication seems to break because the server is not running over HTTPS.

Whereas this is actually a question for the bigger Solid ecosystem, we can test some assumptions on CSS and turn them into recommendations.

If the answer is that some functionality only works over HTTPS, then we might want to make CSS start over HTTPS out of the box (e.g., by auto-generating `localhost` certificates etc.).

RubenVerborgh added the `task` label on 9 Dec 2021

Asignees

RubenVerborgh

→ Asignado a

Labels

task

→ Etiquetas

Projects

None yet

→ Tablero de Proyecto

Milestone

No milestone

→ Milestones

Linked pull requests

Successfully merging a pull request may close this issue.

None yet

→ Pull request linkado

Log del Issue

GitHub

ETIQUETAS



- Sirven para catalogar los issues.
- GitHub provee unas etiquetas genéricas.
- Podéis crear las etiquetas que queráis.(backend, frontend, bbdd, jerarquías)
- Cuanto más organizado mejor.
- Cuidado con no pasarse. Si se crean demasiadas pierden su función que es categorizar.
- GitHub permite personalizar el color y utilizar emojis para dar estilo y ser de ayuda visual 🎨.

GitHub

The screenshot shows the GitHub Issues page with the following navigation bar:

- <> Code
- Issues 105
- Pull requests 13
- Discussions
- Actions
- Projects 1

Below the navigation bar, there are two tabs: **Labels** (selected) and **Milestones**. A search bar says "Search all labels".

The main area displays a list of issues with the following details:

Nombre etiqueta	Issue Description	Numero de Issues etiquetados
worker threads	Making the server multithreaded	5
bug	Something isn't working	17
dependencies	Pull requests that update a dependency file	
developer experience		5
difficulty:high		2
difficulty:low		13
difficulty:medium		9

Pequeña descripción

Número de Issues etiquetados

Uso común en jerarquías dificultad o prioridad suelen tener distintos niveles

Uso de FAQ de arquisoft para preguntas

<https://github.com/Arquisoft/faq/issues>

Compartir preguntas del curso

- Se permite añadir cualquier issue que describa alguna pregunta relacionada con el curso en inglés o español
 - Cualquiera puede contribuir respondiendo la pregunta o añadiendo comentarios
 - Los que contribuyan deben seguir un código de conducta que respete las consideraciones éticas de un curso de la Universidad de Oviedo
- Los profesores podrán borrar cualquier issue o comentario que consideren inapropiado para el curso

Doc adicional

- Diapositivas de Labra sobre [Git](#).
- [Guía rápida](#) de Pablo Gonzalez
- [introducción a git](#) de Hugo Lebreiro
- [guía](#)
- [learngitbranching.js.org](#)

Doc adicional

- Contenido [Miguel Angel Durán.](#)
- Libro de Miguel Angel [Aprendiendo Git.](#)

Youtube

- [Explicación conceptos de Git en 15 minutos](#)
- [Curso/tutorial](#) desde 0
- [GitHub vs GitLab](#) Este video explica la importancia actual de estas empresas.

Práctica de este año



Imagen generada con ChatGPT

Enunciado: <https://arquisoft.github.io/course2425.html#labs>

Primer entregable

- Documentación, versión 0.1
- Primera prueba de concepto
- Aplicación desplegada
 - Aplicación web conectada con GameY
 - Aplicación web accediendo a base de datos

