



ARQUITECTURA DEL SOFTWARE

2025-26

Jose Emilio Labra Gayo

Pablo González

Diego Martín

Celia Melendi



Escuela de
Ingeniería
Informática



Universidad de Oviedo

Laboratorio 1

Introducción a la práctica

Organización de equipos

Git

Github

● Introducción a la práctica

¿Qué vamos a hacer?

Diseñar y desarrollar un juego online basado en el juego Y.

¿Qué **recursos** debemos utilizar?

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

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

70% - Trabajo en grupo 

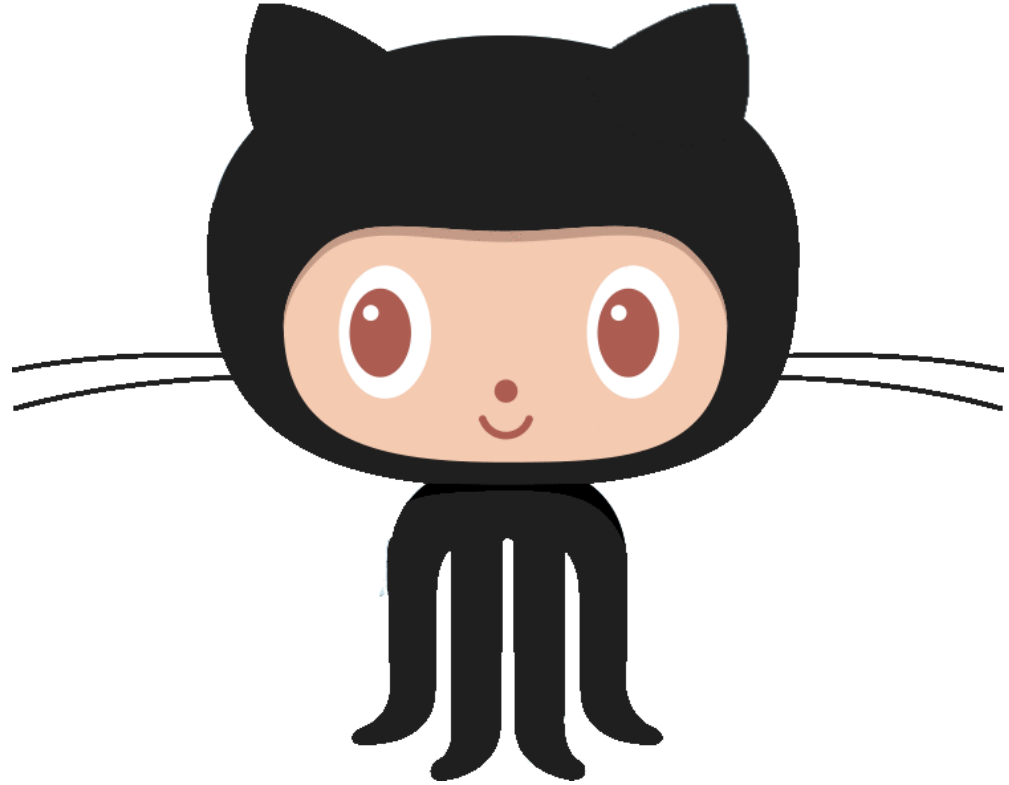
30% - Trabajo individual 

● Organización de equipos

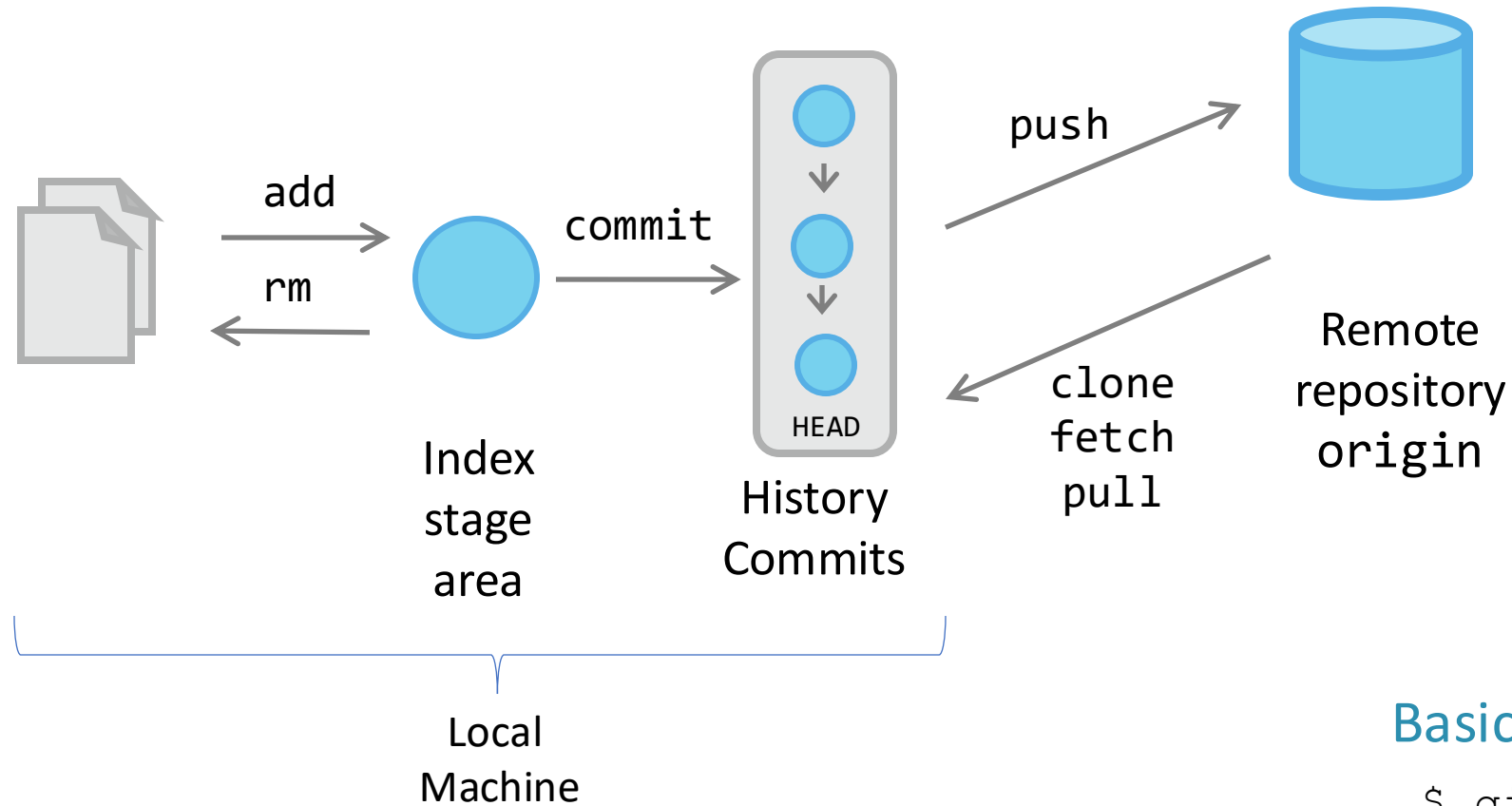
Actas de reuniones

- 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

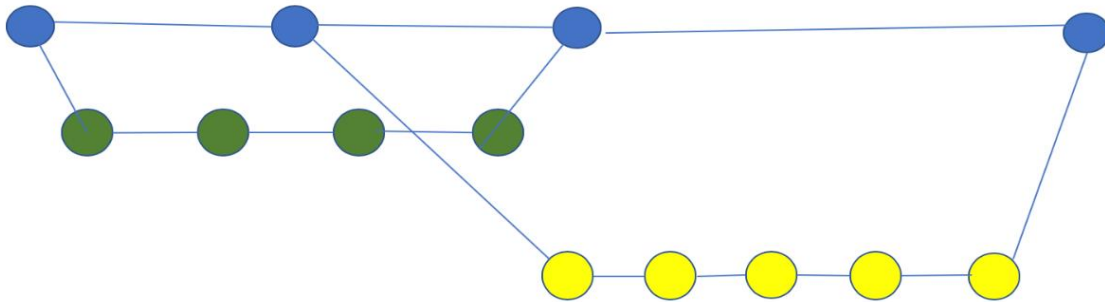


Basic workflow

```
$ 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 ramal`
- Ver en qué rama estamos
`$ git branch`
- Cambiar de rama
`$ git checkout master`
- Ver los cambios entre ramas
`$ git diff --stat master ramal`
- Fusionar ramas
`$ git checkout master`
`$ git merge --no-ff ramal`
- Eliminar la rama
`$ git branch -d ramal`
- 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

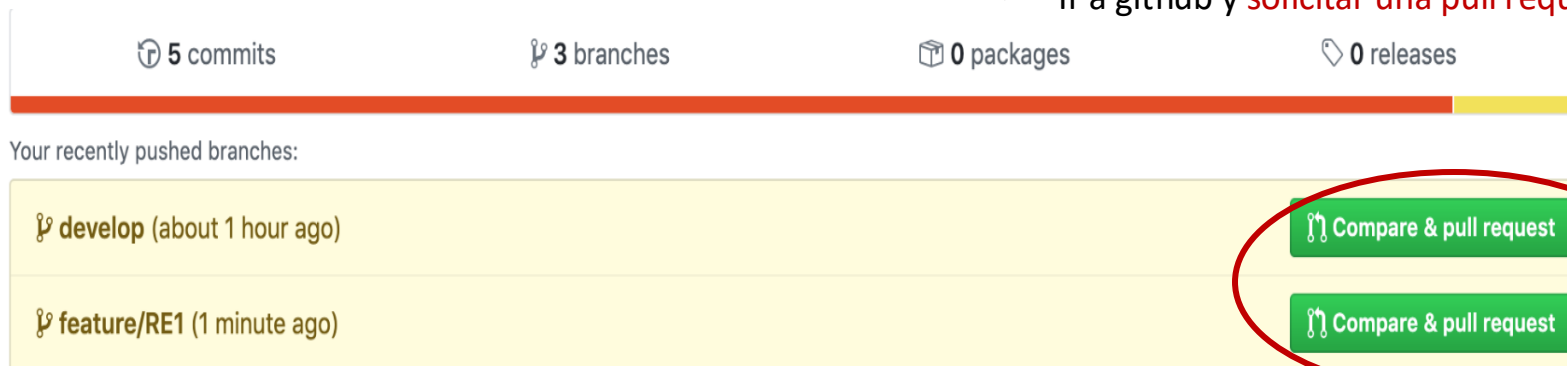
```
$ git flow feature start RE1 develop
```

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

```
$ git add .
```

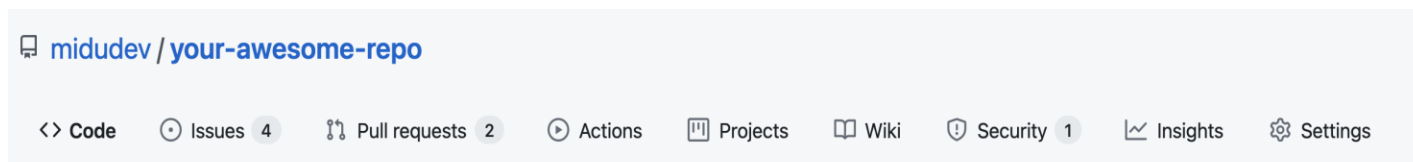
```
$ git commit
```
- Subir los cambios

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





Pull request



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

Rama destino. → 1 base: master

← 2 compare: feature-new-cool-thing → **Rama que queremos fusionar**

3 Fix problem when creating an user

4

Añadir contexto adicional. →

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

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

5 **Crear pull request**

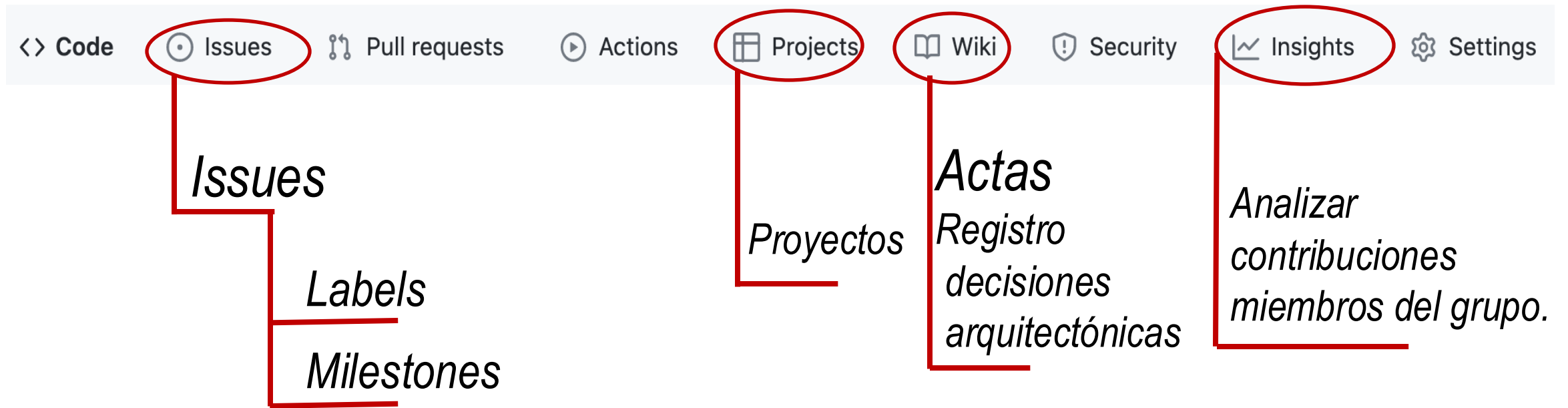
● GitHub

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.

● GitHub

Herramientas de Github para gestión de proyectos





GitHub

GitHub Projects - MapleBox - Development

Backlog To Do In progress Done

33 items

1 Create Docker image for project
magbox-admin-server10 opened by jflory? **unassigned** **priority: low**

2 Throw errors for missing dependencies
magbox-aggregates-mobility12 opened by jflory? **unassigned** **high severity**

3 Curling results in empty response, server crashes
magbox-admin-server10 opened by jflory? **bug** **information** **priority: port**

4 Serve aggregations by admin level
magbox-open-api42 opened by miketab

5 Base API in Swagger Spec Has Extra /
magbox-open-api42 opened by thecode

6 Use arguments to overwrite default configurations
magbox-aggregates-mobility12 opened by thecode **unassigned**

7 Return hasProbsData in API response for schools
magbox-open-api44 opened by aynd17

8 expired token warning gets overridden by libhacker library and return true instead of false
magbox-open-api18 opened by miketab

1 Colombia shapable (mpc.json) should be fetched from api rather than be required in react app
magbox-mapbox opened by miketab

2 Fix "file_times" issue in react-app: publicFile.js
magbox-mapbox opened by jflory? **bug**

3 Comment and refactor code in action-select-country.js
magbox-mapbox opened by miketab

4 Add disclaimer popup on load
magbox-mapbox opened by miketab

5 Reconfigure login code to work in production
magbox-mapbox opened by miketab

1 Add contributing guidelines placeholder (closes #22)
magbox-open-api42 opened by jflory? **unassigned**

2 Determine where to begin writing tests and refactoring
magbox-mapbox opened by miketab

3 Map relationships between apps, maps, and azure resources
magbox15 opened by miketab **in progress** **internal organization** **priority: port**

4 Mobility: Map should display data for mobility being displayed
magbox-mapbox12 opened by miketab

1 Rename test suite (run-tests.sh), add comments
magbox-open-api42 opened by jflory? **unassigned**

2 Refactor WebGLLayer.js
magbox-mapbox11 opened by jflory? **bug**

3 Create production staging dev deployment slots for Mobility Mapping
magbox16 opened by miketab **unassigned** **internal organization** **priority: low**

4 Set up automatic CI testing for new pull requests
magbox16 opened by jflory? **unassigned** **priority: high**

5 Add loading spinner to mobility map
magbox-mapbox17 opened by miketab

6 Add spinner mobility
magbox-mapbox14 opened by thecode **unassigned**

7 Add CodeClimate maintainability + test coverage checks to CI
magbox-open-api42 opened by jflory? **unassigned**

8 Add TravisCI and CodeClimate tests

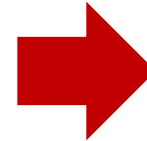
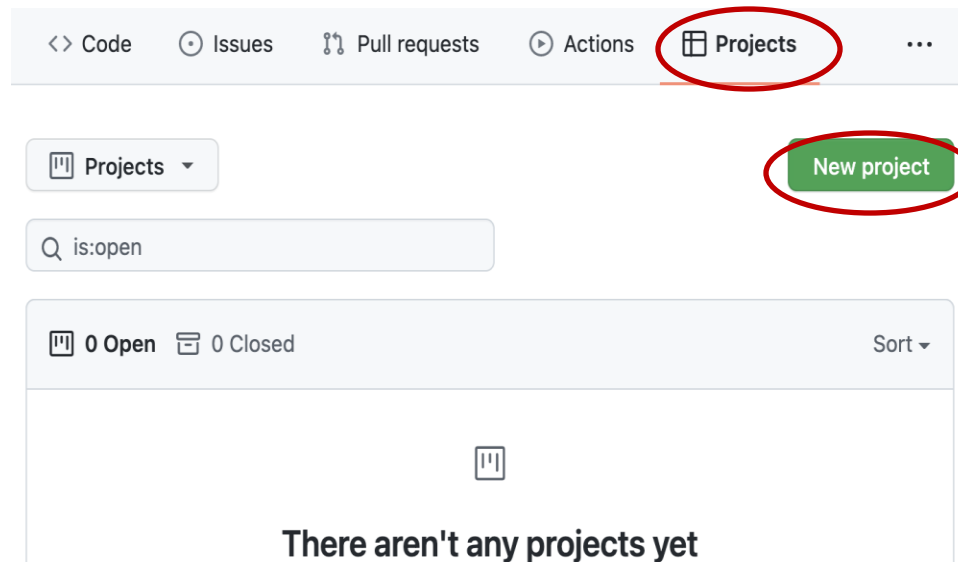
Automated as Backlog Manage Automated as In progress Manage Automated as Done Manage

+ Add column

● 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,...)



Create a new project

Coordinate, track, and update your work in one place, so projects stay transparent and on schedule.

Project board name

Project board name

Description (optional)

Project template

Save yourself time with a pre-configured project board template

Template: None

Create project

Opciones de automatización



Issues

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




👉 Cualquier problema o tarea que nos encontremos 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



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.).

Assignees



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.

Pull request
linkado

None yet

Log del Issue




RubenVerborgh added the task label on 9 Dec 2021

task

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  .



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

Labels Milestones Search all labels

32 labels

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

Nombre
etiqueta

Pequeña descripción

Numero 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 González
- [introducción a git](#) de Hugo Lebrede
- [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

