

ENTORNOS DE DESARROLLO

UD4.4 Github - Repositorios
remotos y distribuidos

Contenidos

- ❖ Plataformas online para gestión de Repositorios.
 - Github
 - Gitlab
 - Bitbucket
- ❖ Alta en Github
- ❖ Gestión de usuarios y permisos (AccessToken)
- ❖ Configuración de repositorio remoto
- ❖ Ramas
- ❖ Forks
- ❖ Pull Requests

Plataformas online para gestión de repositorios

Generalmente, los desarrolladores están de acuerdo en que **GitLab y GitHub** tienen la mejor interfaz. Muchos desarrolladores prefieren **GitHub** porque es más popular y se usan para la navegación.

Sin embargo, los equipos que usan **Jira** (herramienta de gestión de tareas) se benefician de tener integraciones perfectas con el ecosistema de *Atlassian*, por lo que **BitBucket** es una elección natural.

[Bitbucket vs Github vs Gitlab Detailed Comparison 2022](#)

Plataformas online para gestión de repositorios

- **GitHub:** muchas guías y tutoriales, se puede acceder rápidamente a la funcionalidad básica, la interfaz es minimalista y fácil de entender. La desventaja es la falta de integraciones nativas con Jira y la falta de funciones avanzadas de CI (Integración continua).
- **BitBucket:** la interfaz puede resultar confusa, pero a medida que la plataforma se vuelve más popular, hay muchos más recursos. Los equipos que usan Jira suelen preferir BitBucket porque ofrece una visibilidad mucho mejor del proyecto en general.
- **GitLab:** el servicio no está tan consolidado como el de la competencia, por lo que aún faltan guías de interfaz y tutoriales. Sin embargo, la interfaz en sí está muy bien organizada, incluso los complementos avanzados son fáciles de localizar.

Alta en Github

- Vamos a registrarnos en Github y nos crearemos una cuenta con la que trabajaremos a lo largo de este tema.
- <https://github.com/login> → Create an account

Una vez tenemos la cuenta creada vamos a realizar una viaje a través de su interfaz y veremos:

- Gestión de la cuenta
- Crear repositorios
- Permisos de acceso
- Colaboradores
- Gestión de cambios
- Sincronización entre repositorios origin y forks

Alta en Github - Gestión de la cuenta

[Overview](#) [Repositories 29](#) [Projects](#) [Packages](#) [Stars 2](#)

Popular repositories Customize your pins

[vpn-add](#) Public
☆ 2

[project](#) Public
● PHP

[deployer](#) Public
● Shell

[docker-python](#) Public
● Dockerfile ☆ 1 🍴 1

[docker-images](#) Public
Forked from km77com/docker-images
Dockerfile configurations for images published at <https://hub.docker.com/u/km77/>
● Dockerfile

[recipes-drf-pycones19](#) Public
Forked from dukebody/recipes-drf-pycones19
Tutorial for creating an API for cooking recipes using DRF
● Python

1,459 contributions in 2019 Contribution settings ▾

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

Mon

Wed

Fri

Learn how we count contributions

Less More

2023

2022

2021

2020

2019

2018

2017

2016

2015

2014

2013

@km77com

Activity overview

Contributed to [buig/duroc](#), [buig/april](#), [buig/payvifema](#) and 7 other repositories

8% Code review

70% Commits

22% Pull requests

Issues

[Overview](#) [Repositories 29](#) [Projects](#) [Packages](#) [Stars 2](#)

Find a repository... Type ▾ Language ▾ Sort ▾ [New](#)

[furnipop_api](#) Private
● Python 🍴 1 Updated 2 days ago

[congresoFedemAPI](#) Private
Forked from barreratoribiolauro/congresoFedemAPI
● PHP 🍴 2 Updated 2 days ago

[docusenia](#) Private
● CSS Updated 4 days ago

[lasenia-docker-reverse-proxy](#) Private
● Dockerfile Updated 5 days ago


[macbuig-server-connector](#) Private
● PHP Updated last week

[add-multiverse](#) Public
● PHP Updated on Oct 24, 2022

[horario_senia_bot](#) Private
Telegram Bot: Horario
● Python Updated on Sep 27, 2022

[macbuig-infrastructure](#) Private
Updated on Jun 15, 2022

Alta en Github - Gestión de la cuenta

**Antonio Calabuig Puigvert**
Your personal account

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Moderation

Code, planning, and automation

Repositories

Codespaces

Packages

Copilot

Pages

Saved replies

Security

Code security and analysis

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

Developer settings

Public profile

Name

Antonio Calabuig Puigvert

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email

Select a verified email to display

You can manage verified email addresses in your [email settings](#).

Bio

Tell us a little bit about yourself

You can @mention other users and organizations to link to them.

URL

Twitter username

Company

You can @mention your company's GitHub organization to link it.

Location

☐ Display current local time

Other users will see the time difference from their local time.

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

Contributions & Activity

☐ Make profile private and hide activity

Enabling this will hide your contributions and activity from your GitHub profile and from social features like followers, stars, feeds, leaderboards and releases.

☒ Include private contributions on my profile

Your contribution graph, achievements, and activity overview will show your private contributions without revealing any repository or organization information. [Read more](#).

Update preferences

Opciones de configuración de menú y personalización.

Alta en Github - Gestión de la cuenta

Settings / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens ^{Beta}

Fine-grained tokens

Tokens (classic)

Personal access tokens (classic)

Generate new token ▼ Revoke all

Tokens you have generated that can be used to access the [GitHub API](#).

Bashscript — <code>admin:org, admin:repo_hook, delete_repo, repo</code>	Last used within the last week	Delete
⚠ This token has no expiration date.		

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Developer Settings, aquí podemos crear nuestro Personal Access Token para realizar y actualizar cambios en los repositorios

- [Creating a personal access token - GitHub Docs](#)

Alta en Github - Gestión de la cuenta

GitHub Apps
OAuth Apps
Personal access tokens Beta
Fine-grained tokens
Tokens (classic)

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

What's this token for?

Expiration *

30 days The token will expire on Wed, Feb 22 2023

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> manage_runners:org	Manage org runners and runner groups
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks
<input type="checkbox"/> read:repo_hook	Read repository hooks

Nombre del Token

Periodo de expiración

¿Qué puede hacer ese token?


Una vez generado el Token, solo lo podréis copiar una vez y quedará oculto. Tenéis que anotarlo aparte. En caso de pérdida habría que borrar el actual y crear uno nuevo.

Crear repositorios

Create a new repository

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

Owner * Repository name *

 buig /

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

Description (optional)

☒  **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: **None**

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

License: **None**

 You are creating a public repository in your personal account.

Create repository

 Packages

 Stars 2

Type ▾

Language ▾

Sort ▾

 New

Nombre del repo

Descripción (op)

Ámbito

Opcionales

Nuevo repositorio

Los opcionales se recomiendan activarlos en caso de que sea un proyecto que se crea directamente desde Github y no para subir cambios de un repositorio ya existente.

Crear repositorios

Create a new repository

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

Owner *

Repository name *

buig / ed-git

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

Description (optional)

Proyecto de prueba para la asignatura de Entornos de Desarrollo

☒ 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: None

Choose a license

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

License: None

This will set `main` as the default branch. Change the default name in your [settings](#).

1 You are creating a public repository in your personal account.

Create repository

buig / ed-git Public

Pin

Unwatch 1

Fork 0

Star 0

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

main

Go to file

Add file

<> Code

buig Initial commit now 1

README.md Initial commit now

README.md

ed-git

Proyecto de prueba para la asignatura de Entornos de Desarrollo

Go to file

Add file

<> Code

Local

Codespaces New

Clone

HTTPS SSH GitHub CLI

https://github.com/buig/ed-git.git

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

About

Proyecto de prueba para la asignatura de Entornos de Desarrollo

Readme

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Clonar repositorios desde Github

```
git clone https://github.com/buig/ed-git.git
```

Si consultamos los remotos de nuestro repositorio veremos algo así.

```
🍏 ~/Sites/ed/ed-git 🐱 main git remote -v
origin https://github.com/buig/ed-git.git (fetch)
origin https://github.com/buig/ed-git.git (push)
🍏 ~/Sites/ed/ed-git 🐱 main
```

El problema que tenemos con esto, es que ya podemos empezar a hacer cambios pero nos dará error de autenticación por lo que será necesario cambiar la url del remoto para poder subir y bajar cambios.

La url la modificaremos usando el Access Token creado, siguiendo la siguiente sintaxis:

```
git remote remove origin
```

```
git remote add origin https://[USERNAME]:[TOKEN]@github.com/buig/furnipop_api.git
```

Ramas en repositorios remotos

De la misma forma que en un repositorio local, podemos crear ramas, éstas pueden subirse al repositorio remoto y a la inversa.

Desde un repositorio remoto, crear una rama y descargarla o simplemente, una rama creada por un compañero de trabajo y la descargamos a nuestro repositorio local.

Consulta de ramas en repositorio local

```
git branch
```

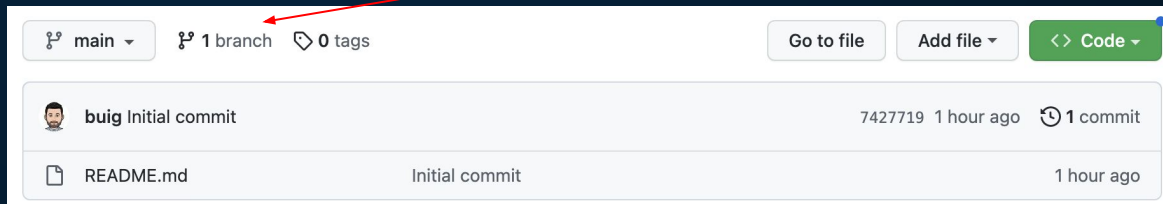
Consulta de ramas en repositorio remotodarth_vader

```
git branch -r [-r remote]
```

Descargar ramas de repositorio remoto

```
git fetch
```

Ramas



Ramas en repositorios remotos

Subir rama a repositorio remoto:

1. Creamos una rama

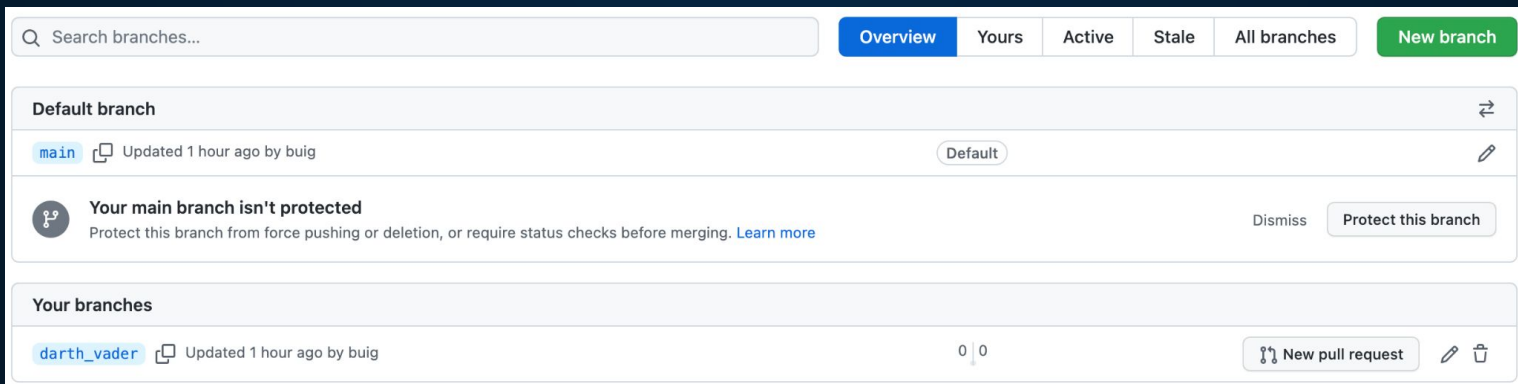
a. `git branch darth_vader`

2. Nos ubicamos sobre la rama local que deseamos subir

a. `git checkout darth_vader`

3. Subimos la rama al repositorio remoto

a. `git push [origin/NOMBRE_DEL_REMOTO] darth_vader`



Ramas en repositorios remotos

Borrar rama de un repositorio remoto:

1. Si borramos una rama de un repositorio remoto, puede ser por varios motivos:
 - a. Hemos hecho cambios en la local y la queremos eliminar para subirla de nuevo
 - b. Hemos acabado el trabajo ya no la necesitamos
 - c. No vamos a hacer uso de ella
 - i. Se puede borrar desde el interfaz de Github o a través del comando. En caso de usar el comando es independiente de la rama sobre la que estemos ubicados en el repositorio local.:
 1. `git push origin :darth_vader`
2. Ya sabemos borrar ramas de un repositorio local
 - a. `git branch -d darth_vader`
 - b. `git branch -D darth_vader [-D]` Solo en el caso de descartar los cambios de la rama local y querer eliminarla

Pruebas en clase

Forks

Una **bifurcación** es un nuevo repositorio que comparte la configuración de visibilidad y código con el repositorio “acendente” original.

Las bifurcaciones se suelen usar para iterar ideas o cambios antes de que se vuelvan a proponer al repositorio ascendente, como en proyectos de código abierto o cuando un usuario no tiene acceso de escritura al repositorio ascendente.

[Cómo realizar un fork](#)

[Clonar un repositorio bifurcado \(forked\)](#)

[Bifurcar un repositorio](#)

Pull Request

Las **Pull Requests** son las solicitudes de incorporación de cambios te permiten comentarle a otros acerca de los cambios que has insertado en una rama de un repositorio en GitHub. Una vez que se abre una solicitud de incorporación de cambios, puedes debatir y revisar los posibles cambios con los colaboradores y agregar confirmaciones de seguimiento antes de que los cambios se fusionen en la rama base.

[Acerca de las solicitudes de incorporación de cambios - GitHub Docs](#)

Pruebas con Fork y Pull Request

Abrimos el siguiente repositorio: https://github.com/buig/hello_class

1. Crea un **fork** del repositorio. Tu fork quedará asociado a tu cuenta.
2. Creamos una rama con nuestro nombre
3. Uno por uno, vamos a ir implementando la función que llame a nuestro nombre y lo imprima.
4. Uno a uno iremos integrando nuestras ramas en el repositorio principal.
 - a. Ello implica, que si nuestra rama está obsoleta, la actualizaremos.
 - b. Probaremos esto subiendo uno o dos cambios a la vez y solucionaremos los conflictos que ocasionen.
5. Los cambios los integraremos en la rama principal mediante Pull Request.

Bibliografía

- [Bitbucket vs Github vs Gitlab Detailed Comparison 2022](#)
- [GitHub](#)
- [Git List Remote Branches](#)
- [Fork a repo - GitHub Docs](#)
- [About pull requests - GitHub Docs](#)