

0.2 Uso de GitHub

Julio Vega

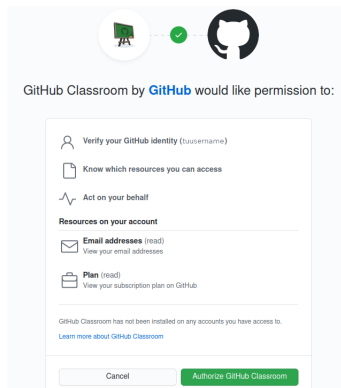
julio.vega@urjc.es



Universidad
Rey Juan Carlos

Contenidos

- 1 Acceso a nuestra clase de prácticas
- 2 Acceso a una nueva práctica
- 3 Cómo trabajar en una práctica con Git
- 4 Anexo. Seguridad en Git



- Con la cuenta creada de GitHub, ya os puedo invitar a nuestra clase.
 - Os paso ahora un enlace por Teams para *enrolaros* a la clase.
- Si accedéis, os aparecerá una pantalla como la que os muestro aquí.
 - Muy importante, comprobar que el *username* que aparece es correcto.
- Más importante, NO PASÉIS DE ESTA PANTALLA TODAVÍA.
 - El siguiente paso, “Authorize GitHub Classroom”, es un paso delicado.

Join the classroom:

idClase

To join the GitHub Classroom for this course, please select yourself from the list below to associate your GitHub account with your school's identifier (i.e., your name, ID, or email).

Can't find your name? [Skip to the next step](#) →

Identifiers	
user1@alumnos.urjc.es	>
user2@alumnos.urjc.es	>
user3@alumnos.urjc.es	>

- Una vez autorizado, nos aparece la pantalla de “Join the classroom”.
- MUY IMPORTANTE: no confundirnos al seleccionar nuestro correo.
 - Es un proceso no reversible, nos impediría trabajar con nuestra cuenta.

Your account is linked to `tuuser@alumnos.urjc.es` on the roster. If this is wrong, please reach out to your instructor.

idClase

Accept the assignment —

Ejemplo inicial

Once you accept this assignment, you will be granted access to the `ejemplo-inicial-tuusername` repository in the [clases-julio](#) organization on GitHub.

Accept this assignment

- Siguiendo el enlace que se os ha proporcionado, aceptamos la tarea.
 - Comprobamos que nuestro *username* es correcto.



You accepted the assignment, **Ejemplo inicial** . We're configuring your repository now. This may take a few minutes to complete. Refresh this page to see updates.



Join the GitHub Student Developer Pack

Verified students receive free GitHub Pro plus thousands of dollars worth of the best real-world tools and training from GitHub Education partners — for free. [Learn more](#)

Apply

- Aceptada la tarea, nos aparece esta pantalla.
 - Ignoramos el ofrecimiento “Join the GitHub Student Developer Pack”.
- Salimos de esta pantalla haciendo clic en nuestro icono (arriba dcha.).



github-classroom[bot] <noreply@github.com>
Lun 09/08/2021 19:15
Para: Usted



GitHub



@github-classroom[bot] has invited you to
collaborate on the
clases-julio/ejemplo-inicial-tuusername repository

You can [accept or decline](#) this invitation. You can also visit [@github-classroom\[bot\]](#) to learn a bit more about them.

This invitation will expire in 7 days.

- Tras lo anterior, recibiremos uno o dos correos de GitHub-Classroom.
 - Siempre: la invitación a colaborar en el repositorio de bienvenida.
 - MUY IMPORTANTE: clic en link “accept or decline” the invitation.
 - Según el caso: aviso de creación de un Pull Request (PR) automático.
 - PR: para solicitar modificaciones en el código en entorno colaborativo.
 - De haber PR, se usará como canal de comunicación de cada repositorio.



Repositorios de prácticas de las clases de Julio Vega

Repositorios de prácticas de las clases de Julio Vega

Spain <https://gsyc.urjc.es/jmvega> @JulioV3ga julio.vega@urjc.es

Overview Repositories 1 Packages People Projects

Popular repositories

clases-julio doesn't have any public repositories yet.

Repositories

Find a repository...

Type

Sort

p1-tuusername

Private

p1-tuusername created by GitHub Classroom

People

This organization has no public members. You must be a member to see who's a part of this organization.

Report abuse

● Resumen:

- Ya nos hemos creado una cuenta en GitHub con correo de URJC.
- Ya hemos enlazado nuestra cuenta GitHub con GitHub-Classroom.
 - Para lo cual, hemos aceptado invitación a la tarea de bienvenida.
- Y ahora, para acceder a esta y a todas las tareas durante el curso:
 - Entramos en *clases-julio*: <https://github.com/clases-julio>
 - OJO: ¡no aparecen en nuestra cuenta, porque son tareas compartidas!
 - Hacemos clic en la tarea en cuestión, e.g. *p1-tuusername*.
 - En la siguiente sección vemos cómo trabajar con la tarea/repositorio.

Repetimos los tres pasos vistos en Sección 2

- En sesión de prácticas, se os enviará invitación de nueva práctica.
 - Recibiréis correo con esta invitación vía *Novedades* del Aula Virtual.
- Una tarea = una práctica = un repositorio de trabajo para el código.

git clone: obtengo una copia del repo, de nube a PC 1

```
> git clone https://github.com/clases-julio/github-starter-course.git
Clonando en 'github-starter-course'...
Username for 'https://github.com': jmvega
Password for 'https://jmvega@github.com':
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 2), reused 2 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (6/6), 5.96 KiB | 2.98 MiB/s, listo.
> ls
github-starter-course
```

- **git clone https://github.com/clases-julio/nombre-del-repo.git**
- Introduzco username y token (OJO, no la contraseña, ¡el token!).
 - Puedo usar *caching* para evitar poner el token reiteradamente:
 - `git config --global credential.helper 'cache --timeout=3600'`
 - Así, al meter el token por 1ª, se guardará 1 hora (3600s.) en caché.
- Y ya tengo una copia del repo en mi ordenador. Es una carpeta más.
- OJO: *clone* solo es para descargar el repo por 1.ª vez.
 - Las siguientes veces, actualizo copia PC 1 mediante **git pull**.

git status: compruebo el estado de mi copia

```
> cd github-starter-course/  
> git status  
En la rama main  
Tu rama está actualizada con 'origin/main'.  
  
nada para hacer commit, el árbol de trabajo está limpio  
> touch mi_practica.txt  
> git status  
En la rama main  
Tu rama está actualizada con 'origin/main'.  
  
Archivos sin seguimiento:  
  (usa "git add <archivo>..." para incluirlo a lo que se será confirmado)  
    mi_practica.txt  
  
no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa  
"git add" para hacerles seguimiento)
```

- Entro dentro de la carpeta que me acabo de descargar.
- **git status**
 - Me fijo en mensaje *nada para hacer commit* \implies Todo OK.
 - Creo un fichero de ejemplo: *mi_practica.txt*.
- Nuevamente ejecuto *git status*
 - Ahora el mensaje es *Archivos sin seguimiento: usa git add....*

git add, git commit y git push: actualizar nube

```
> git add *
> git status
En la rama main
Tu rama está actualizada con 'origin/main'.

Cambios a ser confirmados:
  (usa "git restore --staged <archivo>..." para sacar del área de stage)
    nuevo archivo:  mi_practica.txt

> git commit -m "he añadido un nuevo archivo"
[main be78c09] he añadido un nuevo archivo
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 mi_practica.txt
> git push
Username for 'https://github.com': jmvega
Password for 'https://jmvega@github.com':
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 6 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 300 bytes | 300.00 KiB/s, listo.
Total 3 (delta 0), reusado 0 (delta 0)
To https://github.com/clases-julio/github-starter-course.git
 d878978..be78c09  main -> main
```

- **git add ***: así añadido todo lo nuevo que tenga en local.
- **git status**: veo que los cambios están por confirmar aún.
- **git commit -m**: confirmo cambios (pero aún están en local).
- **git push**: subo los cambios confirmados a la nube.
 - La nube estará up-to-date con el nuevo fichero *mi_practica.txt*.
 - Y esto es un commit: un avance en la implementación de tu práctica.
 - Para evaluación de prácticas jestos commits son muy importantes!

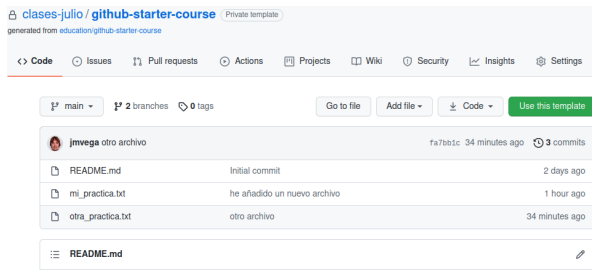
Situación: trabajo en otro puesto, PC 2, y vuelvo a PC 1

- Repito paso 2.
 - Resumen: si es 1ª vez, **git clone**, si no, **git pull**.
- Repito pasos 3 y 4: trabajo en un nuevo fichero, *otra_practica.txt*.
- La nube estará up-to-date con el nuevo fichero *otra_practica.txt*.
 - De modo que PC 2 y nube están up-to-date. ¡Pero en PC 1 no lo tengo!

```
> ls
mi_practica.txt  README.md
> git pull
Username for 'https://github.com': jmvega
Password for 'https://jmvega@github.com':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 2 (delta 0), reused 2 (delta 0), pack-reused 0
Desempaquetando objetos: 100% (2/2), 264 bytes | 264.00 KiB/s, listo.
Desde https://github.com/clases-julio/github-starter-course
   be78c09..fa7bb1c  main      -> origin/main
Actualizando be78c09..fa7bb1c
Fast-forward
  otra_practica.txt | 0
  1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 otra_practica.txt
> ls
mi_practica.txt  otra_practica.txt  README.md
```

- Cuando vuelva a trabajar a PC 1, solo tendré *mi_practica.txt*.
 - Para actualizar PC 1, recuerda ejecutar **git pull** y listo.
- Corolario: recuerda los pasos: *pull*, trabajo..., *commit* y *push*.

Uso del navegador para gestionar el repositorio



- En cualquier momento puedo acceder a GitHub.com con el navegador.
- Entorno web muy útil para hacer varias gestiones del repositorio.
 - Por ejemplo, simplemente, visualizarlo de forma gráfica.
 - Y mucho +: descargar, incidencias (issues), PRs (pull requests), wiki.
- Por defecto, se visualiza el archivo *README.md* del repo.
 - Este es el archivo de presentación de cualquier repo.
 - Extensión *md*: escrito en lenguaje MarkDown. Muy sencillo.
 - Se puede decir que *README.md* de repo = *index.html* de página web.
 - No es obligatorio crearlo/editarlo pero sí altamente recomendable.

- Desde 13/08/21 necesario activar *factor doble autenticación* (2FA).
 - Viene de la directiva europea PSD2. Te habrá afectado en tu banco.
 - Entrar en la cuenta \implies clic en icono avatar \implies Settings \implies Account security \implies Enable two-factor authentication.
 - Se puede habilitar doble autenticación mediante App GitHub o SMS.
 - Recibirás código de verificación y guardar *recovery codes*.
 - Esta tabla de códigos es similar a tabla de coordenadas del banco.
 - Si quieres más info sobre el acceso mediante 2FA [*pincha aquí*](#)
- Con eso es suficiente para acceder vía 2FA mediante navegador.
- Para trabajar línea de comandos (SSH o HTTPS) se necesitan tokens.
- Contraseña Git obsoleto \implies usar token de acceso personal (PAT).
 - PAT = contraseña, pero mucho más seguro; imposible de descifrar.
 - Desde la cuenta \implies clic en avatar \implies Settings \implies Developer settings \implies Personal access tokens \implies Generate new token.
 - Rellenamos campo *Note*: e.g. *Trabajar desde Terminal*.
 - Concedemos acceso a todo y *Generate token*.
 - Una vez generado \implies MUY IMPORTANTE: ¡seleccionar y guardar!
 - Si quieres más info sobre cómo generar el token [*pincha aquí*](#)

0.2 Uso de GitHub

Julio Vega

julio.vega@urjc.es

