

Ejercicio práctico

Prerequisites

- **Instalar Git:** <https://git-scm.com/>
- **Instalar R:** <https://www.r-project.org/>
- **Instalar RStudio:** <https://posit.co/>

- **Crear perfil en GitHub:** <https://github.com>






CONSEJO: Elegir un nombre de usuario sencillo, recomendable todo minúsculas sin espacios ni caracteres especiales. Si venimos usando un mismo nombre en redes sociales u otras plataformas, es bueno usar el mismo para ser identificables.



- **Vincular RStudio y GitHub:**

Necesitamos que RStudio, Git y Github se comuniquen.

1. Creamos un token personal en GitHub.

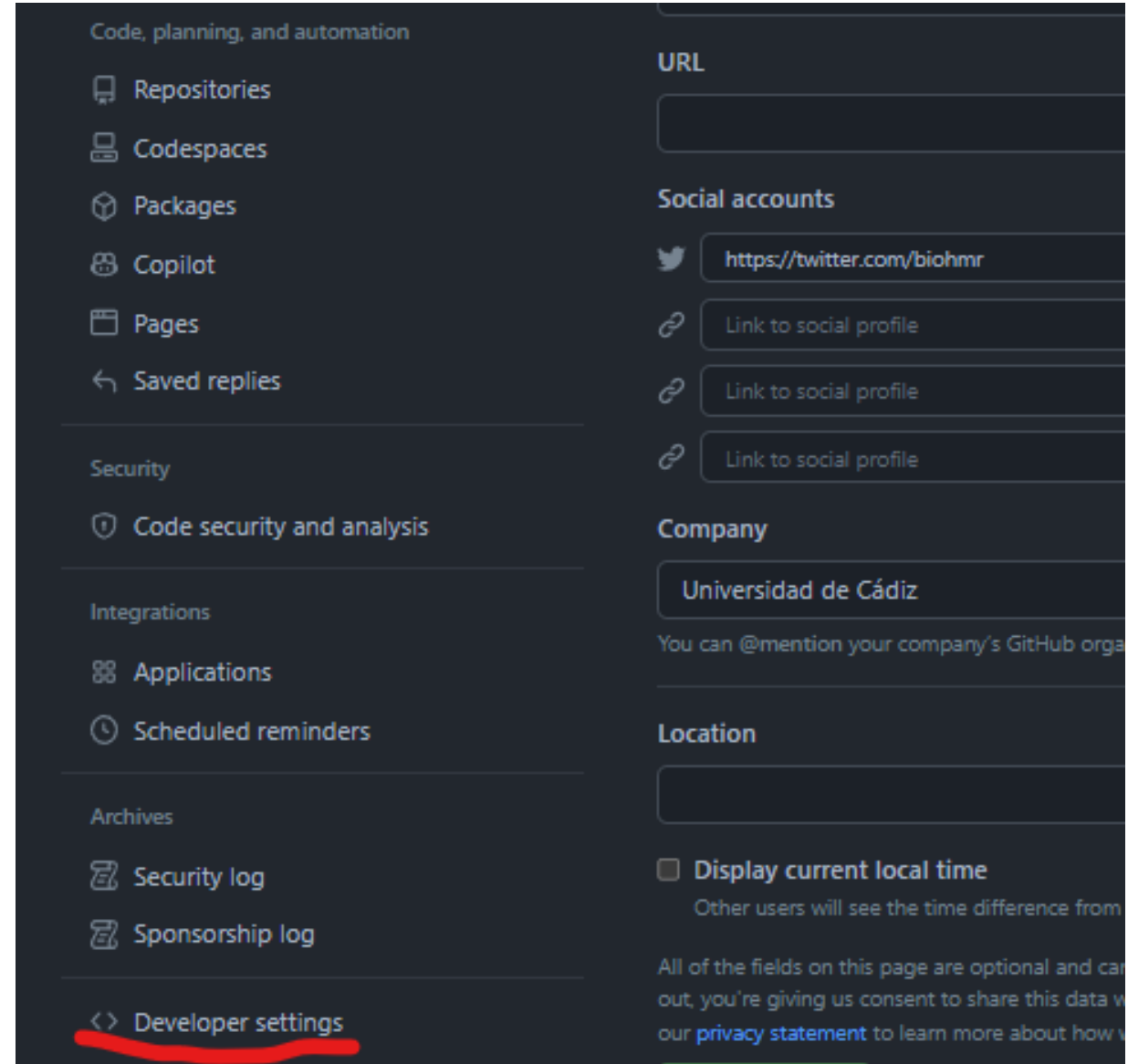
Profile -> Settings

-  Upgrade
-  Try Enterprise
-  Try Copilot
-  Feature preview
-  **Settings**

-  GitHub Docs
-  GitHub Support

Sign out

Developer Settings



The screenshot shows the GitHub Developer Settings interface. On the left, a sidebar lists various settings categories: 'Code, planning, and automation' (with sub-items: Repositories, Codespaces, Packages, Copilot, Pages, Saved replies), 'Security' (with sub-item: Code security and analysis), 'Integrations' (with sub-items: Applications, Scheduled reminders), 'Archives' (with sub-items: Security log, Sponsorship log), and 'Developer settings' (highlighted with a red underline). The main content area on the right is currently empty, showing only the section headers: 'URL', 'Social accounts' (with a Twitter link and three generic 'Link to social profile' buttons), 'Company' (with 'Universidad de Cádiz' and a description), 'Location', and 'Display current local time' (with a checkbox and a note about time differences). At the bottom right, a disclaimer states that all fields are optional and that using them implies consent to share data, with a link to the 'privacy statement'.

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

URL

Social accounts

-  <https://twitter.com/biohmr>
-  [Link to social profile](#)
-  [Link to social profile](#)
-  [Link to social profile](#)

Company

Universidad de Cádiz






You can @mention your company's GitHub orga

Location

☐ Display current local time

Other users will see the time difference from

All of the fields on this page are optional and can be turned off. If you turn them out, you're giving us consent to share this data with our [privacy statement](#) to learn more about how we

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

2. Presentarnos a Git:

Definimos nuestro nombre de usuario:

```
git config --global user.name 'Nombre Apellido'
```

Definimos nuestro email:

```
git config --global user.email 'email'
```

Comprobamos que todo es correcto:

```
git config --global --list
```


Crear un repositorio

Un repositorio es la colección de carpetas y archivos correspondiente a un proyecto. Debemos elegir:

1. Nombre. Corto pero descriptivo.
2. Público o privado.
3. README. Archivo donde explicamos en qué consiste el proyecto y describimos su contenido.
4. Licencia.



María Hurtado

MariaHdMR

PhD student at the Universidad de Cádiz (Spain). Interested in Community ecology and species coexistence.

Find a repository...

Type ▾

Language ▾

Sort ▾

 **New**

[herbivores_pathogen_ucsc](#) Private

● R Updated 20 hours ago

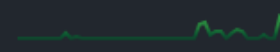
☆ Star ▾



[Multitrophic plants](#) Private

● R Updated 5 days ago

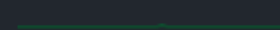
☆ Star ▾



[TFM_article_published](#) Private

● R Updated on Apr 10

☆ Star ▾



[TFM](#) Private

☆ Star ▾

Create a new repository

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

Required fields are marked with an asterisk (*).

Owner *

 MariaHdMR ▾

Repository name *

/

Great repository names are short and memorable. Need inspiration? How about **curly-octo-tribble** ?

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:



Add a README file

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

Add .gitignore


.gitignore template: **None** ▾

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

Choose a license

License: **None** ▾

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

 You are creating a public repository in your personal account.

Clonar el repositorio desde RStudio

File > New Project > Version Control > Git

Copiar la url de nuestro repositorio de GitHub. Se genera una carpeta en local, con el contenido del repositorio.

Se nos pedirá que nos autentiquemos (token, vía navegador, etc.), para demostrar que somos propietarios del repositorio.

¡Ya tenemos nuestro proyecto sincronizado en Rstudio y Github!

Commit, push y pull

En Rstudio tenemos en el panel superior derecha una pestaña llamada "Git". Aquí nos aparecerán aquellos archivos que han sido modificados (M), y cuyos cambios no han sido aún incluidos en el control de versión.

reproducible_uca23 - main - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function

Source Visual

```
40
41
42 Con 'git config user.email' vamos a ver que lista de correo es la asociada a Git y comprobar que todo es correcto.
43
44 ## Crear el proyecto
45
46 3.El siguiente paso es crear un repositorio desde el perfil de Github. Para ello tenemos que seguir los siguientes pasos: un repositorio es un
47 lugar en el que vamos a almacenar un proyecto y al crearlo tenemos que definir las características que queremos que vaya a tener. Se aconseja
48 crear un archivo README para explicar lo que se va a hacer en ese proyecto.
49
50 ![bg center 50%](img/crear_repositorio/git_repositorio1.png)![bg center 50%](img/crear_repositorio/git_repositorio2.png)
51 Este repositorio que hemos creado tenemos que clonarlo en Rstudio, para ello tenemos que ir a la pestaña file \> new project \> version control \>
52 Git y copiar ahí el url del proyecto que teníamos online. Esto nos va a generar una carpeta en local, donde tendremos todo lo relacionado con el
53 proyecto y que además podemos subir a la nub mediante github.
54
55 ¡Ya tenemos el proyecto en Rstudio y Github!
56
57 #Commit, push y pull
58 En Rstudio, en la configuración de paneles que aparece por defecto, en el panel superior derecha aparece una pestaña que pone "Git".
59 Si clicais en esa pestaña aparecen los archivos que hay en las carpetas, y aparecen con na M en azul cuando es un archivo que se ha
60 modificado y guardado en local pero que no se a subido al proyecto de la nube.
61
62 Siempre que compartamos un repositorio con colaboradores tendremos que hacerr Pull (vamos a descargar todos los cambios que los colaboradores han
63 hecho), una vez que hemos hecho pull, modificamos nuestros análisis. Tras modificar los análisis y decidir que son correctos y querer guardarlos,
64 hacemos commit y tenemos que describir brevemente lo que hemos hecho para que posteriormente si hay un error sea fácil de saber dónde se pudo
65 producir el error. Una vez hemos hecho el commit, tendremos que hacer Push para que esos cambios se suban a la nube.
66
67
68 #ejercicio 1
69 Poneros por parejas. Cada uno de la pareja va a crear un pequeño código (contadnos que habéis desayunado) o sube el logo de la uca a la carpeta
70 del repositorio.
71 crea un pequeño código donde ponga tu nombre y una línea de en qué trabajas en la UCA, o sube una foto del logo de la UCA al repositorio
72
73 -----
74
75 ## Analizar los datos
76
77 - crear un script de R para analizar los datos y crear los resultados deseados.
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Environment History Connections Git Tutorial

Diff Commit Pull Push

Staged Status Path

- ?.?. .Rhistory
- M .gitignore
- ?.?. reproducible_uca23.Rproj
- ?.?. ppt/img/crear_repositorio/
- ?.?. ppt/img/vincular_git_rstudio/

Files Plots Packages Help Viewer Presentation

New Folder New Blank File Delete Rename More

C:\Users\maria\Documents\R_repositorios\reproducible_uca23

Name	Size	Modified
..		
.gitignore	13 B	Sep 6, 2023, 12:34 PM
.Rhistory	230 B	Sep 26, 2023, 1:03 PM
data		
Informe.html	1.3 MB	Sep 26, 2023, 12:06 PM
Informe.Rmd	4.7 KB	Sep 26, 2023, 12:06 PM
LICENSE	1.1 KB	Sep 6, 2023, 12:34 PM
ppt		
R		
README.md	1.9 KB	Sep 22, 2023, 12:55 PM
reproducible_uca23.Rproj	218 B	Sep 26, 2023, 4:02 PM
results		

Console Terminal Background Jobs

R 4.2.2 · C:\Users\maria\Documents\R_repositorios\reproducible_uca23/

You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |

El flujo de trabajo habitual debe ser:

1. Hacer `pull` para sincronizar nuestra versión local con la última versión de GitHub.
2. Modificamos nuestros archivos.
3. Hacemos control de versión local: `add + commit`.
4. Subimos nuestros cambios a GitHub: `push`.

Changes History main ☒ Stage Revert Ignore Pull Push

Your branch is ahead of 'origin/main' by 1 commit.

Staged	Status	Path
<input type="checkbox"/>	??	.Rhistory
<input type="checkbox"/>	??	reproducible_uca23.Rproj
<input checked="" type="checkbox"/>	M	ppt/2 - Ejercicio practico.md

Commit message

☐ Amend previous commit

Commit

Show ☒ Staged ☐ Unstaged Context 5 lines ☐ Ignore Whitespace Unstage All

@@ -22,11 +22,11 @@ paginate: false

Unstage chunk

22	22	
23	23	- Instalar RStudio: < https://posit.co/download/rstudio-desktop/ >
24	24	
25	25	-----
26	26	
27		- Crear perfil en GitHub: Lo primero de todo, vamos a tener que crearnos una cuenta en github (< https://github.com/ >)
	27	- Crear perfil en GitHub: Lo primero de todo, vamos a tener que crearnos una cuenta en github (< https://github.com/ >). CONSEJO: Crearos un nombre que ya useis en otro sitio, por ejemplo, Twitter. Que sea un nombre de usuario sencillo, recomendable todo en minúsculas.
28	28	
29	29	-----
30	30	
31	31	- Vincular RStudio y GitHub:

Ejercicio 1

1. Vamos a hacer un sencillo análisis usando un conjunto de datos de éxitos de Spotify 2010-2019.
2. Una vez creado el script, hacer `add` + `commit` + `push` para grabar los cambios en local y en GitHub.

La he liado, ¿cómo vuelvo atrás?

Si he hecho un cambio que deseo eliminar, hay múltiples opciones:

- `restore`: deshace un `add` y/o los cambios del directorio de trabajo.
- `reset`: deshace un `commit` y/o un `git add` y/o los cambios del directorio de trabajo.
- `revert HEAD`: La opción segura de `git reset` para deshacer un commit. Deshace commit pero deja constancia de que se ha deshecho un cambio.

Ejercicio 2

1. Vamos a actualizar el README de nuestro repositorio para contar lo que hemos hecho analizando datos de Spotify.
2. Hacer un commit de los cambios y hacer `git reset --soft HEAD~1`. ¿Qué ha pasado?
3. Hacemos el commit otra vez y hacer ahora `git reset --mixed HEAD~1`. ¿Qué ha pasado ahora?
4. Hacemos el commit de nuevo y hacer `git reset --hard HEAD~1`. ¿Y ahora?

¿Cómo colaborar?

Podemos invitar a colaboradores a nuestro repositorio.

En nuestro repositorio: Settings > Access - Collaborators > Add people

Añadimos el nombre de usuario de nuestro compañero/a.

Aceptamos invitación en Profile > Your Organizations

Ejercicio 2

1. En nuestro repositorio, añadimos como colaborador a la persona a nuestro lado.
2. Clonamos en RStudio el proyecto al que se nos ha invitado.
3. Hacemos un cambio en el script de R, cambiamos la línea de tendencia a roja, por ejemplo.
4. `add` + `commit` + `push`
5. Ver los cambios en nuestro repositorio en GitHub, `pull` para descargar los cambios en local.

Manuscritos reproducibles

Vamos a trasladar nuestro script sencillo en R a un archivo markdown (.Rmd), que genera un archivo de texto con código y resultados integrados, para completar la reproducibilidad del proyecto.

El último paso, archivar

Vamos a crear una cuenta en Zenodo (<https://zenodo.org/>) y subimos la carpeta de nuestro proyecto con un DOI (enlace permanente y citable).