
MANUAL DE INSTALACIÓN Y USO DE GITHUB

ÍNDICE

1. INTRODUCCIÓN.	2
2. PREPARACIÓN.	2
2.1 Descarga de Git.	2
2.2 Creación de cuenta en la plataforma GitHub.	3
3. CREACIÓN DEL REPOSITORIO.	4
4. EMPAREJAMIENTO LOCAL/GITHUB.	6
5. ACTUALIZACIÓN DEL DIRECTORIO LOCAL.	7

1. INTRODUCCIÓN.

En este manual se explicarán los principales pasos para trabajar con repositorios en línea alojados en GitHub.

GitHub (y su programa asociado Git) es una plataforma de desarrollo colaborativo de código abierto que nos permite trabajar con archivos compartidos en repositorios alojados en la nube y poder acceder a ellos desde cualquier terminal. Permite a su vez llevar a cabo un control de versiones de estos archivos. Para poder establecer una conexión desde un ordenador local al servidor de GitHub, utilizaremos el programa Git.

2. PREPARACIÓN.

2.1 Descarga de Git.

Git es un programa de control de sistemas libre y de código abierto, diseñado para gestionar todo tipo de proyectos, especialmente cuando estos se llevan a cabo de manera coordinada entre más de un desarrollador. La función de este software es establecer comunicación entre el host y el servidor de GitHub.

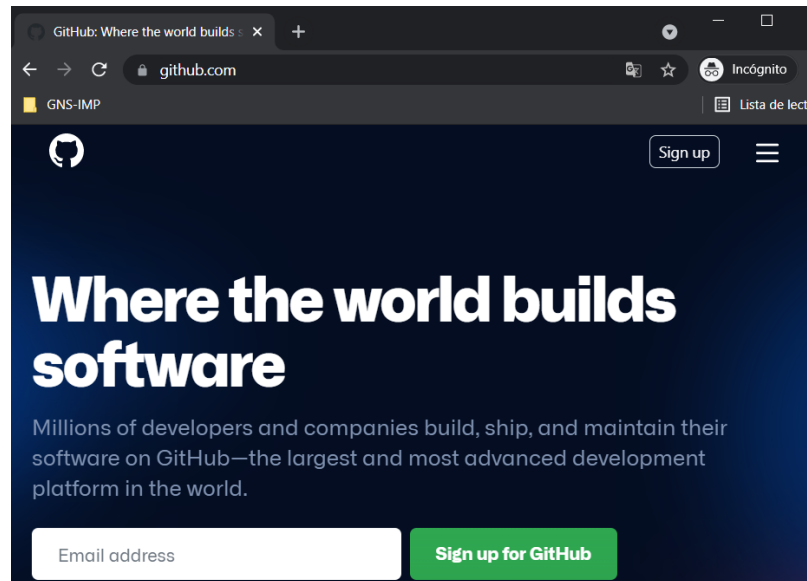
Para descargar este programa nos debemos dirigir a la dirección <https://git-scm.com/downloads>, en la cual podremos descargar el archivo ejecutable para realizar la instalación de la última versión del programa, que en nuestro caso es la 2.33.0 para el sistema operativo Windows.



Cuando finalice la descarga, clicando el ejecutable se iniciará automáticamente el proceso de instalación. Antes de comenzar la instalación de archivos propiamente dicha, el programa nos dejará cambiar ciertas configuraciones de la instalación. En nuestro caso, nos limitaremos a instalarlo con la configuración por defecto.

2.2 Creación de cuenta en la plataforma GitHub.

Para poder crear repositorios en el servidor de GitHub es necesario tener una cuenta en su plataforma, a la cual podemos acceder desde la dirección web <https://github.com/>.



Una vez ya en la página, debemos registrarnos. Para esto, clicaremos en *Sign Up* (Registrarse). Aparecerá un formulario paso por paso como el que se muestra debajo, el cual habremos de cumplimentar con nuestros datos.

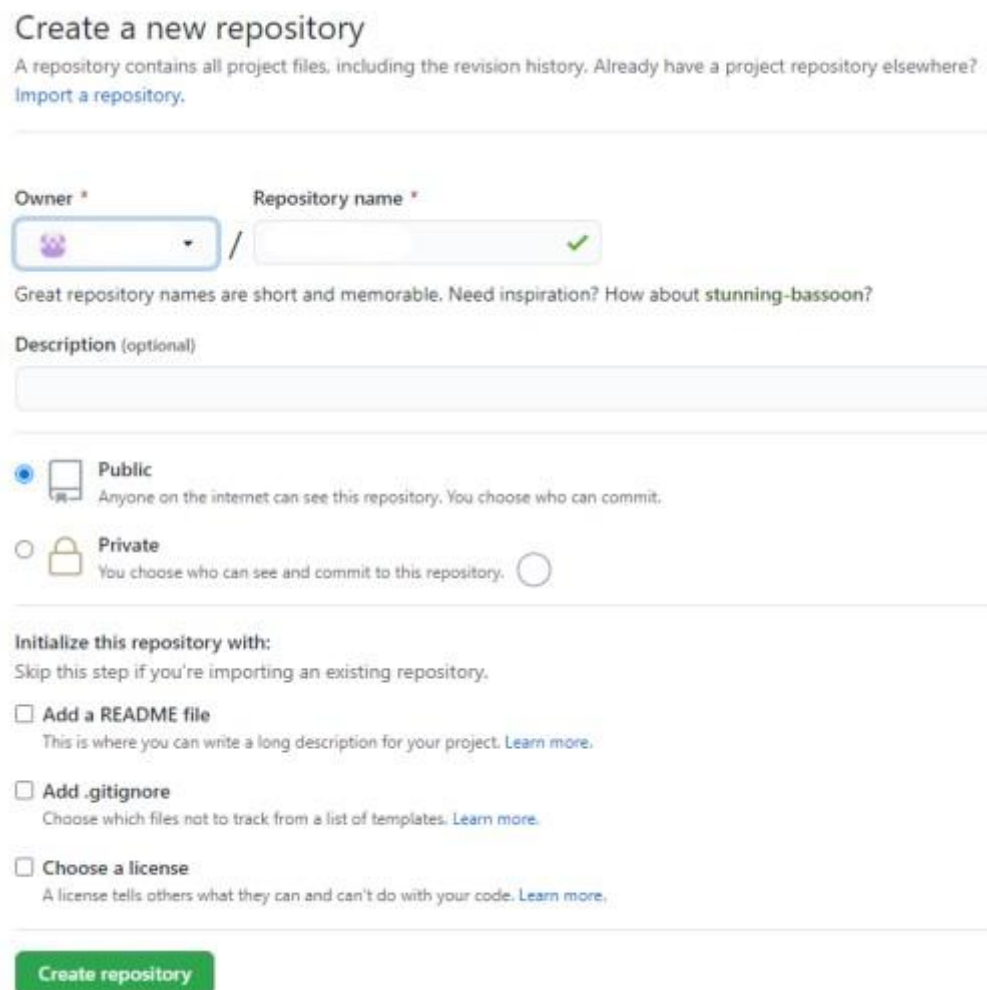
A screenshot of the GitHub registration form, which has a dark blue background with white and green text. The form starts with a welcome message: 'Welcome to GitHub! Let's begin the adventure'. It then presents three steps, each with a green checkmark: 'Enter your email', 'Create a password', and 'Enter a username'. The final step is a question: 'Would you like to receive product updates and announcements via email? Type "y" for yes or "n" for no'. At the bottom right, there is a green button labeled 'Continue'.

Una vez hecho esto y tras la verificación pertinente, ya tendremos nuestra cuenta lista.

3. CREACIÓN DEL REPOSITORIO.

El siguiente paso es crear nuestro primer repositorio. Un repositorio es un espacio centralizado donde se almacena, organiza, mantiene y difunde información digital, y en este caso esta información será todo el contenido de esta práctica, al igual que se hará para todas y cada una de las prácticas de este año en sendos repositorios.

Para crear un repositorio en la plataforma GitHub, el primer paso será clicar en *New/New Repository* (puede aparecer de ambas formas), el cual nos llevará al formulario mediante el cual se creará el repositorio, que tiene el siguiente aspecto:



The screenshot shows the 'Create a new repository' page on GitHub. At the top, it says 'Create a new repository' and provides a brief explanation: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. Below this, there are two input fields: 'Owner' with a dropdown menu showing a user icon, and 'Repository name' with a text input field and a green checkmark. A hint below these fields says: 'Great repository names are short and memorable. Need inspiration? How about `stunning-bassoon`?'. There is a 'Description (optional)' text area below. Then, there are two radio button options: 'Public' (selected) with a lock icon and the text 'Anyone on the internet can see this repository. You choose who can commit.', and 'Private' with an unlocked lock icon and the text 'You choose who can see and commit to this repository.'. Below these are three checkboxes under the heading 'Initialize this repository with:'. The first is 'Add a README file' with the subtext 'This is where you can write a long description for your project. [Learn more.](#)'. The second is 'Add .gitignore' with the subtext 'Choose which files not to track from a list of templates. [Learn more.](#)'. The third is 'Choose a license' with the subtext 'A license tells others what they can and can't do with your code. [Learn more.](#)'. At the bottom, there is a green button labeled 'Create repository'.

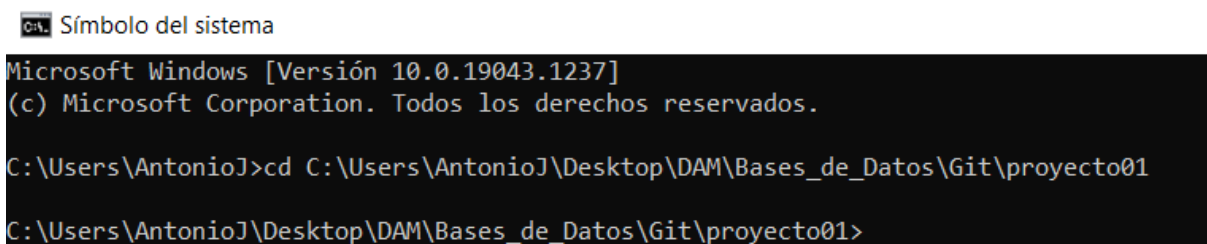
En él podremos, entre otras configuraciones, seleccionar el nombre del repositorio (que en este caso será *proyecto01*), si queremos que sea público o privado, o dotarlo de una descripción. Una vez elegido todo esto, bastará con clicar en *Create repository* para finalizar el proceso y que se cree el repositorio.

4. EMPAREJAMIENTO LOCAL/GITHUB.

Una vez creado el repositorio en la nube, es el momento de emparejarlo con un directorio local para poder compartir nuestros archivos.

Para ello, creamos un directorio con el mismo nombre en nuestro ordenador, e introducimos en él todas las subcarpetas y/o archivos que queramos enviar a la nube (al repositorio de GitHub). Una vez hecho esto, abrimos la consola de comandos de Windows. Una forma de acceder a ella es escribiendo “cmd” en la barra de búsqueda de Windows, aunque no es la única.

Una vez en ella nos situaremos en la carpeta que hemos creado anteriormente, usando para ello el comando “cd” (cambiar directorio), como se muestra en la siguiente imagen:



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19043.1237]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\AntonioJ>cd C:\Users\AntonioJ\Desktop\DAM\Bases_de_Datos\Git\proyecto01
C:\Users\AntonioJ\Desktop\DAM\Bases_de_Datos\Git\proyecto01>
```

Una vez en ella, escribimos la siguiente secuencia de comandos:

- *git init*: con este comando arrancamos el programa Git, y seleccionamos la carpeta como repositorio Git.
- *git add*: este comando nos permite marcar cuáles son los elementos que se quieren añadir al repositorio en línea. Como en este caso queremos añadir todo, usaremos el comando “*git add .*”, donde el punto significa todo aquello que se encuentre en el directorio seleccionado.
- *git commit -m*: este comando nos permite marcar los cambios que se van a realizar como un solo cambio, al cual se le puede añadir una etiqueta para distinguirlo de futuros cambios.
- *git remote add origin https://...:* este comando es el que marca dónde vamos a añadir los elementos en la nube. En este caso, pondremos la dirección de nuestro propio repositorio.
- *git push -u origin master*: este comando es el que envía en sí el directorio al repositorio en línea. Al hacer esto por primera vez, nos saldrá una ventana emergente para autenticarnos en GitHub.

Fuera de estos comandos, podemos comprobar en cualquier momento el estado de la subida usando el comando *git status*.

```

C:\Users\AntonioJ\Desktop\Bases_de_Datos\Git\proyecto01>git init
Initialized empty Git repository in C:/Users/AntonioJ/Desktop/Bases_de_Datos/Git/proyecto01/.git/
C:\Users\AntonioJ\Desktop\Bases_de_Datos\Git\proyecto01> git commit -m "first commit"
Author identity unknown

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.
fatal: unable to auto-detect email address (got 'AntonioJ@DESKTOP-FVPTVSA.(none)')
C:\Users\AntonioJ\Desktop\Bases_de_Datos\Git\proyecto01>git config --global user.email "avilpinb@educand.es"
C:\Users\AntonioJ\Desktop\Bases_de_Datos\Git\proyecto01>git config --global user.name "villanuevapinoantonio"
C:\Users\AntonioJ\Desktop\Bases_de_Datos\Git\proyecto01>git add .
C:\Users\AntonioJ\Desktop\Bases_de_Datos\Git\proyecto01>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   readme.md

C:\Users\AntonioJ\Desktop\Bases_de_Datos\Git\proyecto01>git add .
C:\Users\AntonioJ\Desktop\Bases_de_Datos\Git\proyecto01>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   doc/instrucciones.txt
        new file:   readme.md
        new file:   src/programa.txt






C:\Users\AntonioJ\Desktop\Bases_de_Datos\Git\proyecto01> git commit -m "Primer commit"
[master (root-commit) 87eab1d] Primer commit
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 doc/instrucciones.txt
 create mode 100644 readme.md
 create mode 100644 src/programa.txt

C:\Users\AntonioJ\Desktop\Bases_de_Datos\Git\proyecto01>
-

C:\Users\AntonioJ\Desktop\Bases_de_Datos\Git\proyecto01> git push -u origin main
info: please complete authentication in your browser..
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 568 bytes | 284.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/villanuevapinoantonio/proyecto01.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.

```

Tras esto, podemos comprobar en nuestro repositorio de GitHub si el archivo (o los archivos) se ha subido correctamente.

	villanuevapinoantonio Actualización README.md	69a623f 3 days ago	 5 commits
	doc	Primer commit	3 days ago
	src	Primer commit	3 days ago
	readme.md	Actualización README.md	3 days ago

5. ACTUALIZACIÓN DEL DIRECTORIO LOCAL.

Ya que el contenido del repositorio puede ser modificado por otros usuarios, o por ti mismo en otro sistema, a veces es necesario actualizar tu directorio local con los cambios realizados en el repositorio, para esto, simplemente habrá que utilizar el comando *git pull -u origin master* después de haber posicionado la consola en nuestro directorio local.

En caso de que no tengamos un directorio local, es posible clonar el repositorio de la nube en un nuevo directorio local. Para esto, usaremos el comando *git clone https://...*, donde la dirección web será la de nuestro repositorio en GitHub.