



¿Qué es Git?

Es un sistema de control de versiones, es distribuido, es decir que múltiples personas pueden trabajar en equipo, es open source y también se adapta a todo tipo de proyectos desde pequeños hasta grandes, además, se pueden fusionar archivos, guarda una línea de tiempo a lo largo de todo el proyecto. Maneja una interfaz tipo Bash. <u>GIT, es el software de control de versiones en el que se basa Github.</u>

♦ MINGW64:/d	-	×
DELL@DESKTOP-R21Q3R8 MINGW64 /d		\sim
3		
		\sim

Sitio de descarga: <u>https://git-scm.com/downloads</u> Como instalar Git: <u>https://www.youtube.com/watch?v=ExdLS6lZaAY</u>

¿Qué es Github?

Hechos

QUE

CONECTAN 🗸

A diferencia de Git, Github es un <u>sitio web y un servicio en la nube</u> que ayuda a los desarrolladores a almacenar y administrar su código, al igual que <u>llevar un registro y control de cualquier cambio sobre</u> <u>este código</u>. En otras palabras, es una plataforma de desarrollo colaborativo, o también llamada la red social de los desarrolladores donde se alojan los repositorios, el código se almacena de forma pública pero se puede hacer privado con una cuenta de pago.

La interfaz de GitHub es bastante fácil de usar para el desarrollador novato que quiera aprovechar las ventajas del Git. Sin GitHub, usar un Git generalmente requiere de un poco más de conoci de tecnología y uso de una línea de comando (Bash).







Sitio de descarga: <u>https://desktop.github.com/</u> Como instalar Github: <u>https://www.youtube.com/watch?v=tn6tloweTUs</u>

¿Qué Es una Versión de Control?

C2022[°]

Una Versión de Control ayuda a los desarrolladores a llevar un registro y administrar cualquier cambio en el código del proyecto de software. A medida que crece este proyecto, la versión de control se vuelve esencial.

Con la *bifurcación*, un desarrollador duplica parte del código fuente (llamado repositorio). Este desarrollador, luego puede, de forma segura, hacer cambios a esa parte del código, sin afectar al resto del proyecto.

Luego, una vez que el desarrollador logre que su parte del código funcione de forma apropiada, esta persona podría *fusionar* este código al código fuente principal para hacerlo oficial. Todos estos cambios luego son registrados y pueden ser revertidos si es necesario.

Documentación de Github: <u>https://docs.github.com/es/github</u> Documentación Git: <u>https://git-scm.com/book/es/v2</u>

Creación de una cuenta

Lo primero que necesitas es una cuenta de usuario gratuita. Simp visita <u>https://github.com</u>, elige un nombre de usuario que no esté ya en uso, proporciona y una contraseña, y pulsa el botón verde grande "Sign up for GitHub".







Pick a username Your email Create a password Use at least one lowercase letter, one numeral, and seven characters.
Pick a username Your email Create a password
Pick a username Your email
Pick a username

Lo siguiente que verás es la página de precios para planes mejores, pero lo puedes ignorar por el momento. GitHub te enviará un correo para verificar la dirección que les has dado. Confirmar la dirección ahora, es bastante importante (como veremos después).

Para ampliar esta información: https://n9.cl/nqu9

Crear un repositorio

QUI

Un **repositorio** se usa generalmente para organizar un solo proyecto. Los repositorios pueden contener carpetas y archivos, imágenes, videos, hojas de cálculo y conjuntos de datos, cualquier cosa que su proyecto necesite. Recomendamos incluir un archivo *README* o un archivo con información sobre su proyecto. GitHub facilita agregar uno al mismo tiempo que crea su nuevo repositorio. *También ofrece otras opciones comunes, como un archivo de licencia.*

Paso 1. Crear un nuevo repositorio

1. En la esquina superior derecha, junto a tu avatar o ícono de identidad, haz clic en (+) y luego seleccione **Nuevo repositorio**.

Search or jump to			
	Pull requests hours Marketplace Explore		o +- 🙃-
epositories		Gith	di for mobile ×
Juisquillermomol /CS5		for A	adroid and iOS.
me50/luisguillermomolero	Learn Git and GitHub without any code!	6	App Store Google Play
] luisguillermomol /Ejercicios_C_Sharp	Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.		
Juisguillermomol /Proyecto_Duvan			
	Read the guide Start a project		
forking with a team? itHub is built for collaboration. Set up an irganization to improve the way your team vorks together, and get access to more features.			
Cruate an organization			
	Discover interesting projects and people to populate your personal news feed.		
	Your news feed helps you keep up with recent activity on repositories you watch and people you follow.		
	Explore GitHub		
	👰 PresTipl The feed shows you events from people you fields and republication you watch.		
0 Nevelo	a tu repositorio estructura-datos	6	
 Nombr Escribe 	una breve descripción.		The





Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Finispuillermomolero Finispuillermomolero Great repository names are short and memorable. Model accuration? How about 1 Description (optional) Repositorio ejerticios materia Estructura de Datos Public Anyone on the internet can see this repository. You choose who can commit. Pivate You choose who can see and commit to this repository. Skin this stan if you're importing an existing repository.	terate-funicular?
Great repository names are short and memorable. Meed incoiration: How about 1 Description (optional Repositorio ejerticios materia Estructura de Datos Image: Public Anyone on the internet can see this repository. You choose who can commit. Image: Private You choose who can see and commit to this repository. Skin this sten if you're importing an existing repository.	terate-funicular?
Description (optional Repositorio ejeracios materia Estructura de Datos Image:	
Repositorio ejerticios materia Estructura de Datos Image: Public Anyone on the internet can see this repository. You choose who can commit. Image: Private You choose who can see and commit to this repository. Skin this sten if you're importing an existing repository.	
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. Skin this stan if you're importing an existing repository.	
Private You choose who can see and commit to this repository. Skin this step if you're importing an existing repository.	
Skin this sten if you're importing an existing repository	
ship this step in you're importing an existing repository.	
Initialize this repository with a README This will let you immediately clone the repository to your computer.	
Add .gitignore: None	
Create r pository	

- 4. Haga clic en Crear repositorio.
- 5. Guarda estos valores debido a que luego lo necesitaras para subir tus aplicaciones desde el Bash de Git a este Github

Code Issues In Pull requests O Actions	回 Projects 田 Wika ① Security 🗠 Insights 穏 Settings	
	Quick setup — if you've done this kind of thing before	
	If y Set up in Deakop or HTTPS SSH https://github.com/luisguillermonolere/setTructura-astos.git Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.	
	or create a new repository on the command line	
	echo "# estructura-datos" >> README.md git init git add REAME.md	٥
	git commit === rist commit git remote add origin http://github.com/luisguillermomolero/estructura-datos.git git push -u origin master	
	or push an existing repository from the command line	
	git remote add origin https://github.com/luisguillermomolero/estructura-datos.git git push -u origin master	
	or import code from another repository You can initialize this repository with code from a Subversion, Mercurial, or TFS project. Import code	
		P
		H
		M
schoe		
20105	#MiciónTIC2022	



QUE



Paso 2. Subir nuestros proyectos a Github

1.- Crear una carpeta en el disco duro C:\ (ruta: C:\estructura-datos) para guardar todos los ejercicios resueltos de la materia estructura de datos.

2.- Crear una carpeta dentro de C: \estructura-datos para cada una de las clases (de hacerse de otro modo no serán evaluadas los ejercicios entregados)

vo Inicio Compa clar al Copiar Pegar o rápido Portapa;	artir Vista Cortar Copiar ruta de acceso Pegar acceso directo peles	Mover Copiar a* a* Eliminar Camb	iar Nueva	luevo elemento 🔻 ácil acceso 👻	Propiedade	Abrir - Modificar Historial Abrir	Seleccionar todo No seleccionar nada Invertir selección Seleccionar	^ (
→ ~ ↑ 📕 > E	ste equipo 🔸 Disco local	(C:) > estructura-datos			0	, Buscar	r en estructura-datos	
Introducción a ^	Nombre		- de l'incacio	n Tipo		Tamaño		
Introducción a	Clase 1	01/0	08/2020 16:17	Carpeta de	archivos			
Profesor Luis N	Clase 2	01/0	08/2020 16:17	Carpeta de	archivos			
Este equipo	Clase 3	01/0	08/2020 16:17	Carpeta de	archivos			
Descargas	Clase 4	01/0	08/2020 16:17	Carpeta de	archivos			
Documents	Clase 5	01/0	08/2020 16:17	Carpeta de	archivos			
Escritorio								
🎫 Imágenes								
Música								
Objetos 3D								
Vídeos								
Disco local (C:)								
👃 Disco local (D:								
Red								
~								

3.- Una vez creada las carpetas de cada clase y luego de haber instalado **Git** y **Github**, hacer <u>clic</u> <u>derecho</u> sobre la carpeta "*estructura-datos*" y luego sobre la opción "**Git Bash Here**".

	Archivo Inicio Compart	tir Vista X Cortar Copiar ruta de acc	Abrir Abrir en ventana nueva	Propiedades	~ O
	Acceso rápido Portapape	Pegar acceso directes	Anclar al Acceso rápido	Abrir Seleccionar	
	← → ~ ↑ 🗐 > Est	e equipo 🔸 Disco I	Ahadin a la lista de Vic Abrir en Visual Studio	V D D Buscar en Disco local (C:)	
	Introducción a ^	Nombre	Git Bash Here	Tamaño	
	Profesor Luis N	\$WINDOWS.~ \$Windows.~W	A Reproducir con VLC Google Drive	de archivos de archivos	
	 Este equipo Descargas Documents 	Archivos de pr Archivos de pr azW6GrAl5ilu8	Analizar en busca de virus Comprobar reputación en KSN Campara Analizaria Analizaria	de archivos de archivos de archivos	
	Escritorio	ESD estructura-dat	Kaspersky Application Advisor Conceder acceso a Restaurar versiones anteriores	de archivos de archivos de archivos	
	Música Objetos 3D	Microsoft MSOCache	Incluir en biblioteca Anclar a Inicio	> de archivos de archivos	
	 Vídeos Disco local (C:) Disco local (D:) 	OculusSetup-E OneDriveTemp	Add to archive Add to "estructura-datos.rar" Compress and email	de archivos de archivos de archivos	$\mathbf{D} \cdot \mathbf{Q}$
	🔹 Red	ProgramData	Compress to "estructura-datos.rar" and email	de archivos	
	18 elementos 1 elemento	seleccionado	Cortar Copiar	ria archivos	
			Crear acceso directo Eliminar Cambiar nombre		
			Propiedades		
cho	S		#MisiónTl	C2022	





MinTIC

Se abrirá la siguiente consola Bash

NINGW64:/c/estructura-datos	_	×
DELL@DESKTOP-R21Q3R8 MINGW64 /c/estructura-datos \$		\sim

4.- Escribir la siguiente línea de comando para iniciar el **Git** en esa carpeta. **(ESTO SOLO SE HACE UNA VEZ)**







A continuación, todo lo que agregue dentro de esa carpeta estará dentro del **Git** y se podrá sincronizar en la nube. Asimismo, dentro de esa carpeta se creará una carpeta llamada. git <u>que esta</u> <u>oculta</u>.

ndar al copiar Pegar so rápido	X Cortar Copiar ruta de acceso Pegar acceso directo peles	Mover Copiar a * a * Eliminar Cambiar organizar	Nuev Nueva carpeta Nuev	o elemento * acceso *	Propiedade •	Abrir -	Seleccionar todo	
→ • ↑ 📕 > E	ste equipo 🔸 Disco local	(C:) > estructura-datos >			v ي	, Buscar	en estructura-datos	
📕 Introducción a ^	Nombre	 Fecha c 	e modificación	Tipo		Tamaño		
Introducción a	.ait	/08/2	020 16:40	Carpeta de	archivos			
📕 Profesor Luis N	Clase 1	1/08/2	020 16:17	Carpeta de	archivos			
Este equino	Clase 2	01/08/2	020 16:17	Carpeta de	archivos			
Descargas	Clase 3	01/08/2	020 16:17	Carpeta de	archivos			
Documents	Clase 4	01/08/2	020 16:17	Carpeta de	archivos			
Escritorio	Clase 5	01/08/2	020 16:17	Carpeta de	archivos			
Imágenes								
Música								
Objetos 3D								
Vídeos								
Disco local (C:)								
Disco local (D)								

5.- Escribir la siguiente línea de comandos para establecer la configuración de nombre de usuario / correo electrónico específica del repositorio: (ESTO SOLO SE HACE UNA VEZ). Debes usar el mismo correo que utilizaste al crear tu cuenta en Github.







MinTIC

NINGW64:/c/estructura-datos	-	×
DELL@DESKTOP-R21Q3R8 MINGW64 /c/estructura-datos \$ git init Initialized empty Git repository in C:/estructura-datos/.git/		^
DELL@DESKTOP-P2103R8 MINGW04 /c/estructura-datos (master) \$ git configglobal user.name "luisguillermomolero"		
DELL@DESKTOP-R21Q3R8 MINGW64 /c/estructura-datos (master) \$		
DELL@DESKTOP-R210288_MINGW64 /c/escructura-datos (master) \$ git configglobal user.email "luisguillermomolero@gmail.com"		
DELL@DESKTOP-R21Q3R8 MINGW64 /c/estructura-datos (master) \$		

Supongamos que en esta primera entrega usted. va a subir el primer programa realizado como tarea, como se muestra en la siguiente imagen:

hivo Inicio Comparti	r Vista				_	□ × ^ (?)
nclar al eso rápido Portapapel	X Cortar E Copiar ruta de acceso I Pegar acceso directo les	Mover Copiar a a a Copiar Oroanizar	E I Nuevo elemento ▼ iar Nueva re carpeta Nuevo	Propiedades • Historial	Seleccionar todo No seleccionar nada Invertir selección Seleccionar	
→ ~ ↑ 🔍 Este	e equipo 🔸 Disco local	(C:) > estructura-datos > C	ase 1	v ひ Posca	r en Clase 1	
📕 Introducción a ^	Nombre	Loo	in the second second	Tamaño		
Introducción a	Programa1	01/	8/2020 16:52 Carpeta d	le archivos		
Profesor Luis N						
🞐 Este equipo						
Descargas						
Documents						
🔜 Escritorio						
戻 Imágenes						
Música						
Objetos 3D						
Vídeos						
関 Disco local (C:)						
🤳 Disco local (D:						
Red						
~						
elemento 1 elemento se	eleccionado					800 K

6.- Para subir ese primer ejercicio resuelto, <u>debes repetir **solo** el paso 3</u> y cuando aparezca el Bash de **GIT** escribir la siguiente línea de comando:

(ESTO LO DEBES HACER CADA VEZ QUE MODIFIQUES ALGÚN EJERCICIO O HAGAS UN EJERCICIO NUEVO)

git add .





He

QUE



NINGW64:/c/estructura-datos		—	×
DELL@DESKTOr-R21Q3R8 MINGWOT /c/estructura-datos (mast \$ git add .	er)		^
DELL@DESKTOP-R2103KG MINGW64 /c/estructura-datos (mast \$	er)		
			\sim

git add : Lleva el control de los archivos que se agregan luego de escribir ese comando y **git add** . sirve para agregar todos los archivos modificados después del primer envío (commit).

7.- Una vez ejecutado el git add . ejecutamos la siguiente línea de comando para validar el estado actual de nuestro Git.

git status				x
DELLADORSKIDP-R2103R8 MING#64 /c/estructura-datos (maste	r)		L)	Î
DELLODESKIP / Marsha MINGW64 /C/esca tura-datos (maste \$ git statut				
On branch massar				
<pre>changes to be committed: (use "git m -cached files" to unstage) new file: Clase !/Programal/.vs/Programal/De new file: Clase !/Programal/.rogramal.cg new file: Clase !/Programal.rcgncg new file: rcgncg new file: rcgncg n</pre>	ignTimeBuild/.dtbcache.v2 /.su0			
n new file: Clase 1/Programal/bin/Debug/netcor new file: Clase 1/Programal/bin/Debug/netcor new file: Clase 1/Programal/bin/Debug/netcor new file: Clase 1/Programal/bin/Debug/netcor onfio.dev.icon	хир3.1/Programm1.d11 мир3.1/Programm1.exe лард3.1/Programm1.pdb мир3.1/Programm1.runtimec			
new file: Clase 1/Programa1/bin/Debug/netcore onfig.json				~ (
Info.cs new file: Clase 1/Programa1/obj/Debug/netcore				
InfoInputs.cache new file: Clase 1/Programa1/obj/Debug/netcore ache				2.
new file: Clase 1/Programa1/obj/Debug/netcore oreCompileInputs.cache				
<pre>ileListAbsolute.txt new file: Clase 1/Programa1/obj/Debug/netcore new file: Clase 1/Programa1/obj/Debug/netcore</pre>	app3.1/Programa1.csproj.F app3.1/Programa1.csprojAs			
<pre>semblyReference.cache new file: Clase 1/Programa1/obj/Debug/netcore</pre>	sapp3.1/Programa1.dl1			
new file: Clase 1/Programal/obj/Debug/netcord new file: Clase 1/Programal/obj/Debug/netcord meconfig.cache	app3.1/Programal.exe app3.1/Programal.genrunti			
<pre>new file: Clase 1/Programal/obj/Debug/netcore new file: Clase 1/Programal/obj/Programal.cs; new file: Clase 1/Programal/obj/Programal.cs; new file: Clase 1/Programal/obj/project.asget new file: Clase 1/Programal/obj/project.asget new file: Clase 1/Programal/obj/project.asget</pre>	app3.1/Programa1.pdb rr0.muget.gprops rr0.muget.gprops s.json c.ache			
DELL@DESKTOP-R21Q3R8 MING#64 /c/estructura-datos (maste				1. Contract (1. Contract)
第 2 3 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	K @ @ @ @ @ X @ X O @ <u>@ % ^</u>		🔨 🔿 🖼 🎢 Φ)) 💿 ESP 1820 🖡	
				P
chos	#MisiónTIC2022	2	$\overline{4}$	R
CONECTAN 🗸				





El comando git status te mostrará los diferentes estados de los archivos en tu directorio de trabajo y área de ensayo. Qué archivos están modificados y sin seguimiento y cuáles con seguimiento pero no confirmados aún. En su forma normal, también te mostrará algunos consejos básicos sobre

cómo mover archivos entre estas etapas.

8.- Una vez ejecutado el comando git status ejecutamos la siguiente línea de comando para instanciar los cambios preparados en ese momento.

```
git commit -m "mensaje"
```

ejemplo:git commit -m "primera actualizacion"



git commit : Este comando indica que esta lista alguna funcionalidad para que sea una del código. Este comando se repite *n* veces cada vez se cambia el código.







9.- Una vez ejecutado el git commit ubicamos los datos que guardamos en el sub-paso 5 (**Guarda** estos valores debido a que luego lo necesitaras <u>para subir tus aplicaciones</u> desde el *Bash* de **Git** a este **Github**) de nuestro Paso 1. (**Paso 1. Crear un nuevo repositorio**)

Code	Issues	11 Pull requests	🕞 Actions 🛄 Pro	jects 🖽 Wiki 🛈 Security	🗠 Insights 🛛	Settings	
				Quick setup — if you	've done this	kind of thing before	
				[슈] Set up in Desktop Or	HTTPS SSH	https://github.com/luisguillermomolero/estructura-datos.git	ť
				Get started by creating a new f	ile or uploading an	existing file. We recommend every repository include a README, LICENSE, and .gitignore.	
				or create a new rep echo "# estructura-dato git init git add REAPKE.md git comit -m =ficat git remore and origin h git push -u origin more	s" >> README.md	e command line m/luisguillermomolero/estructura-datos.git	
				or push an existing	repository fr	rom the command line	
				git remote add origin h git push -u origin maste	ttps://github.co er	om/luisguillermomolero/estructura-datos.git	C
				or import code from You can initialize this repositor	m another re y with code from a	pository Subversion, Mercurial, or TFS project.	

10.- Ejecutamos la siguiente línea de comando que tomamos del paso agregar un nuevo control remoto.

git remote add origin https://github.com/luisguillermomolero/estructura-datos.git







11.- Finalmente, luego de ejecutar la línea de comando anterior, ejecutamos un git push para subir tus cambios locales a tu repositorio en línea.

git push --set-upstream origin master



Ya podemos ver nuestro primer proyecto (Programa) en la nuevo de Github

			X 1		
Search or jump to 📝 Pull requests Issues Marketplace E	φlore				¢ +
	Learn Git and	GitHub without any code!			
	Using the Hello World guide, you'l	start a branch, write comments, and open a pull	request.		
		Read the guide			
🗟 luisguillermomolero / estructura-datos				⊙ Unwatch → 1 🛱 Star	r 0 ¥ Fork
<> Code ① Issues □ Pull requests ⊙ Actions □ Projects □ W	ki 🕕 Security 🗠 Insights 🐵 Settings				
P mast	rr - P ^a l branch ⊗0 tags	Go to file Add file - ↓ Code -	About §	8	
() ()	pombre de usuario en Github p	Imera actualiz	Repositorio ejercicios materia Estructura de Datos		
Clar	e 1/Programa1 primera actualizacion	8 minutes ago	>		
Help pec	ple interested in this reportion, one	Add a README	Releases		
			Create a new release		
			Packages		
			No packages published Publish your first package		
			Languages		
			• C# 100.0%	•	
© 2020 GitHub.	nc. Terms Privacy Security Status Help	Contact GitHub Pricing	API Training Blog A	bour	
				2 N	
ecnos	#Mició	TICODOO			6
	#111210	ITICZUZZ			
E) CONECTAN V					~





IMPORTANTE

Cada vez que se modifique un proyecto (programa) o se cree uno nuevo, se debe ejecutar <u>del paso</u> <u>2</u> (**Paso 2. Subir nuestros proyectos a Github**) los siguientes sub-pasos:

3.- Una vez creada las carpetas de cada clase y luego de haber instalado **Git** y **Github**, hacer <u>clic</u> <u>derecho</u> sobre la carpeta "*estructura-datos*" y luego sobre la opción "**Git Bash Here**".

6.- Para subir ese primer ejercicio resuelto, <u>debes repetir **solo** el paso 3</u> y cuando aparezca el Bash de **GIT** escribir la siguiente línea de comando:

7.- Una vez ejecutado el git add . ejecutamos la siguiente línea de comando para validar el estado actual de nuestro Git.

8.- Una vez ejecutado el comando git status ejecutamos la siguiente línea de comando para instanciar los cambios preparados en ese momento.

10.- Ejecutamos la siguiente línea de comando que tomamos del paso agregar un nuevo control remoto.

11.- Finalmente, luego de ejecutar la línea de comando anterior, ejecutamos un git push para subir tus cambios locales a tu repositorio en línea.

