

Aprendiendo a usar Github

Git es un sistema de control de versiones que permite registrar los cambios realizados a archivos de computadora (Ej. Código fuente) con el fin de coordinar los avances de un equipo de trabajo. En este laboratorio aprenderás a crear tu cuenta de GitHub y explorarás todas sus funciones para desarrollar proyectos de ingeniería de software a lo largo de tu carrera y vida profesional.

NOTA → Este laboratorio se diseñó para **Windows 10**. A lo largo de los ejercicios se especificarán los cambios de los pasos con respecto a versiones anteriores de Windows y MAC OS

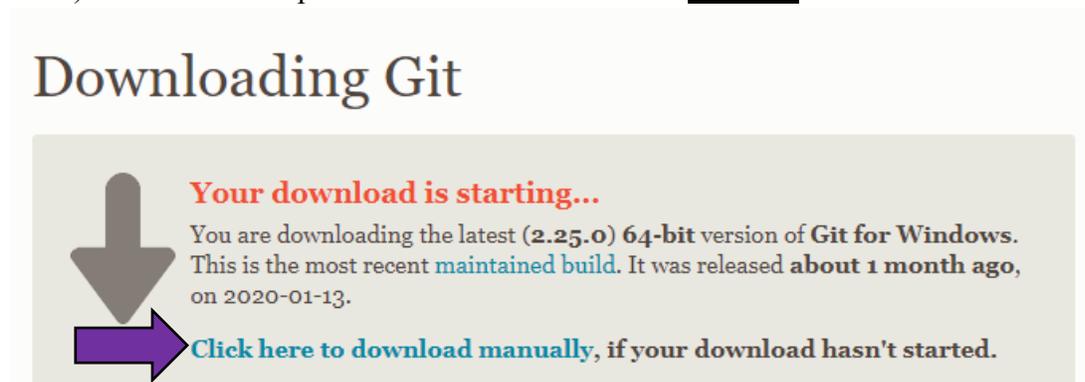
I. Instalar Git en tu computadora

Antes de empezar a usar Github, debemos instalar el software con el cual funciona. A continuación, veremos cómo instalar Git en tu computadora.

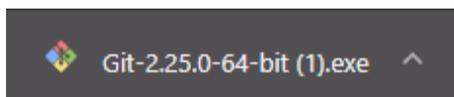
- 1) Ingresa a <https://git-scm.com/download>
- 2) Selecciona el sistema operativo en el cual deseas instalar Git



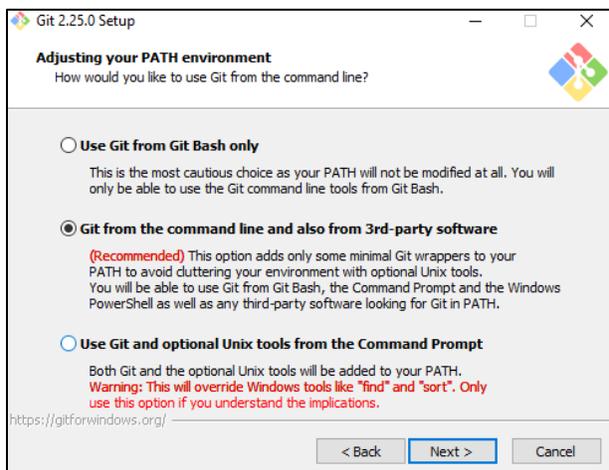
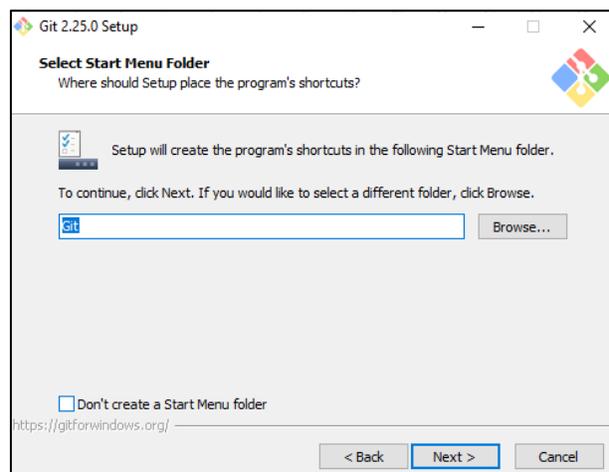
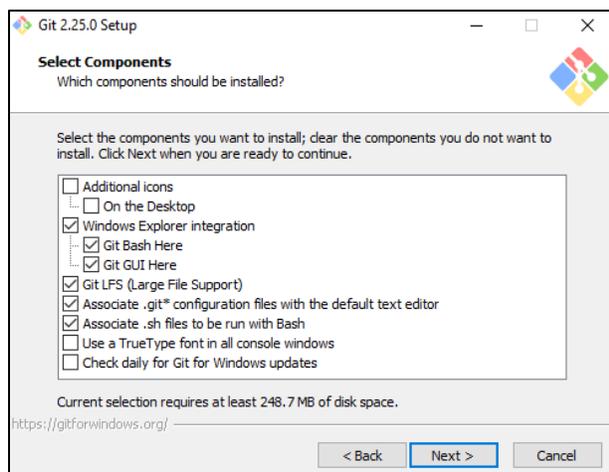
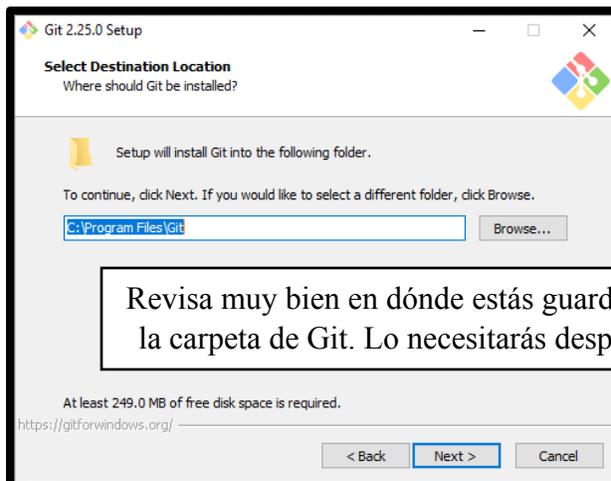
- 3) Selecciona la opción de instalar Git de forma **manual**.

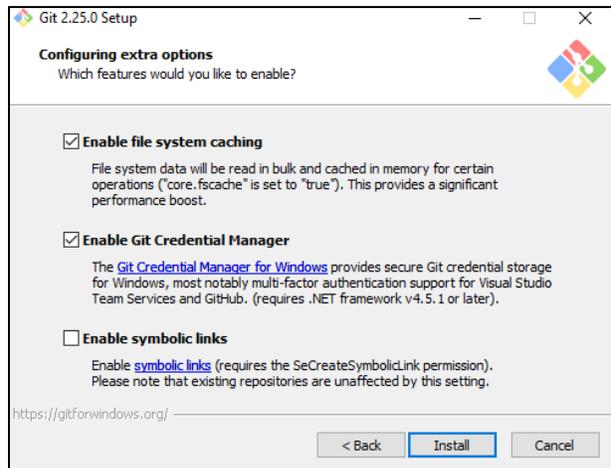
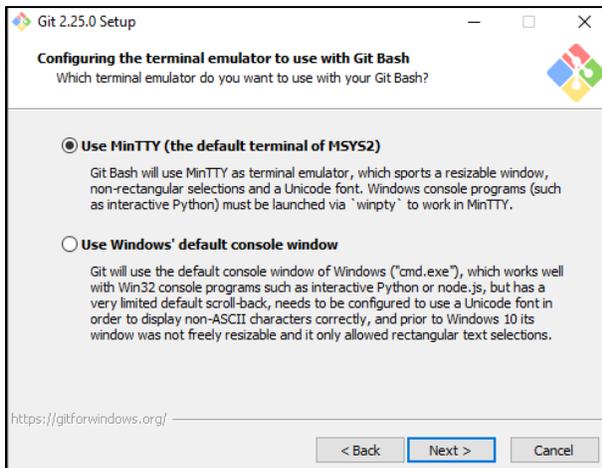
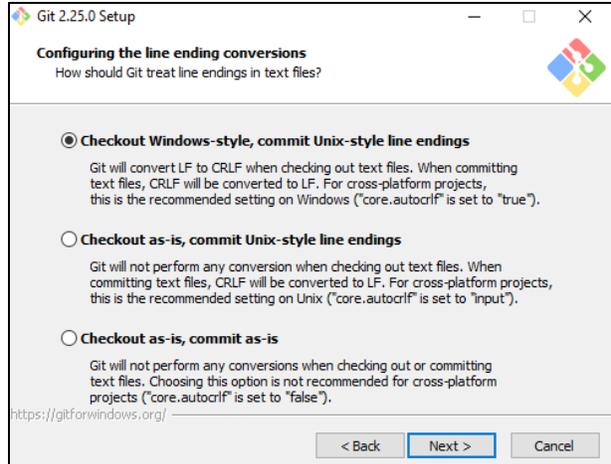
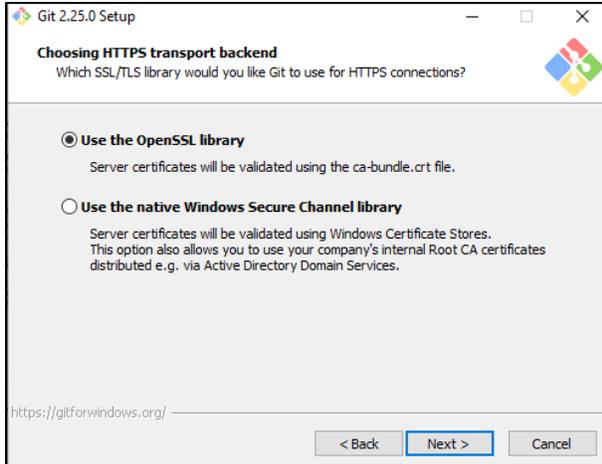


- 4) Se iniciará a descargar el ejecutable de Git. Inícialo una vez que haya terminado.

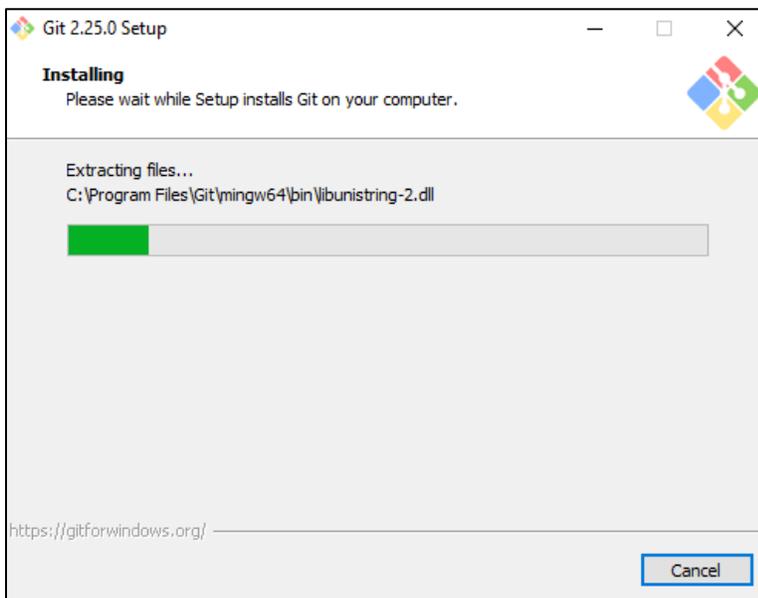


- 5) Presiona el botón “Next” en las siguientes pestañas. No cambies la configuración actual.

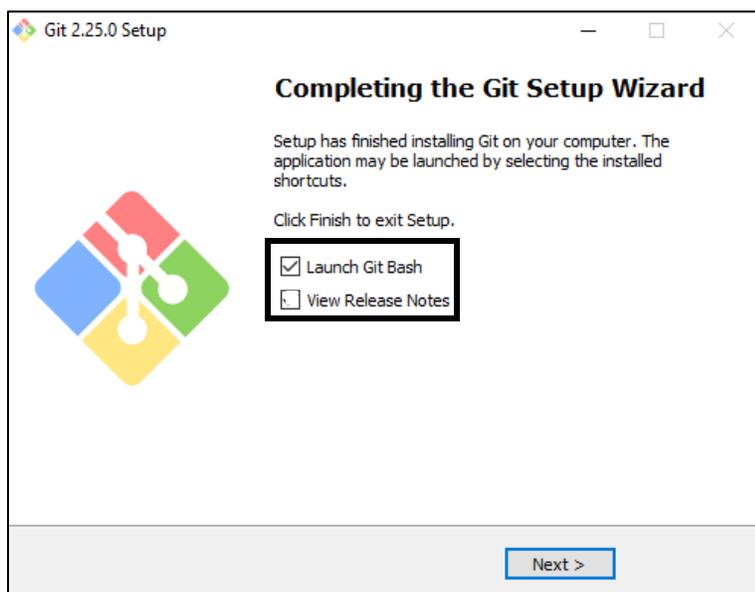




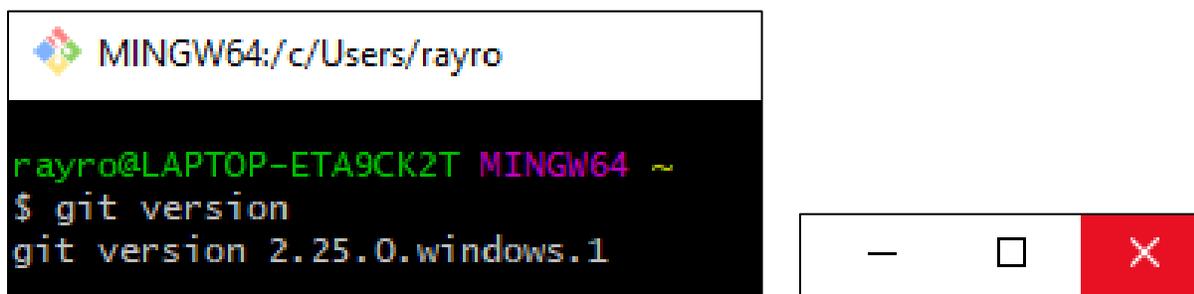
6) Se empezará a instalar Git. Espera unos minutos.



- 7) Selecciona el recuadro ‘Launch Git Bash’ y desactiva el recuadro ‘View Release Notes’. Después presiona el botón “Next”.



- 8) Se abrirá Git Bash, una aplicación que da una capa de emulación para una experiencia de línea de comandos de Git. Para verificar que todo está bien, escribe el comando ***git version***. Si nos arroja la versión de Git, entonces todo se instaló correctamente.



- 9) Presiona el botón Salir en la parte superior derecha para cerrar Git Bash

II. Añadir Git al PATH de Windows

Como vimos anteriormente, Git Bash es una aplicación que te permite usar Git en tu computadora. Sin embargo, ¿qué pasa si queremos usar otra interfaz de línea de comandos, tal como la terminal de Windows o Windows Powershell? Si intentamos usar algún comando de Git en alguna de estas, nos dirán que no lo reconocen como un comando o programa, por lo que, si queremos manejar Git desde cualquier parte de nuestra computadora, tendremos que añadir su ruta/dirección a la variable de entorno PATH del sistema operativo.

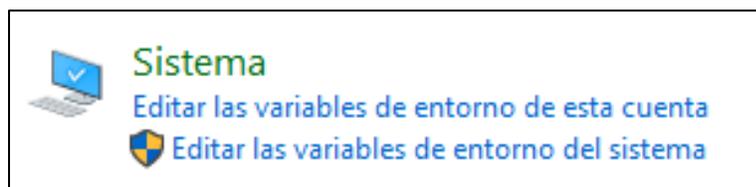
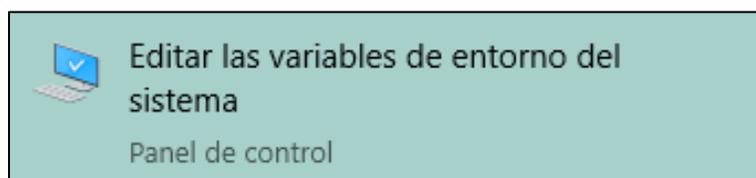
NOTA: Si tienes MAC, **no necesitas hacer este ejercicio**. MAC OS ya tiene instalado en su path la opción de usar Git desde cualquier parte de la computadora

1) Windows 10 → En la barra de búsqueda de Windows, escribe “path”

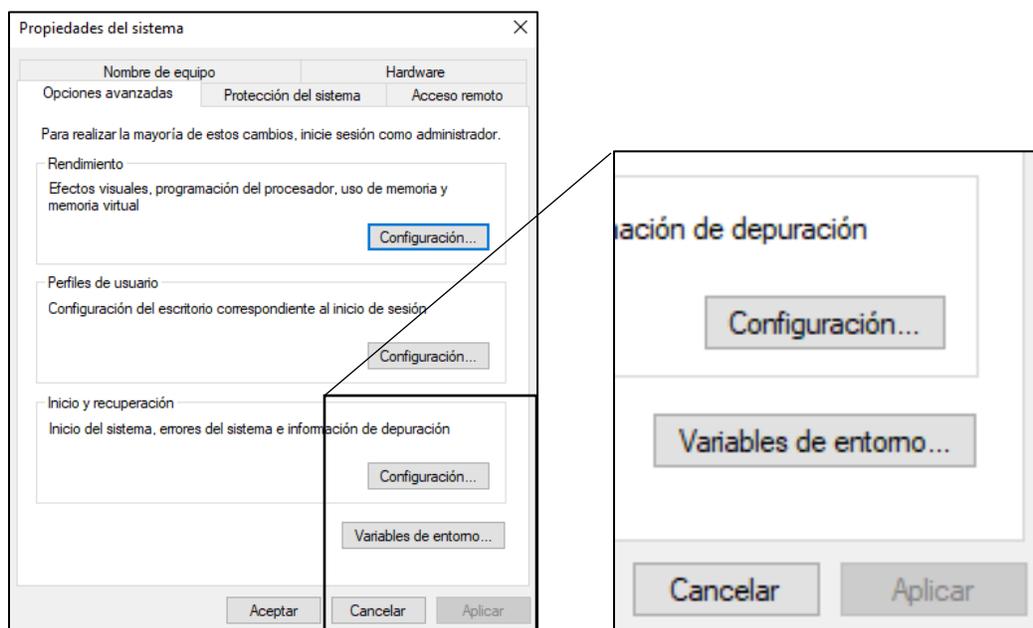
1a) Windows 7/8 → Ingresa al Panel de Control y escribe “path” en la barra de búsqueda del Panel de Control



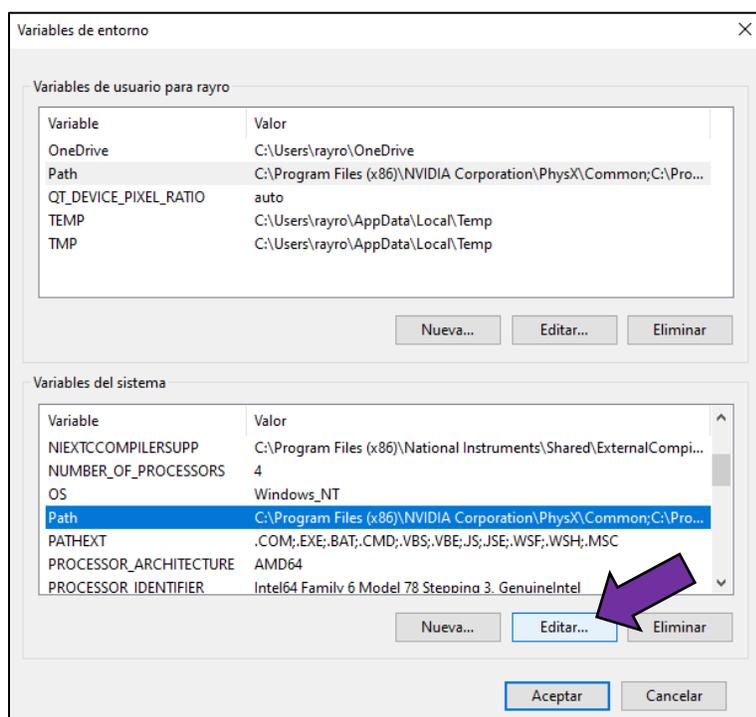
2) Selecciona la opción “Editar las variables de entorno del sistema”



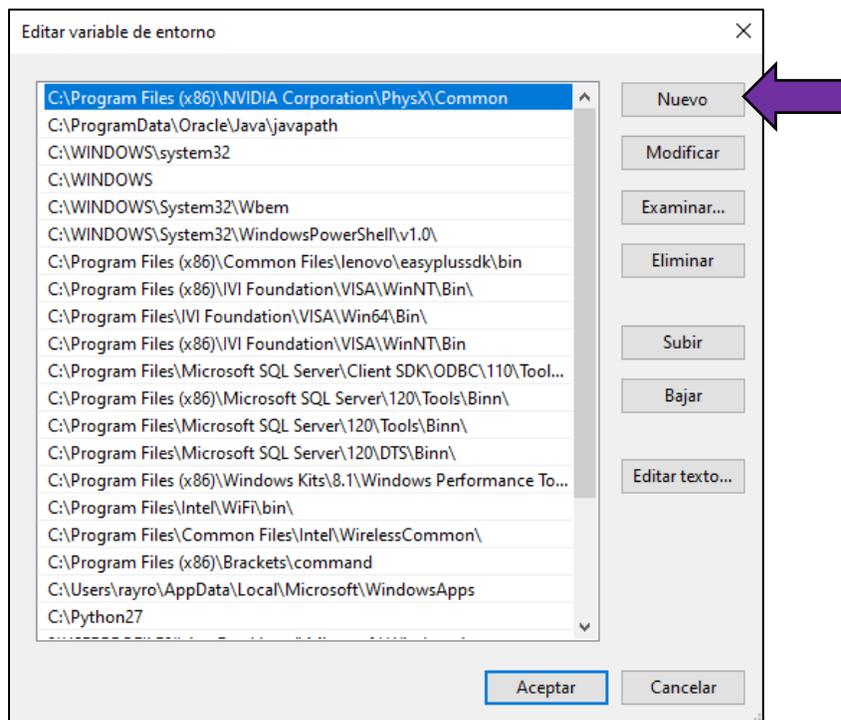
3) Aparecerá la siguiente pantalla. Presiona el botón ‘Variables de entorno’



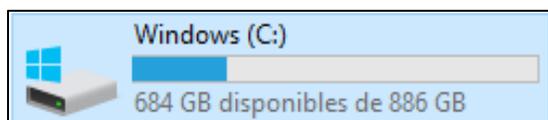
- 4) Se desplegará la interfaz ‘Variables de entorno’. En la sección ‘Variables del sistema’ selecciona la variable ‘Path’ y presiona el botón Editar



- 5) Aparecerán todas las rutas que están configuradas en el Path. Presiona el botón Nuevo



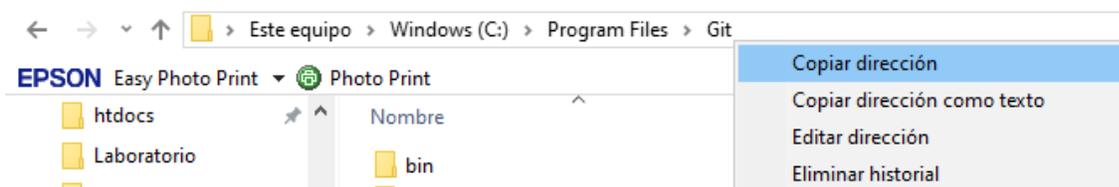
- 6) Copia la dirección en donde se encuentra Git en tu computadora
- Busca la carpeta de Git en tu unidad Windows (C:) en la carpeta donde la hayas guardado durante la instalación.



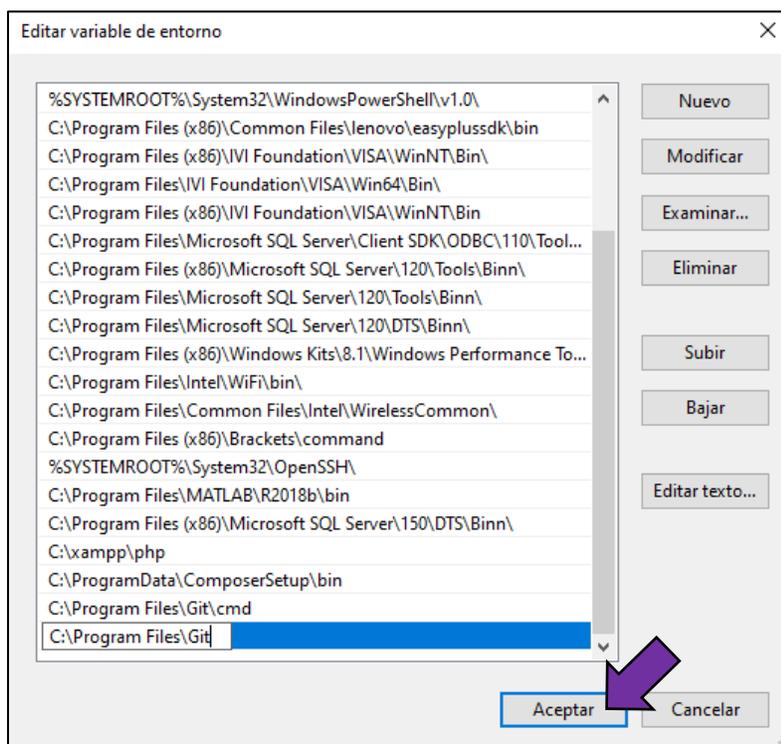
Nombre	Fecha de modifica...	Tipo	Tamaño
Archivos de programa (x86)	26/01/2020 12:51 a...	Carpeta de archivos	
MinGW	21/03/2018 02:36 ...	Carpeta de archivos	
Minitab	03/10/2018 10:32 a...	Carpeta de archivos	
Packages	10/01/2017 12:06 ...	Carpeta de archivos	
PerfLogs	18/03/2019 10:52 ...	Carpeta de archivos	
Program Files	13/02/2020 12:08 ...	Carpeta de archivos	
pspdata	15/11/2019 02:55 ...	Carpeta de archivos	
Python27	27/03/2018 05:41 ...	Carpeta de archivos	
SOLIDWORKS Data	22/09/2018 04:04 ...	Carpeta de archivos	
temp	26/01/2020 12:38 a...	Carpeta de archivos	
Usuarios	01/09/2019 06:59 ...	Carpeta de archivos	
Windows	13/02/2020 03:47 a...	Carpeta de archivos	
Windows.old	08/10/2019 06:19 ...	Carpeta de archivos	

Nombre	Fecha de modifica...	Tipo	Tamaño
Adobe	10/11/2019 11:56 a...	Carpeta de archivos	
Android	13/02/2020 11:49 a...	Carpeta de archivos	
CCleaner	17/10/2019 02:23 ...	Carpeta de archivos	
Cisco Packet Tracer 7.1.1	01/09/2019 01:11 a...	Carpeta de archivos	
Cisco Packet Tracer 7.2.1	12/09/2019 03:14 ...	Carpeta de archivos	
Common Files	01/09/2019 12:45 ...	Carpeta de archivos	
Defraggler	26/01/2020 01:16 a...	Carpeta de archivos	
DIFX	22/08/2016 08:54 a...	Carpeta de archivos	
Dolby	01/09/2019 06:54 ...	Carpeta de archivos	
EPSON	15/01/2018 09:39 ...	Carpeta de archivos	
EpsonNet	15/01/2018 09:39 ...	Carpeta de archivos	
FileZilla FTP Client	13/02/2020 12:28 ...	Carpeta de archivos	
Git	13/02/2020 01:57 a...	Carpeta de archivos	
HandBrake	17/09/2017 04:14 ...	Carpeta de archivos	
Intel	13/02/2020 12:01 ...	Carpeta de archivos	
Intel Security	16/05/2017 11:18 a...	Carpeta de archivos	
Internet Explorer	13/02/2020 03:47 a...	Carpeta de archivos	
IVI Foundation	21/08/2016 10:44 ...	Carpeta de archivos	
Java	07/08/2018 08:03 ...	Carpeta de archivos	

- Entra a la carpeta. Una vez dentro, haz clic derecho sobre el nombre en la barra de dirección. Selecciona la opción 'Copiar dirección'.



- Pega la dirección en la interfaz de configuración del Path
- Una vez insertada la ruta de Git, presiona el botón Aceptar



- 7) Abre la terminal de Windows. Escribe 'git'. Si se te despliega la lista de comandos de Git, entonces insertaste la ruta correctamente.

```

Microsoft Windows [Versión 10.0.18362.657]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\rayro> git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
  [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
  [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
  [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
  <command> [<args>]

These are common Git commands used in various situations:

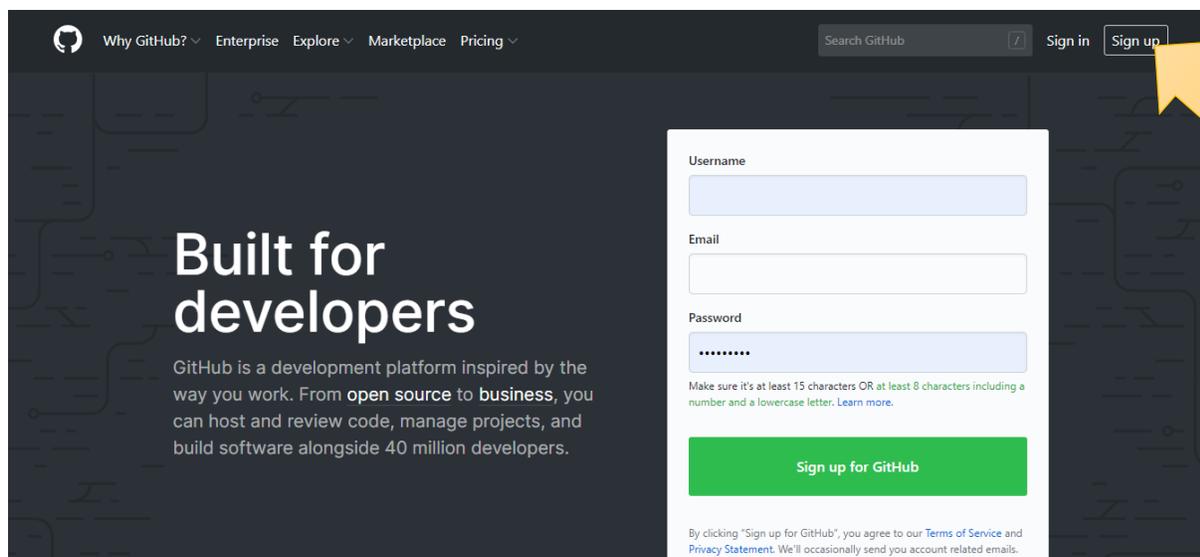
start a working area (see also: git help tutorial)
  clone          Clone a repository into a new directory
  init           Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add            Add file contents to the index
  mv             Move or rename a file, a directory, or a symlink
  restore        Restore working tree files
  rm             Remove files from the working tree and from the index
  sparse-checkout Initialize and modify the sparse-checkout

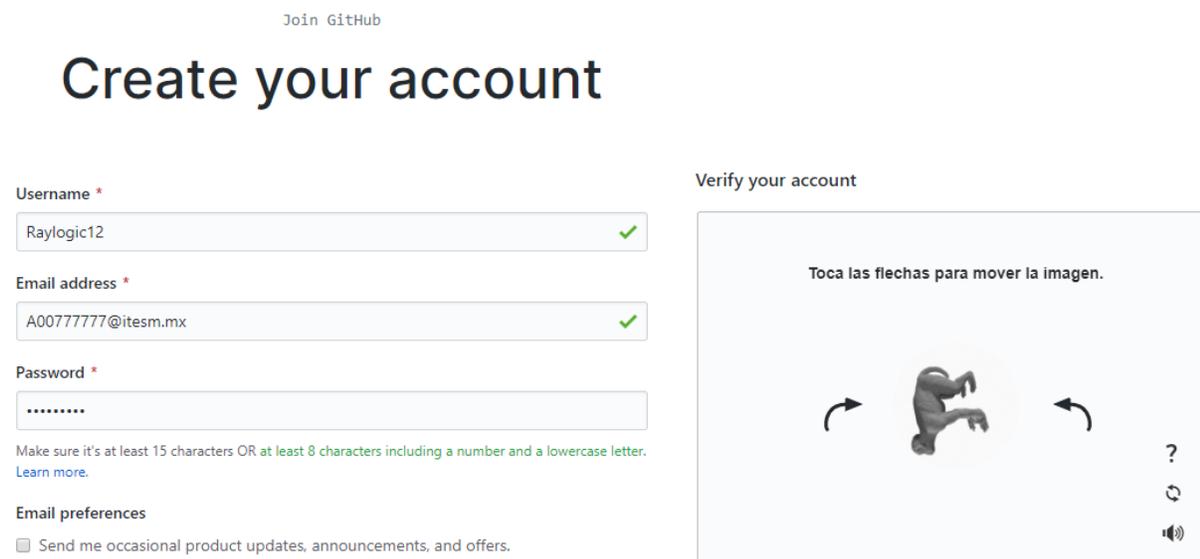
examine the history and state (see also: git help revisions)
  bisect        Use binary search to find the commit that introduced a bug
  diff          Show changes between commits, commit and working tree, etc
  grep          Print lines matching a pattern
  log           Show commit logs
  show          Show various types of objects
  status        Show the working tree status
  
```

III. Crear tu cuenta en Github

- 1) Ingresa a github.com
- 2) En la parte superior derecha, presiona el botón 'Sign up'



- 3) Una vez que se abra la ventana de registro, rellena los campos con la información correspondiente:
 - Nombre de usuario
 - Correo electrónico (De preferencia usa el mail institucional del TEC)
 - Contraseña (Sigue las indicaciones sobre los requisitos que debe tener)
 - No selecciones el recuadro "Send me occasional product updates..."
 - Contesta el CAPTCHA



4) Selecciona el plan gratuito

Individual

Pick the plan that's right for you, personally.

Team

Choose a plan to help your team grow and collaborate.



Free

\$0 USD

The basics of GitHub for every developer

Choose Free



Pro

\$7 USD

Per month

Pro tools for developers with advanced requirements

Choose Pro

5) Contesta las preguntas sobre tus intereses y propósito de usar Github

Welcome to GitHub

Woohoo! You've joined millions of developers who are doing their best work on GitHub. Tell us what you're interested in. We'll help you get there.

How much programming experience do you have?

<p>None</p> <p>I don't program at all</p>	<p>A little</p> <p>I'm new to programming</p>
<p>A moderate amount</p> <p>I'm somewhat experienced</p>	<p>A lot</p> <p>I'm very experienced</p>

¿Cuánta experiencia en programación tienes?

What do you plan to use GitHub for?

(Select up to 3)

 Learn to code	 Learn Git and GitHub	 Host a project (repository)
 Create a website with GitHub Pages	 Collaborating with my team	 Find and contribute to open source
 School work and student projects	 Use the GitHub API	 Other

I am interested in:

languages, frameworks, industries

We'll connect you with communities and projects that fit your interests.

For example: [server](#) [pihole](#) [telegram](#)

Aquí pon tus intereses personales y profesionales

[Complete setup](#)

[Skip this step](#)

- 6) Tras haber contestado las preguntas, Github mostrará un mensaje diciendo que se te envió un mail de confirmación al correo que registraste. Cierra la pestaña actual.



Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address. An email containing verification instructions was sent to paty.arenas@gmail.com.

Didn't get the email? [Resend verification email](#) or [change your email settings](#).

- 7) Abre tu correo electrónico en una nueva pestaña y busca el mail de Github. Se te mostrará el siguiente mensaje. Presiona el botón “Verify email address”.

Almost done, **@Raylogic12!** To complete your GitHub sign up, we just need to verify your email address:

paty.arenas@gmail.com.

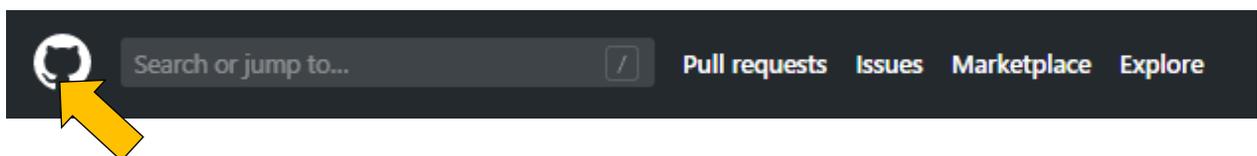
[Verify email address](#)

Once verified, you can start using all of GitHub's features to explore, build, and share projects.

Button not working? Paste the following link into your browser: https://github.com/users/Raylogic12/emails/94366977/confirm_verification/f119fa624cbae9bc701fadf5d386869812f990d2

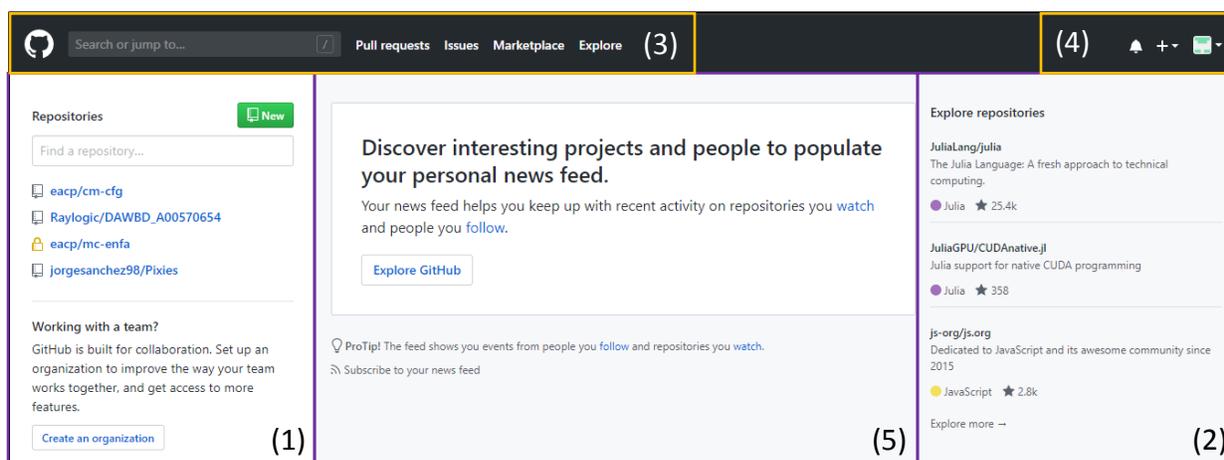
You're receiving this email because you recently created a new GitHub account or added a new email address. If this wasn't you, please ignore this email.

- 8) Con tu correo verificado, se abrirá la página de Github de nuevo y te preguntará si quieres crear un nuevo repositorio. Antes de empezar con esto, tenemos que saber cómo movernos por Github, por lo que veremos más adelante. De momento, presiona el logo de Github que se encuentra en la parte superior izquierda para ir al menú principal.



IV. Explorar Github

1) Se te abrirá el menú principal de tu cuenta de Github



(1) Del lado izquierdo se encuentran todos los repositorios que has creado o estás participando. También te permite crear un nuevo repositorio.

(2) Del lado derecho aparece la opción de explorar repositorios de otras personas

(3) En la parte superior izquierda está la barra de búsqueda, donde puedes buscar repositorios, usuarios, avances y códigos de otras personas.

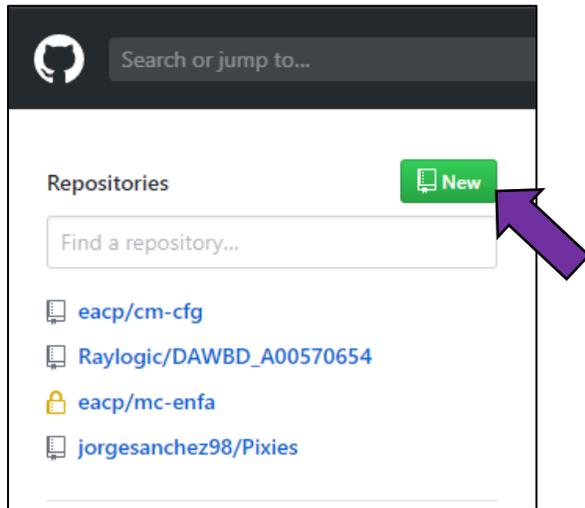
(4) En la parte superior derecha está la configuración de tu cuenta, donde puedes modificar los datos de tu perfil, consultar tus repositorios, revisar tus notificaciones y configurar aspectos de tu cuenta (seguridad, facturación, etc...)

(5) En la parte central se encuentra el botón 'Explorar Github', el cual te mostrará repositorios y proyectos relacionados con los intereses que estableciste al crear tu cuenta.

V. Crear y explorar un repositorio

Un repositorio es un espacio centralizado donde de suben, guardan, organizan y descargan archivos informáticos. En la ingeniería de software se usan para gestionar proyectos que involucran el uso de archivos de código fuente. En Github puedes crear varios repositorios para trabajar junto a otras personas en proyectos interdisciplinarios y compartir tus avances para recibir retroalimentación o ayudar a los demás en sus respectivas labores. En esta sección aprenderemos a crear un repositorio y saber cómo moverse en él.

- 1) En el menú principal, presiona el botón Nuevo en la parte superior izquierda para crear un nuevo repositorio.



- 2) Se desplegará la interfaz 'Crear un nuevo repositorio'. Rellena los campos con la siguiente información. Una vez terminado, presiona el botón 'Crear repositorio'.

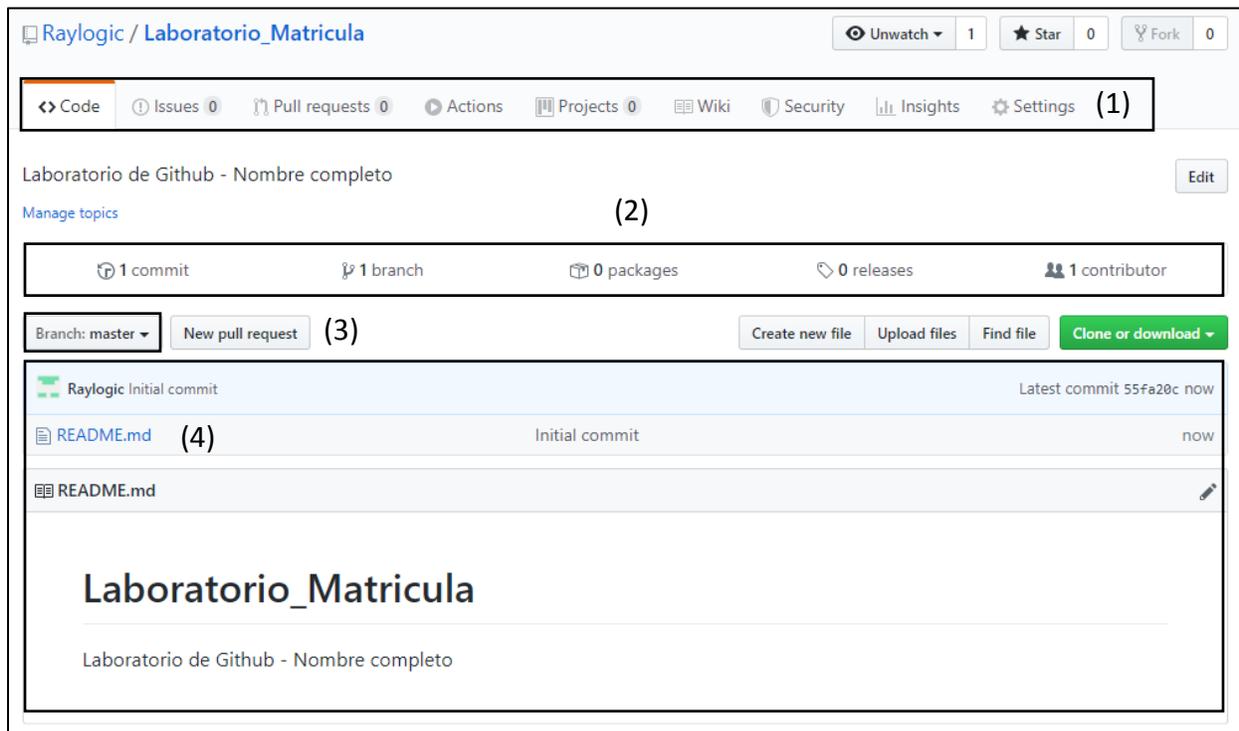
 A screenshot of the "Create a new repository" form on GitHub. The form has several fields:

- Owner:** A dropdown menu showing "Raylogic".
- Repository name:** A text input field containing "Laboratorio_Matricula" with a green checkmark icon to its right. A black box highlights this field.
- Description (optional):** A text input field containing "Laboratorio de Github - Nombre completo". A black box highlights this field.
- Visibility:** Two radio buttons. The "Public" option is selected, with the text "Anyone can see this repository. You choose who can commit." below it. A black box highlights this section. The "Private" option is unselected, with the text "You choose who can see and commit to this repository." below it.
- Initialization options:** A checkbox labeled "Initialize this repository with a README" is checked. Below it is the text "This will let you immediately clone the repository to your computer." There are two dropdown menus: "Add .gitignore: None" and "Add a license: None".
- Buttons:** A green "Create repository" button at the bottom. A purple arrow points to this button.

 There are two text boxes with arrows pointing to the form:

- On the left, a box contains the text: ".gitignore es un archivo que le especifica a Git qué archivos y directorios debe ignorar cuando se hace un commit." An arrow points from this box to the ".gitignore" dropdown menu.
- On the right, a box contains the text: "Una licencia permite que otras personas usen, cambien y distribuyan el proyecto en tu repositorio." An arrow points from this box to the "Add a license" dropdown menu.

3) Se desplegará la pantalla principal del repositorio



- (1) En la parte superior, se encuentran las pestañas para revisar el estado del repositorio: Archivos, Conflictos, Pull Requests, Configuración, etc...
- (2) Abajo del nombre del repositorio, se encuentra el número de commits, ramas y usuarios que hay en el repositorio
- (3) Debajo del número de commits, aparece el nombre de la rama en la que estás actualmente. La rama default de cualquier repositorio se llama “master”. (Más adelante veremos esto).
- (4) Debajo del nombre de la rama, aparece todo el contenido que está guardado en ella (archivos, documentos, código fuente, etc...)

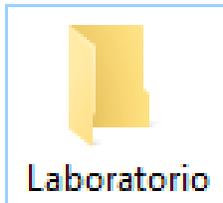
V. Clonar un repositorio en tu computadora

Como hemos visto, un repositorio es un espacio para guardar, administrar y organizar los archivos de un proyecto. Sin embargo, si tu computadora es en donde realizas todo el trabajo, ¿cómo puedes subir tus avances al repositorio?

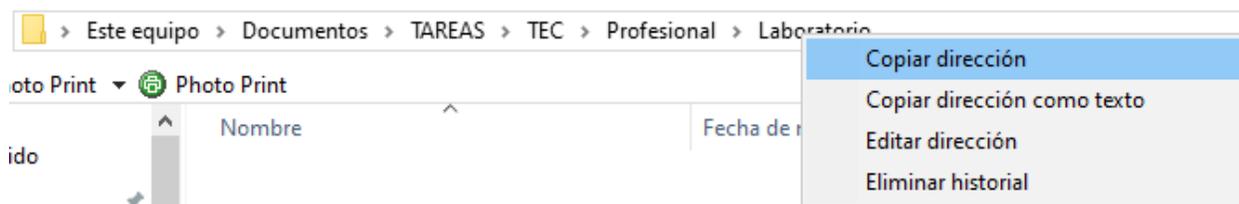
Aquí es donde entran los conceptos de repositorio **local** y **remoto**. Verás, todos tus repositorios en Github son versiones de tu proyecto que se encuentran hospedados en Internet o cualquier otra red, por lo tanto, son espacios **remotos**.

Para acceder a sus contenidos, necesitas crear un espacio en tu computadora desde el cual puedas enviar y recibir los cambios que se hagan en estos. A este lugar le llamaremos repositorios **locales**. En esta sección, aprenderás a crear un repositorio local en tu computadora para después conectarlo con el repositorio remoto que acabas de crear en Github.

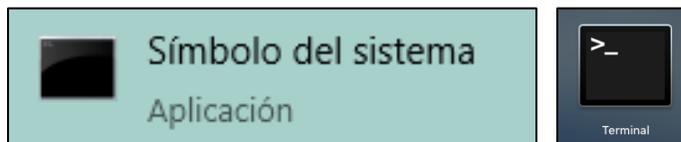
- 1) Crea una carpeta en tu computadora (ponle con un nombre que tenga una sola palabra).



- 2) Desde la terminal de Windows, accede a la dirección de la carpeta que acabas de crear
 - En el explorador de archivos entra a la carpeta. Una vez dentro, haz clic derecho sobre el nombre en la barra de dirección. Selecciona la opción ‘Copiar dirección’.



- Abre la terminal de Windows/MAC



- Escribe el código: ***cd (Dirección de la carpeta)*** y estarás dentro de la carpeta

```

C:\> Símbolo del sistema
Microsoft Windows [Versión 10.0.18362.592]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.
C:\Users\rayro> cd C:\Users\rayro\Documents\TAREAS\TEC\Profesional\Laboratorio
C:\Users\rayro\Documents\TAREAS\TEC\Profesional\Laboratorio>
  
```

3) En Github, entra a tu repositorio Laboratorio_(Matricula)

The screenshot shows the GitHub interface for the repository 'Raylogic / Laboratorio_Matricula'. At the top, there are navigation links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. Below this, the repository name 'Laboratorio de Github - Nombre completo' is displayed with an 'Edit' button. A summary bar shows 1 commit, 1 branch, 0 packages, 0 releases, and 1 contributor. A secondary bar includes a branch selector (set to 'master'), a 'New pull request' button, and buttons for 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The main content area shows the 'README.md' file, which contains the text 'Laboratorio_Matricula' and 'Laboratorio de Github - Nombre completo'.

4) En la parte derecha, presiona el botón “Clone o download”

A close-up of the 'Clone or download' button, which is highlighted in green. A purple arrow points to the right side of the button, indicating the click location.

5) Aparecerá un cuadro de diálogo con una dirección URL. Cópialo con el método que desees.

The screenshot shows the 'Clone or download' dialog box. It has a title bar with 'new file', 'Upload files', 'Find file', and 'Clone or download'. The main content area is titled 'Clone with HTTPS' and includes a 'Use SSH' link. Below the title, it says 'Use Git or checkout with SVN using the web URL.' A text input field contains the URL 'https://github.com/Raylogic/Laboratorio_'. At the bottom, there are two buttons: 'Open in Desktop' and 'Download ZIP'.

6) Escribe el comando ***git clone***. Luego, pega el URL que copiaste del repositorio.

```
git clone https://github.com/Raylogic/Laboratorio_Matricula.git
```

7) Se clonará tu repositorio remoto en la dirección seleccionada, creando tu repositorio local

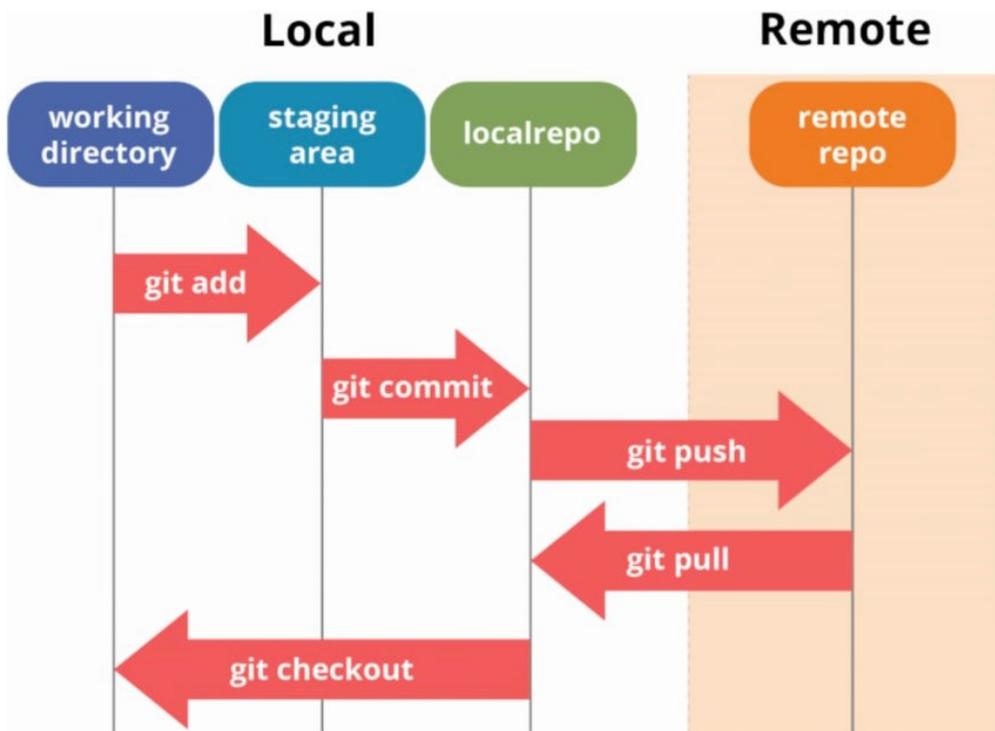
```
Cloning into 'Laboratorio_Matricula'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

VI. Subir contenido al repositorio remoto

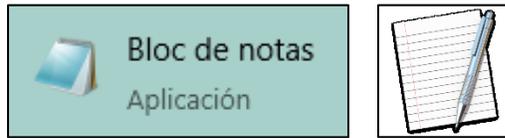
Ahora que tenemos listo el repositorio local en nuestra computadora, podemos empezar a trabajar. Para subir contenido a una rama del repositorio remoto, los archivos deben pasar por varias etapas en las que se actualizan y prueban los cambios que se han realizado sobre estos. Para ello se usan algunos comandos de Git que realizan esta transferencia:

- **git add -A** = Sube los cambios realizados a los archivos a una zona de pruebas para que Git empiece a rastrearlos. En esta zona de pruebas es donde uno puede revisar de que el código esté preparado para subirse al repositorio remoto.
- **git commit -m "Descripción del commit"** = Sube los archivos al repositorio local
- **git push origin [Rama]** = Sube los archivos de tu repositorio local al repositorio remoto

Con establecido esto, podemos empezar a subir contenido al repositorio que acabamos de crear.



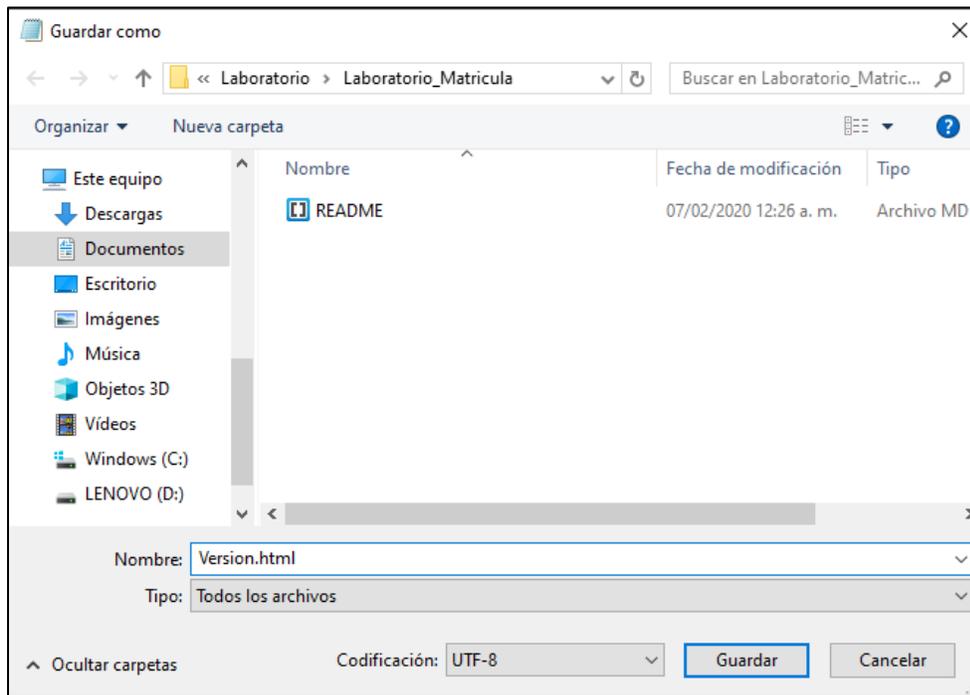
1) Abre el Bloc de Notas de Windows (TextEdit en MAC)



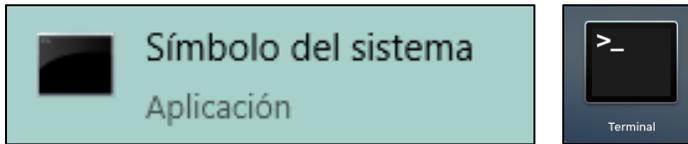
2) Copia el siguiente código en un archivo de texto nuevo

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Control de versiones</title>
  <meta charset="UTF-8">
</head>
<body>
  <main>
    <h1>Laboratorio de Github</h1>
  </main>
</body>
</html>
```

3) Guarda el archivo bajo el nombre “Version.html” en la dirección de tu repositorio local. Asegúrate de que en el campo “Tipo” esté seleccionada la opción “Todos los archivos”.

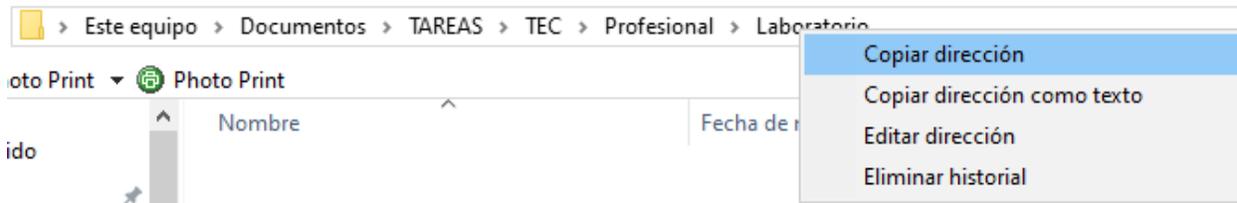


4) Abre la terminal de Windows



5) Accede al directorio en el que está tu repositorio local

- En el explorador de archivos entra a la carpeta. Una vez dentro, haz clic derecho sobre el nombre en la barra de dirección. Selecciona la opción 'Copiar dirección'



- Escribe el comando ***cd (Dirección de la carpeta)*** y estarás dentro de la carpeta

```

C:\> Símbolo del sistema
Microsoft Windows [Versión 10.0.18362.657]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.
C:\Users\rayro> cd C:\Users\rayro\Documents\TAREAS\TEC\Profesional\Laboratorio\Laboratorio_Matricula
C:\Users\rayro\Documents\TAREAS\TEC\Profesional\Laboratorio\Laboratorio_Matricula>
  
```

6) Inserta los siguientes comandos

```
git add -A
```

```
git commit -m "Versión 1"
```

```
git push origin master
```

7) Si aparece esta información en la terminal, entonces el commit se hizo correctamente

```

Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 488 bytes | 488.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/Raylogic/Laboratorio_Matricula.git
 c17f5d9..ab09ad2 master -> master
  
```

8) Regresa a Github para ver el archivo insertado en la rama “master”

The screenshot shows the GitHub repository page for 'Raylogic / Laboratorio_Matricula'. At the top, there are navigation links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security, Insights, and Settings. Below this, the repository name 'Laboratorio de Github - Nombre completo' is displayed with an 'Edit' button. A summary bar shows 9 commits, 2 branches, 0 packages, 0 releases, and 1 contributor. A 'Branch: master' dropdown and a 'New pull request' button are visible. Below this, there are buttons for 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A file list shows 'Raylogic Versión 1' (Latest commit 136790b 1 minute ago), 'README.md' (Initial commit 7 days ago), and 'Version.html' (Versión 1 1 minute ago).

9) Si quieres ver el contenido del archivo que subiste, haz clic sobre él. Se desplegará en una interfaz de editor de texto.

The screenshot shows the GitHub file viewer for 'Laboratorio_Matricula / Version.html'. The page title is 'Branch: master Laboratorio_Matricula / Version.html' with 'Find file' and 'Copy path' buttons. Below the title, it shows 'Raylogic Versión 1' (136790b 2 minutes ago) and '1 contributor'. The file content is displayed in a code editor with 12 lines (12 sloc) and 182 Bytes. The code is HTML:


```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <title>Control de versiones</title>
5   <meta charset="UTF-8">
6 </head>
7 <body>
8   <main>
9     <h1>Laboratorio de Github</h1>
10  </main>
11 </body>
12 </html>
  
```

 At the top right of the code editor, there are buttons for 'Raw', 'Blame', 'History', and icons for chat, edit, and delete.

VII. Obtener Github Student Development Pack

GitHub Student Development Pack es un paquete que le ofrece a alumnos varias herramientas de ingeniería de software (bases de datos, editores de texto, servidores, diseños web, dominios, etc...) usadas ampliamente en la industria para que puedan realizar sus propios proyectos. Como alumno del Tecnológico de Monterrey, tienes la posibilidad de adquirir este plan y usarlo para tus futuros trabajos de la carrera o en la vida profesional. A continuación, veremos cómo obtenerlo fácilmente.

- 1) Ingresa a la página: <https://education.github.com/pack>
- 2) Presiona el botón Get the Pack



[Home](#) / [Students](#) / GitHub Student Developer Pack

Learn to ship software like a pro.

There's no substitute for hands-on experience. But for most students, real world tools can be cost-prohibitive.

That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free access to the best developer tools in one place so they can learn by doing.

Get the Pack

Tweet

Like 57K

Before you receive access to the offers, we'll need to [verify that you are a student](#).

Teachers, researchers, faculty, and staff are not eligible for the Pack, but can [get free and discounted access to GitHub](#).

- 3) Se actualizará la página con una forma. Contesta las preguntas que se te piden.

NOTA: La forma pide que ingreses un mail institucional. Ingresa tu mail del TEC en el recuadro correspondiente

NOTA: La forma pide que subas una foto que pruebe tu estatus académico. Para efectos prácticos, sube una foto escaneada de tu credencial.

*Para dudas adicionales, consulta el siguiente link:

<https://help.github.com/es/github/teaching-and-learning-with-github-education/applying-for-a-student-developer-pack>

- 4) Una vez que hayas completado la forma, presiona el botón "Submit your information"
- 5) Si tu solicitud es aprobada, recibirás un mail de confirmación. Revisa tu correo electrónico de forma constante para ver si hubo respuesta.
- 6) Si deseas conocer la lista completa de beneficios que ofrece el paquete, regresa a la página <https://education.github.com/pack> y explora los servicios.