



Maestría en Ingeniería Electrónica con Énfasis en Tecnologías de la Información

ALUMNO

Ing. Jhonny L. Colman

MATERIA

Desarrollo de Sistemas Web

DOCENTE

Dr. Julio César Mello Román

TEMA DEL TRABAJO

GitHub y Git

San Lorenzo, Paraguay

2025

Contenido

1. Instalación del Cliente Git	3
<i>Paso 1:</i>	3
<i>Paso 2:</i>	3
2. Configuración de la Información del Usuario para Repositorios Locales	3
<i>Paso 1:</i>	3
3. Creación de una Cuenta en GitHub.....	3
<i>Paso 1:</i>	3
<i>Paso 2:</i>	4
4. Creación del Proyecto “Sistemas de Información Web”	4
<i>Paso 1:</i>	4
<i>Paso 2:</i>	4
5. Clonación del Proyecto en la Máquina Local	4
<i>Paso 1:</i>	4
6. Creación de una Rama a partir de la Rama Principal	4
<i>Paso 1:</i>	4
7. Creación de la Carpeta “Semana 01” y Adición de los Ejercicios	5
<i>Paso 1:</i>	5
<i>Paso 2:</i>	5
<i>Paso 3:</i>	5
8. Creación de la Carpeta “Semana 02” y Adición de los Ejercicios	5
<i>Paso 1:</i>	5
<i>Paso 2:</i>	5
9. Subir los Cambios a la Rama Remota.....	6
<i>Paso 1:</i>	6
10. Solicitar Merge Request	6
<i>Paso 1:</i>	6
<i>Paso 2:</i>	6
11. Mezcla de la Rama con la Rama Principal.....	6
<i>Paso 1:</i>	6
<i>Paso 2:</i>	6

1. Instalación del Cliente Git

Paso 1:

Se procede a la instalación del cliente Git en los ordenadores locales. Para ello, se accede a la página oficial de Git <https://git-scm.com/downloads> y se descarga el instalador correspondiente al sistema operativo (Windows, macOS o Linux).

Paso 2:

Se sigue el proceso de instalación guiado. Es importante aceptar las opciones por defecto, ya que estas configuraciones son las más recomendadas para el correcto funcionamiento de Git.

Una vez que se hayan realizado los pasos anteriormente mencionados el cliente Git estará instalado y disponible en la terminal o consola de comandos.

2. Configuración de la Información del Usuario para Repositorios Locales

Paso 1:

Tras instalar Git, se configura la información del usuario para todos los repositorios locales.

Abajo se muestra como configurar a través de la terminal con los siguientes comandos:

```
git config -- global user.name "jicolman"
```

```
git config -- global user.email "jhonnylcc@gmail.com"
```

Estos comandos configuran globalmente el nombre de usuario que para este caso es "jicolman" y el correo electrónico para este caso "jhonnylcc@gmail.com", los cuales serán asociados a los commits realizados en todos los repositorios locales.

Realizados estos pasos la información del usuario está configurada correctamente y se utilizará para cada repositorio en el que se trabaje.

3. Creación de una Cuenta en GitHub

Paso 1:

Se accede a <https://github.com/> y se crea una cuenta nueva proporcionando los datos requeridos (nombre de usuario, correo electrónico y contraseña).

Paso 2:

Tras la creación de la cuenta, se confirma mediante un correo de verificación y se completa el registro.

Así es como se cuenta con una cuenta activa en GitHub.

4. Creación del Proyecto “Sistemas de Información Web”

Paso 1:

En GitHub, se crea un nuevo repositorio. Se accede al panel de repositorios de GitHub y se hace clic en “Nuevo” para crear un repositorio.

Paso 2:

En el formulario de creación, se le asigna el nombre “Sistemas de Información Web”, se establece como repositorio público o privado según corresponda, y se inicializa con un archivo README.

Entonces se crea el repositorio “Sistemas de Información Web” en GitHub.

5. Clonación del Proyecto en la Máquina Local

Paso 1:

Se clona el repositorio recién creado en la máquina local utilizando el siguiente comando en la terminal:

```
git clone https://github.com/usuario/Sistemas-de-Informacion-Web.git
```

Este comando descarga una copia local del repositorio para poder trabajar en él.

El repositorio “Sistemas de Información Web” se encuentra clonado en la máquina local.

6. Creación de una Rama a partir de la Rama Principal

Paso 1:

Se accede al repositorio clonado y se crea una nueva rama a partir de la rama principal (master o main). Esto se realiza mediante el siguiente comando:

```
git checkout -b semana-01
```

Con esto se crea una rama nueva llamada "semana-01" y se cambia automáticamente a esa rama para comenzar a trabajar en ella.

La nueva rama "semana-01" ha sido creada y se está trabajando sobre ella.

7. Creación de la Carpeta "Semana 01" y Adición de los Ejercicios

Paso 1:

En la rama "semana-01", se crea una carpeta llamada "Semana 01" en el repositorio local.

Paso 2:

Dentro de esta carpeta, se agregan los ejercicios de la tarea 1 y la tarea 2. Se asegura que todos los archivos relevantes sean cargados dentro de la carpeta "Semana 01".

Paso 3:

Una vez añadidos los archivos, se guardan los cambios con los siguientes comandos:

```
git add Semana\ 01/
```

```
git commit -m "Agregar ejercicios de la semana 01"
```

Los archivos de las tareas 1 y 2 se encuentran dentro de la carpeta "Semana 01" y se han registrado los cambios en el repositorio local.

8. Creación de la Carpeta "Semana 02" y Adición de los Ejercicios

Paso 1:

Se crea una carpeta denominada "Semana 02" en el repositorio local y dentro de ella se añaden todos los ejercicios relacionados con el material de la diapositiva de JavaScript.

Paso 2:

Los archivos se agregan y se guardan mediante los comandos:

```
git add Semana\ 02/
```

```
git commit -m "Agregar ejercicios de la semana 02"
```

La carpeta "Semana 02" contiene los ejercicios de JavaScript y los cambios han sido registrados.

9. Subir los Cambios a la Rama Remota

Paso 1:

Se suben los cambios realizados a la rama remota en GitHub utilizando el siguiente comando:

```
git push origin semana-01
```

Los cambios de la rama “semana-01” se reflejan en el repositorio remoto en GitHub.

10. Solicitar Merge Request

Paso 1:

Se abre el repositorio en GitHub y se selecciona la opción de crear un *Pull Request* para fusionar la rama “semana-01” con la rama principal.

Paso 2:

Se agrega una descripción del trabajo realizado y se solicita la revisión del *Pull Request*.

El *Pull Request* es creado y está listo para ser revisado y fusionado.

11. Mezcla de la Rama con la Rama Principal

Paso 1:

El encargado de la revisión aprueba el *Pull Request* y procede a fusionar la rama “semana-01” con la rama principal (main o master) en GitHub.

Paso 2:

Una vez realizada la fusión, se elimina la rama “semana-01” de manera segura tanto local como remotamente con los siguientes comandos:

```
git branch -d semana-01
```

```
git push origin --delete semana-01
```

La rama “semana-01” ha sido fusionada exitosamente con la rama principal y eliminada tanto en local como en el repositorio remoto.