

UNIVERSIDAD NACIONAL DE ASUNCIÓN

FACULTAD POLITÉCNICA

**MAESTRÍA EN INGENIERÍA EN ELECTRÓNICA CON ÉNFASIS EN
TECNOLOGÍA DE LA INFORMACIÓN**

MÓDULO: SISTEMA DE INFORMACIÓN WEB

“Tarea 03: Reporte Técnico sobre GitHub y Git”

Alumno:

- **Oscar Aureliano Caballero Mendoza**

Profesor:

- **Dr. Julio César Mello Román**

23/03/2025

Reporte Técnico sobre GitHub y Git.

Introducción.

Este reporte técnico detalla el proceso paso a paso para completar la Tarea 03, que involucra el uso de **Git** y **GitHub** para el control de versiones y la gestión de proyectos.

Git es un sistema de control de versiones distribuido que permite rastrear los cambios en los archivos de un proyecto a lo largo del tiempo, mientras que **GitHub** es una plataforma de alojamiento de código que facilita la colaboración entre desarrolladores.

El objetivo de esta tarea es familiarizarse con los conceptos básicos de **Git** y **GitHub**, así como con el flujo de trabajo típico para desarrollar y compartir código en un entorno colaborativo.

Enlace para acceder a mi cuenta de GitHub.

Para verificar los ejercicios, favor ingresar en el siguiente enlace:

- <https://github.com/OACM79/Sistemas-Informacion-Web>.

1. Instalación del Cliente Git en el Ordenador.

Primeramente, se instaló el **Cliente Git** en el ordenador siguiendo los siguientes procesos:

- Se accedió al sitio web oficial de Git: <https://git-scm.com/downloads>.
- Se descargó el instalador correspondiente al sistema operativo utilizado (Windows, macOS o Linux/Unix). En mi caso el SO es Windows de 64 bits.
- Se ejecutó el instalador, siguiendo las instrucciones predeterminadas.
- Como resultado **Git** se instaló correctamente en el sistema del Ordenador.

2. Configuración de la Información del Usuario para todos los repositorios.

Para configurar la información del usuario, se siguieron los siguientes procesos:

- Ir a un directorio en una carpeta dada, hacer clic derecho y se observa en el sub menú **Git Bash Here**, que vamos a utilizar para ver en donde está posicionado la Carpeta.
- Luego se accede a la terminal o líneas de comandos donde vamos a usar y realizar las respectivas configuraciones.

A screenshot of a MINGW64 terminal window. The title bar shows the path 'MINGW64/c/Users/Melius/Documents/Maestria en Ingeniería Electronica - ...'. The terminal content shows the user 'Melius@DESKTOP-2SPV8TO' at the prompt 'MINGW64 ~/' with the directory path '/Documents/Maestria en Ingeniería Electronica - FPUNA'. The user enters the command '\$ pwd' and the output is '/c/Users/Melius/Documents/Maestria en Ingeniería Electronica - FPUNA'. The prompt '\$' is shown again on the next line.

- Como referencia utilizamos el siguiente enlace [github-git-cheat-sheet](#) con los siguientes comandos:
 - `git config --global user.name Oscar Caballero` (User Name).
 - `git config --global user.email oacm79@gmail.com` (User mail).

Con la ejecución de los 2 comandos, se configuró la información del usuario de **Git** para todos los repositorios locales.

3. Creación de Cuenta en GitHub.

Para la creación de cuenta en GitHub, se siguieron los siguientes procesos:

- Se accedió al sitio web de **GitHub**: <https://github.com/>.
- Se completó el formulario de registro con la información requerida (nombre de usuario, correo electrónico, contraseña).
- Se siguió el proceso de verificación de la cuenta.
- Como resultado se creó una cuenta en **GitHub**.

4. Creación del Proyecto "Sistemas de Información Web".

En GitHub inicialmente los que nos interesa es la parte de Repositorios (**Your repositories**), vamos a crear un Proyecto denominado **Sistemas-Informacion-Web**, con los siguientes Procesos:

- Se inició sesión en la cuenta de **GitHub**.
- Se hizo clic en el botón "**New**".
- En **Create a new repository**, se ingresó **Sistemas-Informacion-Web** como nombre del repositorio. Como opcional se puede poner una descripción como **Módulo de Sistemas de Información Web**.

- Se seleccionaron las opciones deseadas:
 - **Public.** Que sea Público para ser accedido por todos.
 - **Add a README file.** Donde generalmente se ponen las funcionalidades del código.
 - **Add.gitignore**, se agrega de acuerdo al lenguaje en que se estará trabajando, Ej: Java. Por el momento no se agrega ningún **gitignore**.
 - **Choose a license**, que tipo de licencia vamos a usar, Ej: GNU General Public License v3.0.

Se va a tener una rama **main** y se crea el repositorio utilizando el botón **Create repository**.

Ya creado el repositorio se observan que están los 2 archivos **LICENSE** y **README.md**.

En **README** se tienen, como se llama el Módulo que es **Sistemas-Informacion-Web** y la descripción del Módulo como **Módulo de Sistemas de Información Web**.

5. Clonación del Proyecto en la Máquina Local.

Para clonar en la máquina local vamos a utilizar el comando **git clone [url]**, que es para descargar un proyecto y todas sus historias de versión.

Para la clonación del Proyecto, los procesos a seguir son:

- En la Página de **GitHub**, en **Code** (Código) está la parte de clonar y se copió la URL o enlace del repositorio.
- Luego desde la terminal o línea de comandos se ejecutó el comando **git clone <https://github.com/OACM79/Sistemas-Informacion-Web.git>**.
- El proyecto **Sistemas-Informacion-Web** se clonó en la máquina local.
- Para verificar si se descargó bien, desde la terminal se utiliza el comando **ls -a**.

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

Desde la terminal o línea de comandos, se entra al **Proyecto Sistemas-Informacion-web** con el comando **cd Sistemas-Informacion-Web/** y se observa la rama principal que es **main**.

```
Oscar@DESKTOP-L5UI4UM MINGW64 ~/Documents/Maestria FP-UNA/Sistemas-Informacion-Web (main)
$
```

Para crear una rama a partir de la rama principal **main** se ejecutó el comando **git checkout -b feature/disenoweb**, donde **feature/disenoweb** es el nombre de la rama.

```
Oscar@DESKTOP-L5UI4UM MINGW64 ~/Documents/Maestria FP-UNA/Sistemas-Informacion-Web (main)
$ git checkout -b feature/disenoweb
```

Se creó la rama llamada **feature/disenoweb** a partir de la rama principal **main**.

```
Oscar@DESKTOP-L5UI4UM MINGW64 ~/Documents/Maestria FP-UNA/Sistemas-Informacion-Web (feature/disenoweb)
$
```

Para cambiar de ramas, se realizan:

- Si no hay cambios en las ramas, primeramente, se vuelve a la rama principal **main** con **git checkout main**.
- Luego se elige la rama donde se quiera acceder, en este caso la que se creó **feature/disenoweb** con **git checkout feature/disenoweb**.

7. Creación de la Carpeta "Semana 02" y Levantar los Ejercicios.

Dentro del directorio del **Proyecto Sistemas-Informacion-Web**, se creó una carpeta llamada **Semana 02**.

Dentro de la carpeta **Semana 02**, se copió el archivo **Tarea 03_ GitHub y Git** que son las instrucciones de la **Tarea 03**.

Desde la terminal o línea de comandos, se verifica el archivo modificado:

- **git status** (se observa en rojo el archivo que se haya modificado)

Para levantar la rama, se siguen los siguientes procesos:

- **git add .** (agrega todos los archivos modificados y nuevos que se van a levantar).
- **git status** (se observa en verde, ya está preparado para ser levantado el archivo).
- **git commit -m "feat:main Se levanta documento para las instrucciones de la Tarea 03"** (Se guardan los cambios que se preparó con **git add .**).

Se creó la carpeta **Semana 02** y se levantó el documento **Tarea 03_ GitHub y Git** correspondiente a la rama **feature/disenoweb**.

8. Levantar Rama.

Para levantar la rama **feature/disenoweb**, se ejecutó el siguiente comando:

- **git push origin feature/disenoweb** (Se levantan todos los archivos desde la rama local **feature/disenoweb** y los envía al repositorio remoto **origin**).

La rama **feature/disenoweb** se levantó al repositorio remoto en **GitHub**.

9. Solicitar Merge Request.

Se accedió al repositorio en **GitHub**, ya se observa la nueva rama **feature/disenoweb** y se hizo clic en el botón **Compare & pull request**.

Se revisaron los cambios, se agregó un título y una descripción para la solicitud.

Add a title

feat:main Se levantar documento para las instrucciones de la tarea 03

Add a description

Write Preview

Se levantar documento para las instrucciones de la tarea 03

Markdown is supported Paste, drop, or click to add files

Create pull request

Luego se hizo clic en el botón **Create pull request**.

Se creó una solicitud de **Merge Request** para fusionar la rama **feature/disenoweb** con la rama principal **main**.

10. Mezcla de la Rama con la Rama Principal.

Si no hay conflicto y está todo en verde, se hizo clic en el botón **Merge pull request**.

Open feat:main Se levantar documento para las instrucciones de la tarea 03 #1
jomello wants to merge 1 commit into main from feature/disenoweb

Se levantar documento para las instrucciones de la tarea 03

feat:main Se levantar documento para las instrucciones de la tarea 03 221aa81

No conflicts with base branch
Merging can be performed automatically.

Merge pull request You can also merge this with the command line. [View command line instructions](#)

Para confirmar se hizo clic en el botón **Confirm merge**.

Commit message

Merge pull request #1 from jcmello/feature/disenoweb

Extended description

feat:main Se levantar documento para las instrucciones de la tarea 03

This commit will be authored by prof.juliomello@gmail.com.

Confirm merge Cancel

Add a comment

Write Preview

H B I

Add your comment here...

La rama **feature/disenoweb** se fusionó con la rama principal **main**, incorporando los cambios de la carpeta **Semana 02** al proyecto principal **Sistema-Informacion-Web**.

Creación de la Carpeta "Semana 01" y Levantar los Ejercicios.

Para crear la carpeta Semana 01 y subirla, se siguieron los siguientes procesos:

- Se posiciona nuevamente sobre la rama principal **main** con (**git checkout main**).
- Se actualiza la rama principal **main** con **git pull**.
- Se crea una rama a partir de la rama principal **main** con **git checkout -b feat/disenoweb**.
- La nueva rama se llama **feat/disenoweb**.
- Dentro del directorio del **Proyecto Sistemas-Informacion-Web**, se creó una carpeta llamada **Semana 01**.
- Dentro de la carpeta **Semana 01**, se copiaron los archivos de los ejercicios de **Tarea01** y **Tarea02**.
- Se preparó los archivos a ser levantados con **git add ..**
- Se guardaron los cambios que fueron preparados con **git commit -m "feat:main Se levantan las Tareas 01 y 02 de la Semana 01"**.
- Se levantaron todos los archivos desde la rama local **feat/disenoweb** y se enviaron al repositorio remoto con **git push origin feat/disenoweb**.
- Se accedió al repositorio en **GitHub**, y se observa la nueva rama **feat/disenoweb**.
- Se solicitó **Merge request** para fusionar la nueva rama **feat/disenoweb** con la rama principal **main**.

- Si no hay conflictos se confirma el **marge**.

La rama **feat/disen** se fusionó con la rama principal **main**, incorporando los cambios de la carpeta **Semana 01** al proyecto principal **Sistema-Informacion-Web**.

Conclusión:

Este informe detalla el uso de **Git** y **GitHub** para gestionar el proyecto "**Sistemas-Informacion-Web**". Se cubren los pasos iniciales: instalación, configuración de usuarios, creación de repositorios y clonación local. La configuración adecuada de **Git** y la creación de un repositorio remoto en **GitHub** facilitan la colaboración y el seguimiento de cambios desde el inicio.

Se explica el flujo de trabajo con ramas ("**feature/disenoweb**" y "**feat/disen**"), permitiendo el desarrollo aislado de funcionalidades y pruebas exhaustivas. El uso de "**Pull Requests**" asegura la revisión del código y la calidad antes de la fusión con la rama principal. La fusión de ramas y la capacidad de revertir cambios garantizan un control total sobre el desarrollo del proyecto.

La aplicación de **Git** y **GitHub** en el proyecto "**Sistemas-Informacion-Web**" mejora la productividad y fomenta la transparencia y responsabilidad en el desarrollo de software. Estas herramientas optimizan el flujo de trabajo colaborativo, asegurando la integridad del código y facilitando la gestión de versiones a lo largo del tiempo.

Bibliografía.

- **Acerca de GitHub y Git.** <https://docs.github.com/es/get-started/start-yourjourney/about-github-and-git>.
- **Creación de una cuenta en GitHub.** <https://docs.github.com/es/getstarted/start-your-journey/creating-an-account-on-github>.
- **Hoja de referencia para GitHub Git.** https://training.github.com/downloads/es_ES/github-git-cheat-sheet.pdf.
- **Hola mundo.** <https://docs.github.com/es/get-started/start-your-journey/hello-world>.