



***Seminario de Solución de Programación de sistema
embebidos***

Daniel Giovanni Martínez Sandoval

“Practica 3”

INRO

Por:

Donnovan Said Santoyo Meza 219520552

David Dario Castro Zaragoza 219535711

Raúl de Jesús Quiroz Rincón 219700917

Índice

<i>1. Introducción.....</i>	<i>3</i>
<i>2. Desarrollo.....</i>	<i>3</i>
<i>2.1 Marco teórico.....</i>	<i>4</i>
<i>2.2 Metodología/Procedimiento</i>	<i>5</i>
<i>4. Conclusión</i>	<i>6</i>

Introducción

En esta práctica, se llevó a cabo la configuración y gestión de un repositorio en GitHub utilizando el protocolo SSH. El objetivo principal fue aprender a crear un repositorio, enlazarlo con una máquina local y administrar cambios mediante el sistema de control de versiones Git. Esto permite trabajar de manera organizada en proyectos, asegurando un control eficiente de los archivos y versiones.

Desarrollo

1. Marco teórico

Github

GitHub es una plataforma de desarrollo colaborativo que permite almacenar, gestionar y compartir proyectos mediante el uso del sistema de control de versiones Git. Git facilita el seguimiento de cambios en los archivos a lo largo del tiempo, permitiendo restauraciones y colaboración efectiva entre múltiples desarrolladores.

SSH (Secure Shell) es un protocolo que permite una conexión segura entre una computadora local y GitHub. Al configurar una clave SSH, se habilita una autenticación segura sin necesidad de ingresar credenciales en cada interacción con el repositorio remoto.

2. Metodología/Procedimiento

Para comenzar, se creó una cuenta en GitHub y posteriormente un repositorio público con un archivo README.md. Luego, se configuró la autenticación SSH generando una clave mediante ssh-keygen, agregándola a GitHub y verificando la conexión con `ssh -T git@github.com`.

Una vez establecida la conexión, se clonó el repositorio con `git clone` y se configuró Git con el correo y nombre de usuario. Se creó un archivo `ejemplo.txt` dentro del repositorio y se realizó un primer commit con `git add`, `git commit -m` y `git push`, enviando los cambios al repositorio remoto.

Para mantener un flujo de trabajo organizado, se creó una nueva rama `practica_n` y en ella se añadieron carpetas para código y reporte. Dentro de estas carpetas, se generaron archivos de ejemplo, se confirmaron los cambios con `git status`, `git add`, `git commit -m` y se subieron con `git push --set-upstream origin practica_n`.

Finalmente, se realizó un pull request en GitHub para fusionar los cambios con la rama principal, asegurando que la práctica quedara correctamente integrada en el repositorio.

Conclusión

El uso de Git y GitHub con autenticación SSH permite una gestión eficiente y segura del código, facilitando el trabajo colaborativo y la organización de proyectos. La implementación de ramas permite un control estructurado de los cambios sin afectar la rama principal hasta que se considere oportuno. Estas herramientas son esenciales en el desarrollo de software, garantizando la integridad y trazabilidad del código fuente.