

COMUNICACIONES II – 27145 INTRODUCIÓN A GITHUB

JUSTIFICACIÓN

Debido a que en la actualidad el trabajo colaborativo ha venido tomando cada vez más fuerza, es necesario que los estudiantes aprendan a trabajar con plataformas que ofrecen herramientas que permiten que el trabajo sea muy sencillo, y que les permita acceder a múltiples funcionalidades.

GitHub se presenta como una de las herramientas más utilizadas en la actualidad para poder actualizar documentación, proyectos y objetivos claros en diferentes organizaciones.

Objetivo general

Utilizar herramientas para comunicación colaborativa que permitan mostrar actualizaciones de los contenidos al usuario.

Objetivos específicos

- Crear un repositorio (y una cuenta si no la tuviera) en GitHub para el desarrollo de la parte práctica de la materia.
- Generar una llave ssh en su estación de trabajo para configuración de la terminal de Linux y establecer tal llave para su repositorio en GitHub.
- Clonar su repositorio en un directorio específico para el desarrollo del trabajo durante el semestre.
- Realizar cambios en la estación y subirlos por terminal de Linux para verificar que queden en línea.

Software a utilizar

- Linux (en lo posible Ubuntu).
- Terminal de Linux.
- Git (dentro del terminal).

Procedimiento

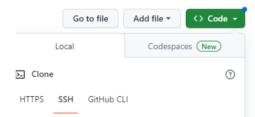
- 1. Si no tiene una cuenta en GitHub, se recomienda crear una. Para esto, abra un navegador web y visite el siguiente vínculo: https://github.com/. Allí suscríbase a una cuenta o utilice una de sus cuentas de correo para ingresar (se sugiere utilizar la cuenta de correo institucional).
- A partir de la cuenta de github, se solicita que se cree un repositorio para el trabajo del semestre como muestra la siguiente guía: https://docs.github.com/es/get-started/quickstart/create-a-repo. Utilice el nombre CommunicationsII_2024_1_Name (Name debe ser reemplazado por el nombre del grupo a utilizar en todos los casos de aquí en adelante).
- 3. Después de creado el repositorio se solicita trabajar en la terminal de Linux. Antes de ejecutarla, es necesario que ingrese a la sesión de usuario dada en tal laboratorio (la contraseña está escrita en papel en la CPU). La aplicación "terminal" de Linux la puede



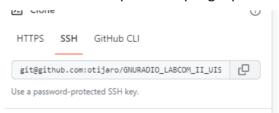


encontrar en el menú o ventana de aplicaciones (es similar a la ventana de comandos de Linux).

- 4. Al ejecutar la terminal, genere una llave ssh para poder clonar el repositorio que creo en GitHub, sin embargo, ino se adelante!, ejecute cada uno de los pasos que aquí le recomendamos: utilice el siguiente tutorial https://docs.github.com/es/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent (vaya al título cómo generar claves SSH localmente) para la creación de la llave ssh.
 - **Nota:** Por favor no utilice las traducciones automáticas de los navegadores para que no se distorsionen los comandos que debe incluir en la terminal.
- 5. Vaya al punto 7 que lo lleva a otro vínculo. Si por algún motivo, al añadir la llave no le funciona, pudo ser porque el comando "clip" no operó correctamente, por favor utilice uno de los editores de texto que más le convengan para ello (por ejemplo: gedit, cat o vim).
- 6. En este punto ya debe tener la llave ssh creada y activa en su cuenta de Github.
- 7. Cree a partir del comando mkdir los directorios necesarios para almacenar la información y descargar su repositorio (directorio sugerido: /home/com2_B2/2024_1/Laboratorio). Verifique que esta ruta se pueda ejecutar en el PC asignado.
 - **Nota**: Recuerde que el directorio /home/com2_B2 es al que se asigna la sesión en la que usted ingresó. Si usted ingresa o crea un directorio diferente, en una ruta diferente entonces tendrá problemas de autenticación de la llave.
- 8. Después de creado el directorio, utilice el comando cd para acceder al directorio o a los directorios de la ruta generada en el paso anterior.
- 9. Ahora, vuelva al navegador a su repositorio de GitHub (el que creó en el punto 2), haga clic en el botón que dice <>Code:



10. Y busque que la pestaña SSH del menú que se despliega quede seleccionada.



11. Copie lo que le indica la página. Y en el directorio creado en el paso 7 y direccionado en el paso 8, ejecute el comando:

git clone git@github.com.. (las letras en azul las cambia por lo copiado en el paso 10).

Nota: en las ventanas de terminal puede utilizar la combinación de teclas Shift + Insert para pegar lo copiado en el navegador.

- 12. Si le indican como crear el archivo Readme.md hágalo, verifique que esta información esté quedando almacenada correctamente en la nube (plataforma GitHub del navegador).
- 13. Si encuentra mensajes de error, por favor diríjase al docente o al auxiliar.
- 14. En la terminal de Linux, por favor ejecute el comando git pull.







- 15. Verifique con el comando ls que los archivos que ve en la nube, son efectivamente los que están en su estación de cómputo.
- 16. Si el punto 15 es correcto, utilice una imagen o un mensaje de prueba y guárdelo en la carpeta de trabajo.
- 17. Utilice el comando "git status" para verificar que hubo cambios de manera local en su PC.
- 18. Si en la ventana de comandos se encontraron cambios, se mostrará en color rojo los archivos que cambiaron (se agregaron o quitaron). Si está de acuerdo con esos cambios, utilice el comando "git add ."
- 19. Si no encuentra errores, ejecute el comando "git commit -m "mensaje" ". El "mensaje" debe ir entre comillas y generalmente es un comentario que hace referencia a los archivos que subió (o eliminó) o el porqué hizo los cambios. Si encuentra errores ejecute el paso 13.
- 20. Ejecute el comando git push y verifique que los cambios hayan sido subidos a la nube (plataforma GitHub en el navegador).

Debido a que esta herramienta se configura el primer día de clases prácticas, no es necesario hacer un informe, pero por cada asistencia al laboratorio si es necesario generar una rama a partir de la rama maestra. Se sugiere adicionalmente que cada práctica esté asociada a un directorio diferente y donde se utilicen los desarrollos de Gnu Radio.

Un repaso sencillo de los comandos de git:

Download	Upload
git clone: Se utiliza una única vez para descargar el repositorio. Y este crea una copia en el directorio local que usted tiene (por defecto se descarga de la rama "main o master"). git pull: Se utiliza después que se ha descargado el repositorio y sirve para que los cambios de la rama en la que	git add: Allí se preparan los cambios que se deben subir a la rama, pero no son subidos aún, solo se dejan listos para subir. git commit: En este comando se sugiere agregar un mensaje ("- m") para identificar los cambios que va a subir y que se tenga un registro de los cambios. Esto genera una forma ordenada de trabajar en un proyecto. git push: En este caso, se suben los archivos que
está trabajando puedan ser actualizados en su estación de trabajo.	fueron cargados con "git add" y registrados con "git commit", allí ya son subidos al espacio en github de la nube.
Generales	

git status: sirve para identificar que cambios se han realizado en su rama.

git checkout: sirve para poder navegar en otra rama.

git branch -a: consulta las ramas que tiene el respositorio y le indica en cual rama esta (en el caso de tener más de una rama).

Muchos más comandos se utilizan y se actualizarán a medida que se vaya avanzando en los laboratorios, por favor téngalos en cuenta.

