**INTEGRACION CONTINUA   
SEGUNDA ENTREGA**

TUTOR:

Natalia Martínez Rojas

PRESENTADO POR:

Iván Rene Barrios Ávila

Robinson Daniel Clemente Córdoba

Ingrid Lorena Rivera Pineda

Manuel

 Institución Universitaria Politécnico Grancolombiano

Facultad de Ingeniería, Diseño e Innovación

Integración Continua

Colombia

2021

**TABLA DE CONTENIDO**

[CREACION DE PROYECTO Y CONTENEDORES 3](#_Toc73483178)

[Configuración inicial 3](#_Toc73483179)

# CREACION DE PROYECTO Y CONTENEDORES

Para el proyecto actual se ha creado un repositorio en github, en <https://github.com/Robinson-Clemente/IntegracionContinuaModulo> donde se ha creado un archivo de configuración inicial, *README.md* en el cual se especifica la configuración y creación de dos contenedores Docker a partir de imágenes de Linux Ubuntu.

Dichos contenedores se han creado en los siguientes repositorios:

<https://hub.docker.com/r/robinsonclemente12/maquina1>

<https://hub.docker.com/r/robinsonclemente12/maquina2>

## Configuración inicial de Git

El primer paso consiste en la instalación de Git, el cual es un cliente para el Sistema Operativo que funciona mediante una *terminal*, o línea de comandos, si bien existen clientes gráficos con GUI.

Una vez se tiene el cliente, un proyecto nuevo surge con el envío de los primeros archivos a la rama principal de un repositorio de Github, rama que en nuestro caso hemos llamado *main*. La rama o *branch* es una copia del proyecto con los archivos que tienen un estado en un instante determinado de tiempo.

Para copiar a cada uno de los computadores de manera local, usando credenciales de acceso de Git, se ejecuta el comando *git clone*, que copiara a una carpeta el contenido del repositorio de la rama main.

Posteriormente, de manera local, cada archivo es modificado. La diferencia que se genera, con los archivos agregados y/o modificados se puede ver mediante el comando *git status*.

Si uno de esos archivos quiere ser compartido en esa misma rama, los cambios se confirman de manera local con el comando *git add <ruta/archivo>.*

Cada modificación o agregación de uno o más archivos requiere un registro que resuma los cambios que se han hecho (control de cambios), por lo cual es buena practica agregar un comentario al mismo. Dicho registro se lleva mediante el log de modificaciones o *commits,* y un nuevo commit se agrega mediante el comando *git commit -m “<comentario\_de\_cambios>”.*

Finalmente, para enviar esos cambios locales al repositorio en Github, se realiza mediante el comando *git push origin main,* que actualizara los archivos que hacen parte de esa nueva modificación o commit.

Ahora bien, si el proyecto ya se ha clonado, y se requiere actualizar de manera local dichos archivos, el comando *git pull origin main* hará lo pertinente.