MS2

COMANDOS

C:\Users\usuario>

C:\>cd cursoa

C:\cursoa>dir

C:\alumnos>tree

C:\alumnos>md c:\cursoa\oython

C:\Windows>cd ..

Dir

Variables de entorno; variables que se almacenanen el SO

Tipo de arquitectura:

Mvc Modelo vista controlador

Path : variable de entorno que permite ejecutar desde

SGBD sistema de gestión de bases de datos

Proyectos

-Entornos

Entorm¡no de UAT

Desarrollo/ Preacectación /Aceptación y Producción

Replica lo mas parecida posible a Producción.

Documentación

HTML Lenguaje de marcado, no de programación

Interface de usuario:

Pero es ejecutable en WWW

Con Notepad++ con Ext. HTML

<html>

<head>

<title>Esto es mi primera pagina HTML</title>

</head>

<body>

<h1>esto es un encabezado h1</h1>

<H2>hola es un encabezado h2</h2>

<p>Esto</p> <p>es</p> <p id="identificador" title="valor deltitle de mi" >mi</p> <p>primera</p> <p>pagina</p> <p>HTML</p>

<img src="c:\alumnos\cursos\nuevo\gatitos.jpg" alt="no encuentra imagen"/>

<ol/> type"I"------ Identifica la lista

<li>alexis</li>---------

<li>ANA</li>

<ol/>

</body>

</htmi>

Protocolo://ip: PUERTO/ Ruta al recurso

https:// 192.168.2.1: xxxxx a que maquina de Internet te tienes que conectar

Formularios:

Se envía cadenas a través de botones ya prefijados

Con post se oculta la información mandada en los formularios

Con get lo masusual, pero cuando no se quiere que se vea la información post

El number en el valor input sube y baja valores numericos

<html>

<head>

<title>Esto es un formulario</title>

</head>

<body>

<fieldset>

<legend>Biografia: de este niño o niña: </legends>

<form action="#" method="get">

<table>

<tr>

<td><label for="nombre">Nombre:</label></td>

<td><input type="text" name="nombre"/></td>

<label for="nombre">Nombre:</label/>

<input type="text" name=nombre"/>

<br/>

<td><label for="edad">Edad:</label></td>

<td><input type="number" name="edad"/></td>

</tr>

<label for="edad">Edad:</label/>

<input type="number" name=edad"/>

<br/>

<td><label for="cajapassword">cajapassword:</label></td>

<td><input type="password" name="cajapassword"/></td>

</tr>

<label for="cajapassword">cajapassword:</label/>

<input type="password" name=cajapassword"/>

<br/>

<td><label for="color">Favorito:</label></td>

<td><input type="color" name="colorfav"/></td>

</tr>

<label for="color<rfav">Favorito:</label/>

<input type="color" name=colorfav"/>

<br/>

<td><label for="fecha">Fecha de nacimiento:</label></td>

<td><input type="date" name="fecha"/></td>

</tr>

<label for="fecha">Fecha de nacimiento:</label/>

<input type="date" name=fecha"/>

<br/>

<td><label for="unFichero">Fotografia:</label></td>

<td><input type="file" name="unFichero"/></td>

</tr>

<label for="unFichero">Fotografia:</label/>

<input type="file" name=unFichero"/>

<br/>

<td><label for="sexo">Sexo:</label></td>

<td><input type="sexo" name="Monjil"/></td>

<td><input type="sexo" name="Energumeno"/></td>

<td><input type="sexo" name="Cero Patatero"/></td>

</tr>

<label for="sexo">Sexo:</label/>

<input type="radio" name=sexo" value="mucho">Monjil</input>

<input type="radio" name=sexo" value="poco">Energumeno</input>

<input type="radio" name=sexo" value="nada">Cero Patatero</input>

<br/>

<td><label for="esSoltero">Soltero:</label></td>

<td><input type="checkbox" name="Soltero"/></td>

</tr>

<label for="esSoltero">Soltero:</label/>

<input type="checkbox" name=sexo" value="Soltero"/>

<br/>

<label for="esEstudiante">Estudiante:</label/>

<input type="checkbox" name=esEstudiante" value="Estudiante"/>

<br/>

<label for="provincia">Provincia de Nacimiento:</label/>

<select name="provincia">

<option value="1">Toledo</option>

<option value="2">Teruel</option>

<option value="3">Soria</option>

</select>

</br><br/>

<label for="provinciavisitada">Provincia Visitada:</label/>

<select title="pulse CTRL para realiza¡r selecion multiple"

name="provinciavisitad"

multiple>

<option value="1">Toledo</option>

<option value="2">Teruel</option>

<option value="3">Soria</option>

</select>

</br>

<input type="submit" value="Eviar Datos"/>

</form>

</body>

</html>

<html>

<head>

<title>Esto es el titulo de mi pagina</title>

</head>

<body>

<a name="inicio"></a>

<a href="#ir arriba"></a>

<a href="#gatitos">ir arriba</a>

<hr/>

<img src="c:\alumnos\cursos\gatitos.jpg" alt="no encuentra imagen"/>

<hr/>

<a href="#cuarta">ir arriba</a>

<hr/>

<img src="c:\alumnos\cursos\html\NAVEGACION\galeria\cuarta.jpg" alt="no encuentra imagen"/>

<hr/>

<a href="#primera">ir arriba</a>

<hr/>

<<img src="c:\alumnos\cursos\html\NAVEGACION\galeria\primera.jpg" alt="no encuentra imagen"/>

<hr/>

<a href="#tercera">ir arriba</a>

<hr/>

<img src="c:\alumnos\cursos\gatitos.jpg" alt="no encuentra imagen"/>

<hr/>

<a href="#cuarta">ir arriba</a>

<hr/>

<img src="c:\alumnos\cursos\html\NAVEGACION\galeria\cuarta.jpg" alt="no encuentra imagen"/>

<hr/>

<a href="#primera">ir arriba</a>

<hr/>

<<img src="c:\alumnos\cursos\html\NAVEGACION\galeria\primera.jpg" alt="no encuentra imagen"/>

<hr/>

<a href="#tercera">ir arriba</a>

<hr/>

<img src="c:\alumnos\cursos\gatitos.jpg" alt="no encuentra imagen"/>

<hr/>

<a href="#cuarta">ver gato</a>

<hr/>

<img src="c:\alumnos\cursos\html\NAVEGACION\galeria\cuarta.jpg" alt="no encuentra imagen"/>

<hr/>

<a href="#primera">ir arriba</a>

<hr/>

<<img src="c:\alumnos\cursos\html\NAVEGACION\galeria\primera.jpg" alt="no encuentra imagen"/>

<hr/>

<a href="#tercera">ir arriba</a>

<hr/>

<a href=""ir arriba"></a>

<body>

<html>

CURSO 09/01/20

Para UBUNTU usuario forma2

**VirtualBox**

**Notepad++**

### Github <https://github.com/MATT68> [FullStack](https://github.com/MATT68/FullStack)

**LINUX**

**Ls listar directorio**

**Cd cambiar directorio**

**Mkdir crear directorio**

**Cp [ ] copiar directorio / generar una copia / -i pregunta antes de copiar**

**Rmdir o rm [ -r] borrar directorio**

**Mv renombar / mover / pasar un fichero de un lado a otro**

**Wc / contar líneas**

**Tail /**

**Su / a usuario root cambiar de usuario**

**Passwd cambiar de password**

**Sudo lanzar LOS COMANDOS COMO USUARIO ROOT**

**Sudo su cambiar como root sin cambiar la password**

**Chmod cambiar las asignaciones de lectura escritura o**

**Chown**

**Directorio BIN con todos los Comandos diferentes**

**Useread**

**Ej de Alias;**

**Alias lf=”ls lhFr –group-directories-first”**

**Grep / para buscar algo**

**Kill / cargarse algo**

**Ifconfig / para que indique parámetros de conexión de mqV**

# Download Linux Mint 19.3 Tricia

<https://github.com/MATT68/FullStack>

mi Ubuntu red 172.16.8.62

mi Windows

Buscar procesos activos;

Ps – ef | tail

Kill / borrar

ASCII  **American Standard Code for Information Interchange**

**EBCDIC = CCSID**

Extended Binary Coded Decimal Interchange Code

**coded character set identifier**

**U**NICODE estándar

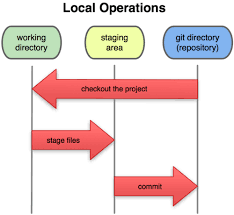
Sourcer ( Recursos )

Gestor de Repositorios (fuentes/documentos)

TAG

GITHUB

Describiendo el flujo de trabajo en Git

[](https://www.google.com/imgres?imgurl=https%3A%2F%2Fmiro.medium.com%2Fmax%2F500%2F0*HeEKExh4Z0nlev1m.png&imgrefurl=https%3A%2F%2Fmedium.com%2Flaboratoria-how-to%2Fdescribiendo-el-flujo-de-trabajo-en-git-ede2eee5b589&docid=VvYyWp2pcSiJLM&tbnid=LaCPCny0lq5JsM%3A&vet=10ahUKEwiB1cWhgYjnAhUOy6QKHQrWAFgQMwhFKAEwAQ..i&w=500&h=460&bih=570&biw=1188&q=procesos%20en%20git&ved=0ahUKEwiB1cWhgYjnAhUOy6QKHQrWAFgQMwhFKAEwAQ&iact=mrc&uact=8)

Git init / crea un archivo .git

**git add .** /// meter un fichero en staging area  
**git commit -m "Nombre de la nueva version"** /// almacenar en el repositorio local lo que se ha creado con el “git add”  
**git push -u origin master ///**

Como podemos ver, nuestro repositorio local **(el folder que clonamos a nuestra computadora)** tiene 3 diferentes fases por la cual pasa nuestro código:

1. **Working Directory**
2. **Staging Area**
3. **Git Repository**

Al igual que las 3 fases, el código tiene 3 estatus en cada diferente fase:

1. **Modificado**
2. **Preparado**
3. **Confirmado**

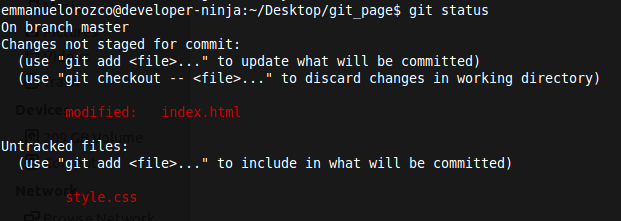
## **Fase 1: “Working Directory”.**

Aquí es donde podemos hacer cualquier cambio y no afectar nuestro repositorio en lo absoluto.

En cuanto modificamos algo de nuestro código, éste tiene status de **modificado**.

Si ejecutamos el comando **git status,**nos mostrará qué archivos han sido modificados (o creados).





“Untracked file” es para archivos recién creados y “modified” para los modificados

Una vez que hemos hecho los cambios necesarios, pasamos nuestros archivos al “**staging area**” con el comando:

**git add nombreDelArchivoModificado.js**

Nota que no es necesario escribir:

**git add nombreDelArchivoModificado.js**

si escribimos:

**git add .**

Agregamos todos los archivos modificados dentro de **working directory** a **staging area**.

Cuando pasas el código de Working Directory a Stagin Area, cambias el estado del código de **modificado** a **preparado**.

## Fase 2: “Staging Area”

Aquí es donde le podemos dar nombre a nuestra nueva versión. Y crear una “copia” de cómo quedaría nuestra nuestro repositorio en producción.

Para pasar nuestro código de staging area al Git Repository (aun no se publica el código en Github), escribimos el siguiente comando:

git commit -m "Nombre del la nueva versión"

Nota que cuando haces el commit el código pasa de estado **preparado** a **confirmado**.

## Fase 3. “Git repository”

Una vez que el código esta confirmado ya esta listo para sincronizarse con el servidor de Git (github, bitbucket,etc). Para hacer esto, escribimos:

git push -u origin master

Como truquito adicional, te cuento que **-u**sirve para **recordar** “**origin**” que es la dirección de tu repositorio de git y “**master**” que es la branch del repositorio que se esta usando.

Como estamos “recordando” origin y master, para el próximo pusheo, ú**nicamente** podríamos hacer:

git push

Y git ya sabe que significa:

git push origin master

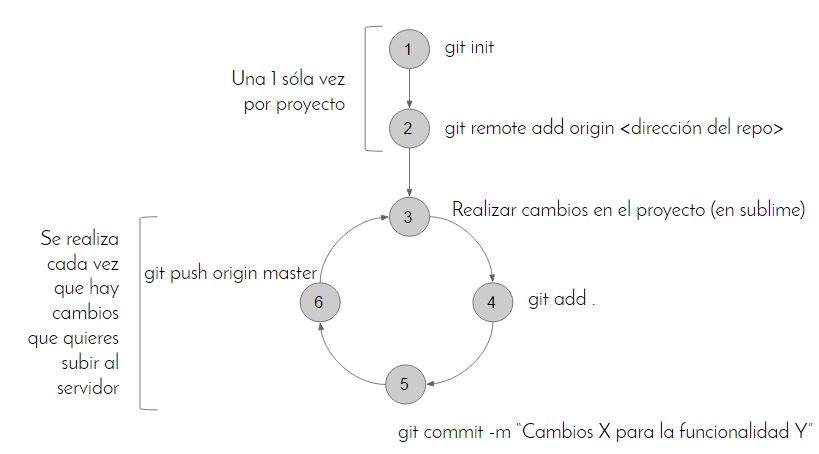
Si estas trabajando en un repositorio con varias personas en algún momento vas a querer descargar las cosas que tus compañeros agreguen o modifiquen.

Para eso ejecutamos:

git pull origin master

Y git automáticamente agrega los cambios, (y se pasa por el proceso **checkout the proyect**, mostrado en el diagrama de arriba).

Este nuevo diagrama, te puede ayudar entender un poco mejor el flujo que acabamos de explicar.



## **Y ¿cuáles son las ventajas de tener add y commit? ¿No podría solo existir una fase para sincronizar mi código con github?**

Buena pregunta, en teoría sí, se podría tener solo una fase. Sin embargo existe una ventaja fundamental en tener los 2 procesos separados.

Esta ventaja es poder nombrar CUALQUIER cambio de una forma especifica.

Por ejemplo:

Si creamos 2 archivos, un archivo index.hmtl y otro archivo funciones.js, para agregarlos a Git, podríamos escribir:

git add .  
git commit -m "Agregado index y Javascript"

Pero también podemos hacer esto:

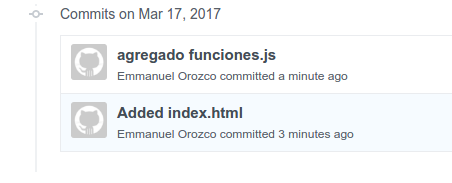
git add index.html  
git commit -m "Agregado HTML"git add funciones.js  
git commit -m "Agregado JS"

Y pushear con un simple:

git push origin master

Y en github al momento de pushear nuestros commits se verian separados y por tanto, es mucho mas legible nuestro control de versiones:





Ahora que hemos visto cuales son las fases y estatus de nuestro código de Git, podemos con mas confianza ejecutar comandos y perder el miedo a romper cosas.

GITHUB

Usuario ae08497

Password aluche1968

<https://github.com/ae08497/RepositorioGoyo.git>