Clase 1

**GitHub – GitLab:** Son plataformas en la nube que permiten almacenar el código fuente o proyectos

Comandos para configurar git desde consola

Lo siguiente servirá para configurar git con un usuario especifico y son obligatorios para subir la información a la nube (forma global)

1. Instalar Un git- scm (gestor de código fuente)—es un servicio: es un software que realiza seguimiento a los cambios de versiones realizados a lo largo del tiempo
2. Configurar servicio de git-scm

* git config --global user.name “Nombre de usuario”
* git config --global user.email “Correo electrónico”

para eliminar lo siguiente

* git config --global user.name “ ”
* git config --global user.email “ ”

revisar que las credenciales se efectuaron en la máquina virtual

1. git config --list o git config --list --global

para salie preciono chif, dos puntos y q al mismo tiempo

-----------------------------------------------------------------------------------------------------------------------

Clase 2

Comandos para configurar git-scm en (forma local) crear credenciales en un proyecto específico dentro de un directorio

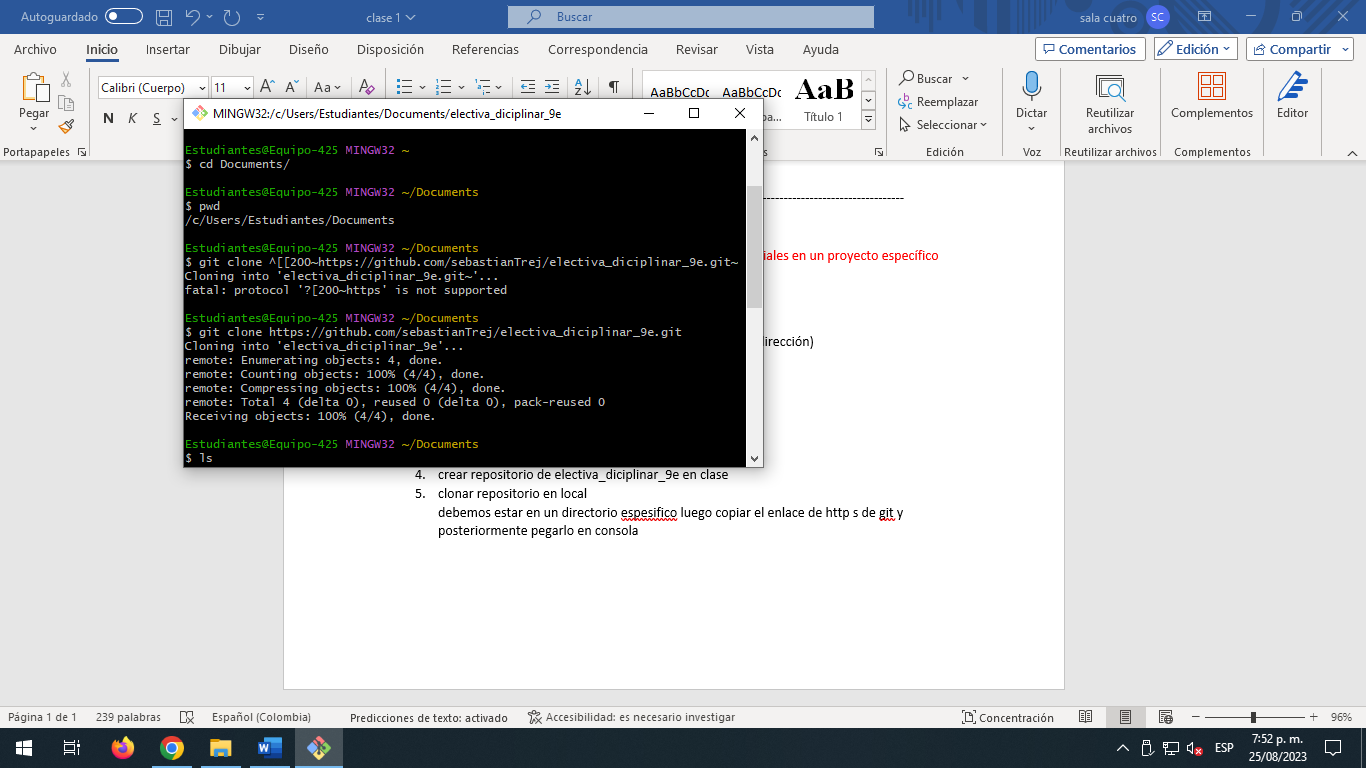
* git config user.name “Nombre de usuario”
* git config user.email “Correo electrónico”
* pwd listar directorios (en que parte me encuentro de la dirección)

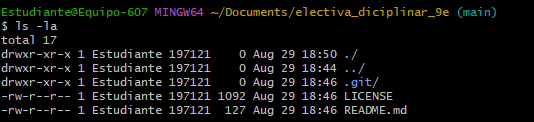
para eliminar lo siguiente

* git config user.name “ ”
* git config user.email “ ”

1. crear cuenta usuario en GitHub
2. crear repositorio de electiva\_diciplinar\_9e en clase
3. clonar repositorio en local

debemos estar en un directorio espesifico luego copiar el enlace de http s de git y posteriormente pegarlo en consola





-------------------------------------------------------------------------------------------------------------

Clase 3

Hay dos formas de crear repositorios con git/gitHub

1. crearlo en la nube 🡪 crearlo desde la aplicación web de GitHub

con el comando (code .) se encarga de abrir el proyecto en visual studio code

* mkdir 🡪 crear directorio o carpeta
* git init 🡪 comando para iniciar una carpeta en tierra a repositorio
* git branch 🡪 comando que lista las ramas que tengo en mi repositorio

cuando la carpeta tiene el (main) quiere decir que es una rama, un repositorio

* git checkaut -b test 🡪 comando para crear una rama y cambiarse a ella (test)
* git branch dev 🡪 crear una rama
* git checkout dev 🡪 cambiar de rama

staging envia los repositorios de tierra a la nube

----------------------------------------------------------------------------------------------------------------------

Clase 4

* git add . 🡪 este comando sirve para agregar a cola todos los cambios presentes al repositorio
* git add --all 🡪 este comando sirve para agregar cambios
* git add index.html 🡪 sirve para versionar un archivo en especifico
* git add app/ 🡪 sirve para subir toda la carpeta

🡪 git commit -m “mi primer commit” 🡪 sirve para enviar un mensaje sobre los cambios que realice en determinado proyecto y entre mas claro mejor porque luego podre saber que cambios realice con este mensaje

* git push origin main 🡪 subir el documento

para poder subir las modificaciones o nuevos archivos entonces debo estar en la ruta del repositorio luego se coloca lo siguiente

git status 🡪 con este puedo ver cuáles son los archivos nuevos que se encontraron o cuales fueron modificados

git add . 🡪 subir el archivo

git commit -m “initial commit” 🡪 hacer un comentario

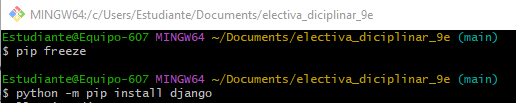
git push origin main 🡪 para subir a mi cuenta de git

------------------------------------------------------------------------------------------------------------------------------

Clase 5

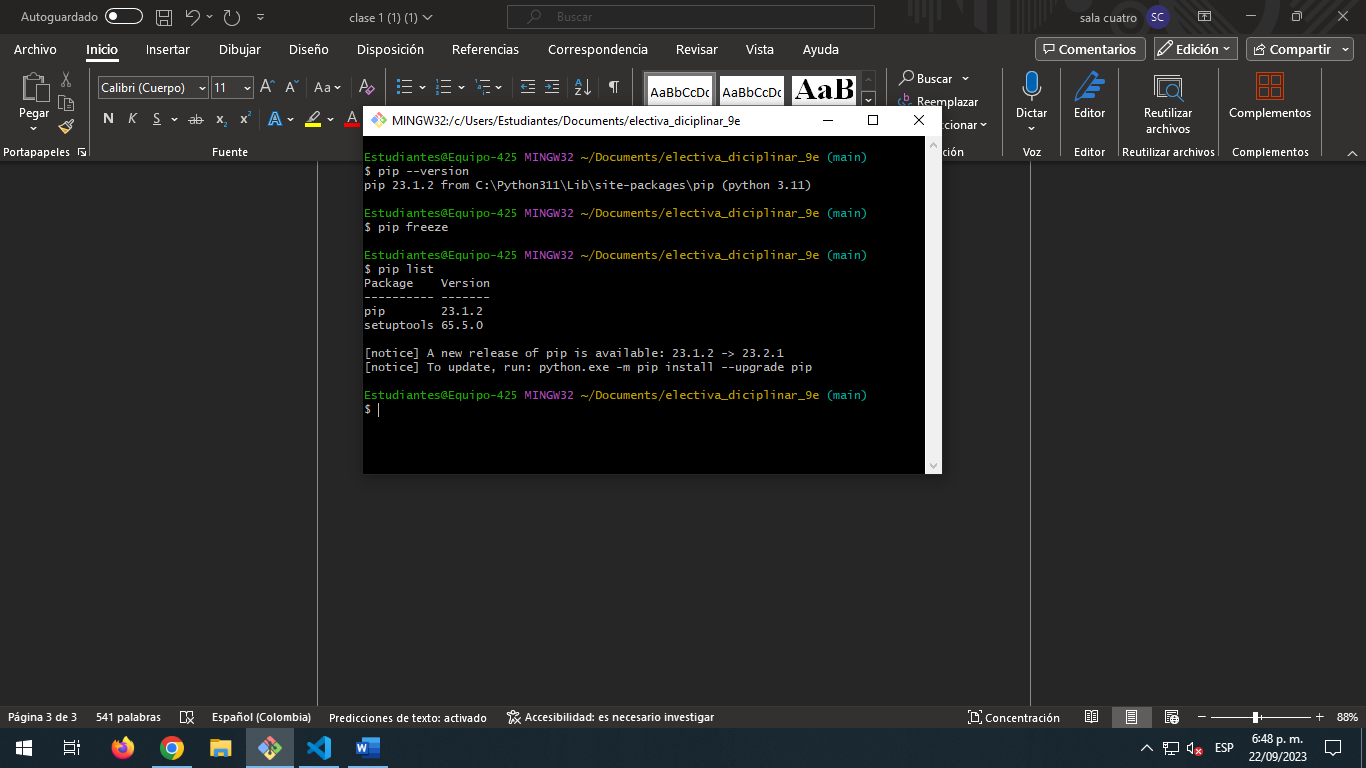
git pull 🡪 esto para descargar los datos actualizados desde el repositorio

instalar django

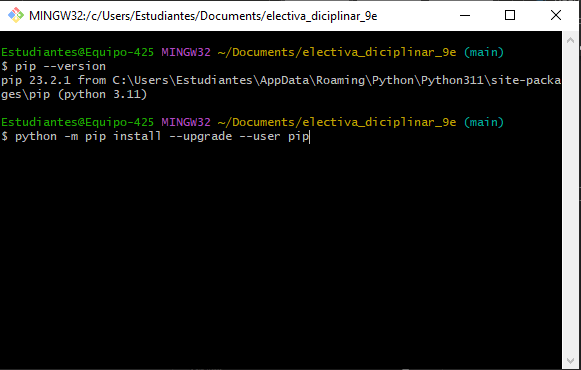


Clase 6

Comandos para revisar las versiones



Comando para actualizar la versión del pip



Instalar django de manera global



Crear un entorno virtual 

Activar modo virtual

Texto

Descripción generada automáticamente con confianza media

Instalar django en entorno virtual



Crear un projecto de django

Texto

Descripción generada automáticamente con confianza baja

Patron de desarrollo de django es MVT (Model-Views-Template)

Motores de BD que soporta django mysql, postgrest, SQlite, mariaDb, Oracle.

Correr el servidor



Clase 6

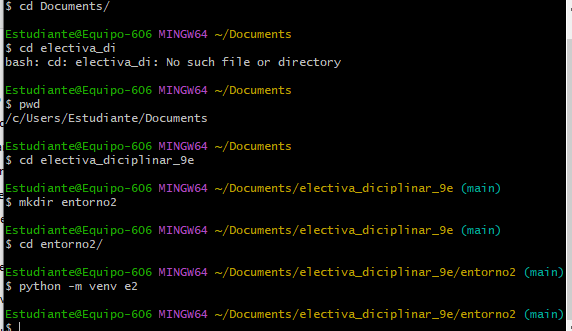
pip freeze > requirements.txt 🡪 es un comando que muestra una lista de todos los paquetes de Python instalados en tu entorno virtual redirige la salida a un archivo llamado "requirements.txt".

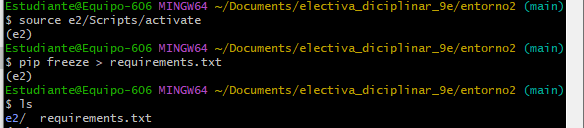
pip install -r requirements.txt 🡪 pip leerá ese archivo y buscará los paquetes y las versiones especificadas en el archivo para instalarlos en el entorno virtual actual.

Antes de ejecutar cualquier de los 2 debo activar el entorno virtual

ejecutar el archivo para cargar las librerias

a continuacion creación de entorno virtual 2

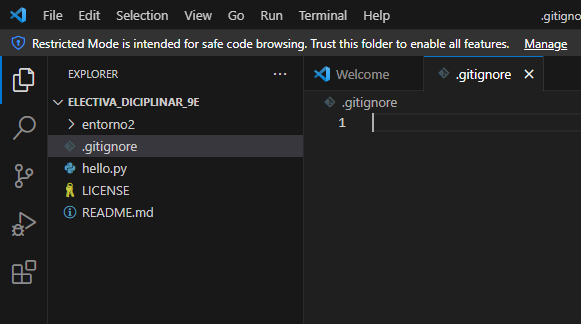








Crear el archivo .gitignore



Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Para actualizar las librerias

Clase 7

Antes de entrar a clase

1. git pull, git fetch
2. Source e1/Scripts/actívate
3. # esto solo si no he instalado un entorno virtual y hay que crearlo 🡪 Python -m venv e1

Comandos

* Pip install -r requirements.txt
* python manage.py runserver 🡪 correr servidor
* python manage.py showmigrations 🡪muestra migraciones de mi proyecto django



Clase 1-3 corte

* python manage.py showmigrations 🡪muestra migraciones de mi proyecto django
* python manage.py makemigrations 🡪crea migraciones de mi proyecto django
* python manage.py migrate 🡪crea la tabla

clase 2

crear super usuario 🡪 Python manage.py createsuperuser desde cmd no git bash

actualizado

Clase final

python manage.py Shell 🡪 ingresar al Shell  
  
from students.models import Users 🡪 crear un orm

Users.objects.all() 🡪 hacer un select desde la consola de orm

new\_user=Users(email="andres@gmail.com", password="123", status=True)

new\_user.save()