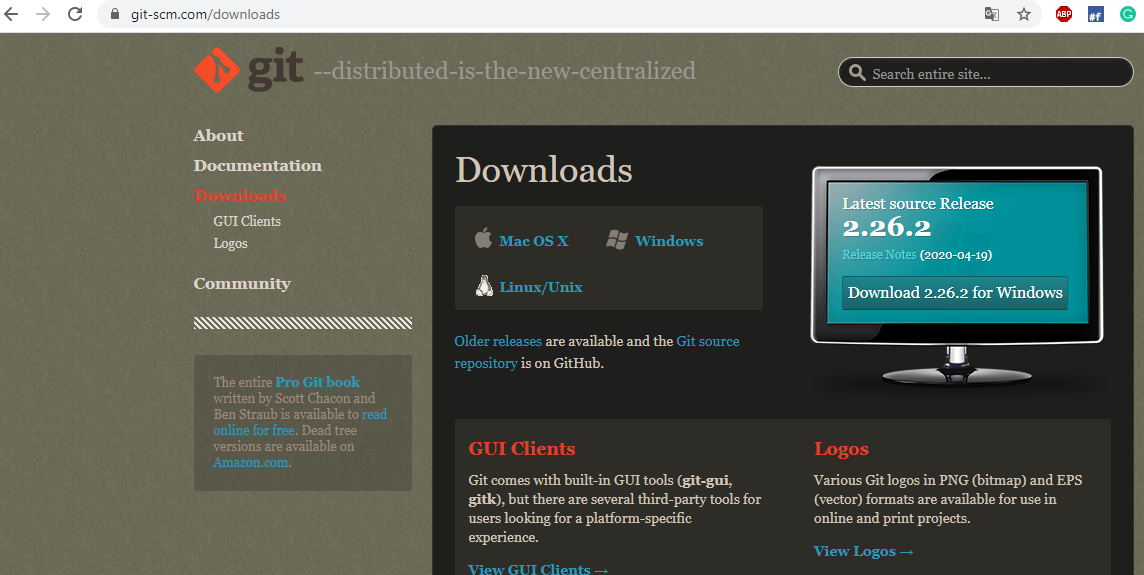
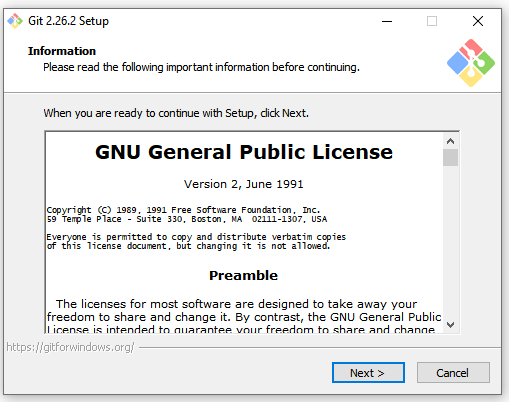
# Instalar GIT

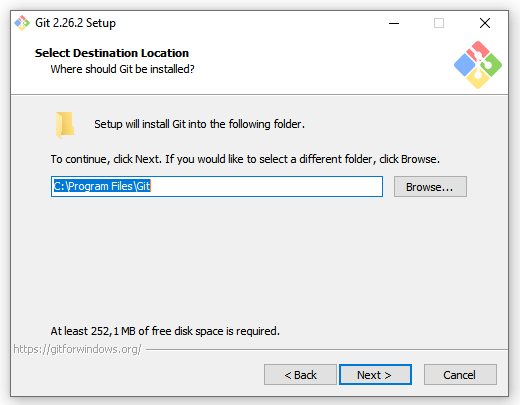
1.Ingresamos al siguiente link “<https://git-scm.com/downloads>”, el cual nos mostrara la siguiente ventana y le damos click a “Windows” para comenzar a descargar.



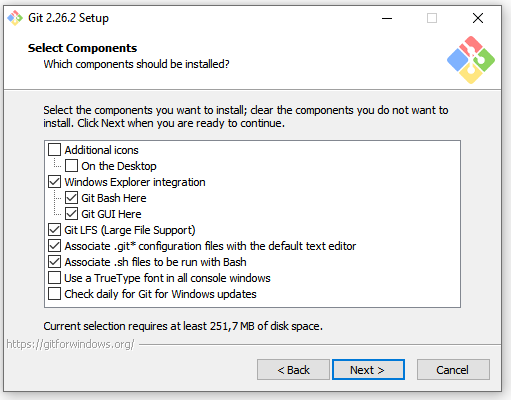
2.Ejecutamos el instalador “Git”.Nos mostrara los términos de instalación, le damos click al botón “Next”



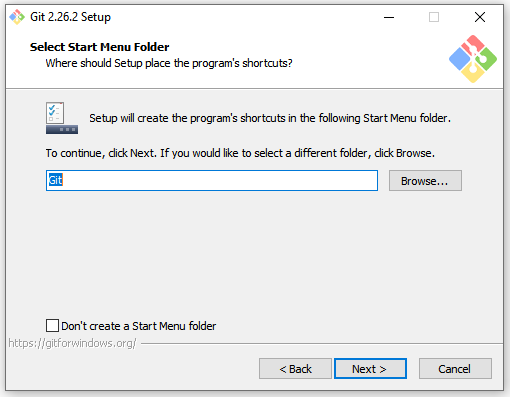
3.Nos muestra la ruta de instalación, en mi caso lo dejo predeterminado y le doy click al botón “Next”.



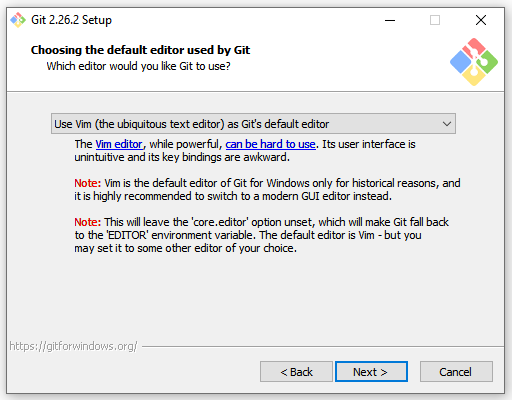
4.Nos muestra la selección de Componentes, en mi caso lo dejo predeterminado y le doy click al botón “Next”.



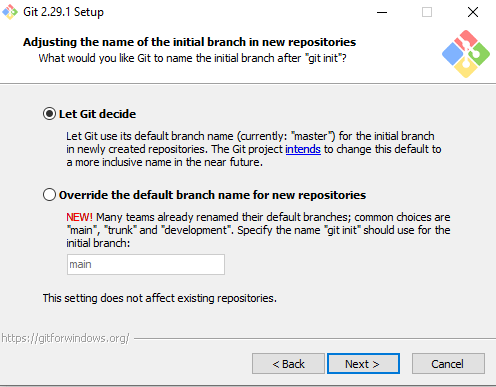
5.Nos muestra a la carpeta del menú de inicio, en mi caso lo dejo predeterminado y le doy click al botón “Next”.



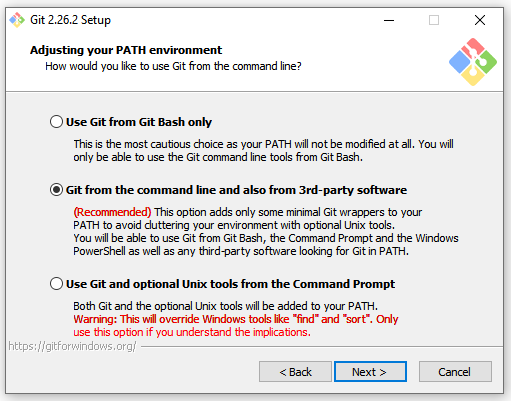
6.Seleccionamos el editor usado por Git que viene predeterminado, dando click al botón “Next”



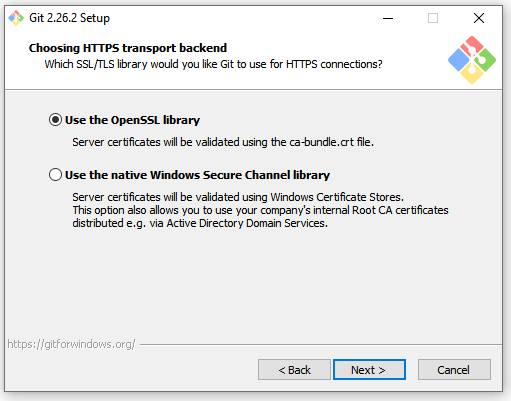
7. Le damos click a “Next”



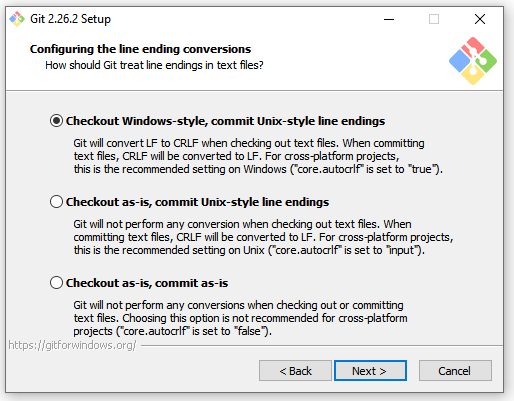
8.Seleccionamos la segunda opción para añadir el Git en el PATH.Y le damos click al botón “Next”.



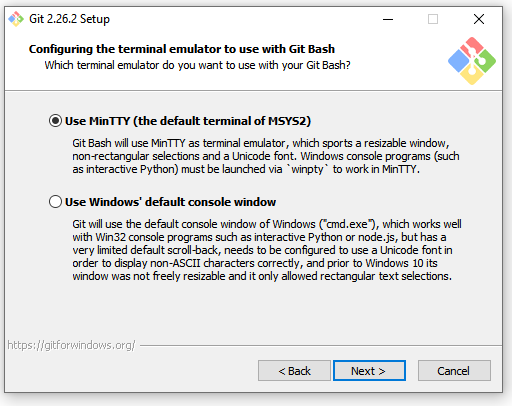
9.Nos muestra las conexiones de HTTPS que usara el Git, en mi caso lo dejo predeterminado y le doy click al botón “Next”.



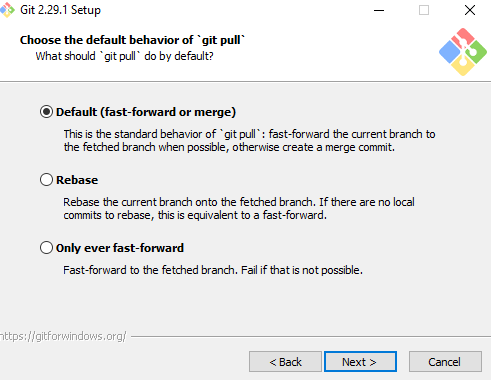
10.Nos muestra la configuración de las conversiones de final de línea, lo dejamos predeterminado. Y le damos click al botón “Next”.



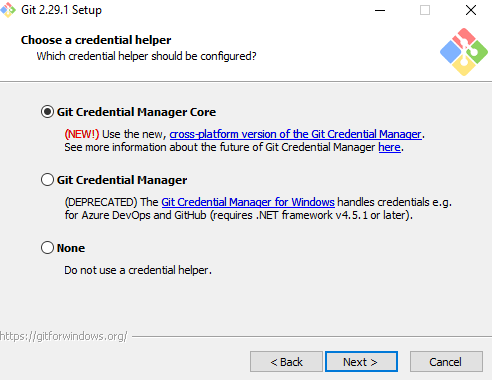
11.Nos muestra la configuración del emulador de terminal para usar con git bash, lo dejamos predeterminado. Y le damos click al botón “Next”.



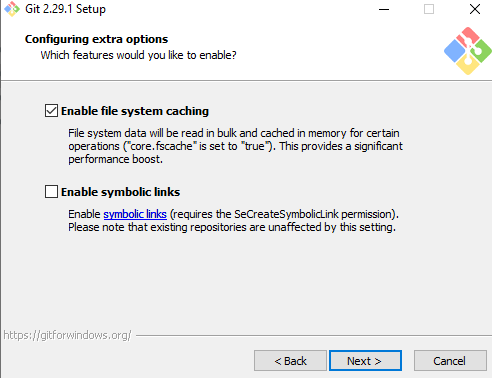
12.Seleccionamos la Opción “Default” y le damos click a “Next”



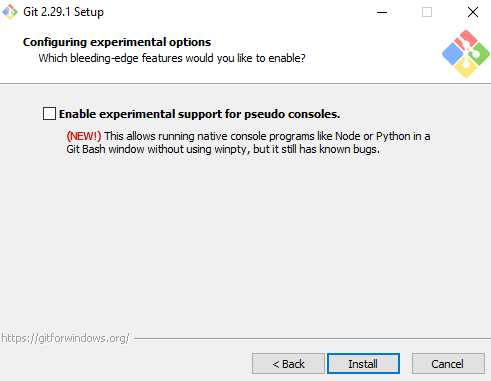
13.Seleccionamos el Git Credential Manager Core y luego le damos click a “Next”



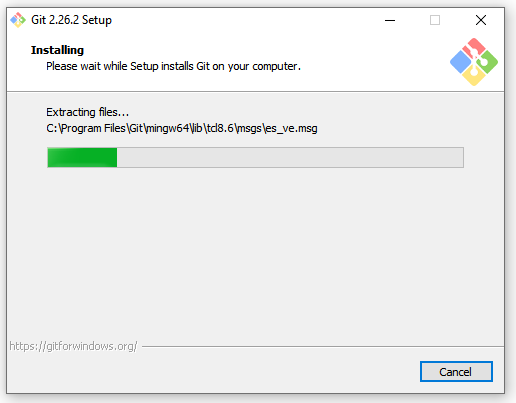
14.Nos muestra la configuración de opciones extras, lo dejamos predeterminado. Y le damos click al botón “Next”.



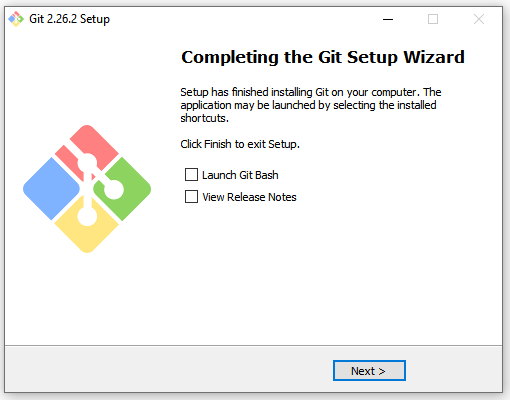
15.Nos aparecerá para probar una configuración experimental y no seleccionamos la opción y le damos a “Install”



16.Nos mostrara en la siguiente ventana el progreso de la instalación.



17.Por último nos mostrara el siguiente cuadro. En mi caso no dejo ninguno seleccionado y ledo al botón “Next”.

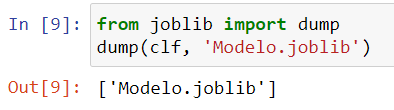


# Crear aplicación Dash Local

1.Una vez que definimos que modelo predictivo queremos desplegar necesitamos guardarlo en un archivo

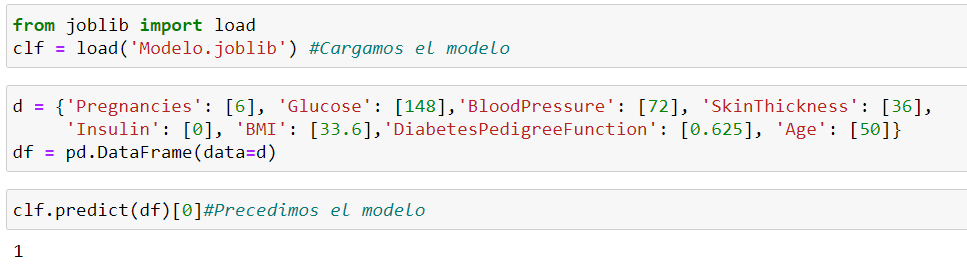


El algoritmo ya entrenado esta guardado en “clf”



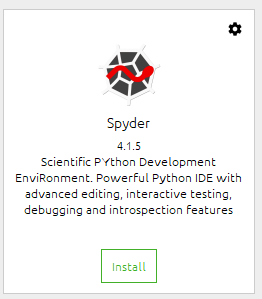
El archivo generado estará en el mismo directorio que creaste el notebook

Ahora un ejemplo de como cargar el modelo y realizar una predicción

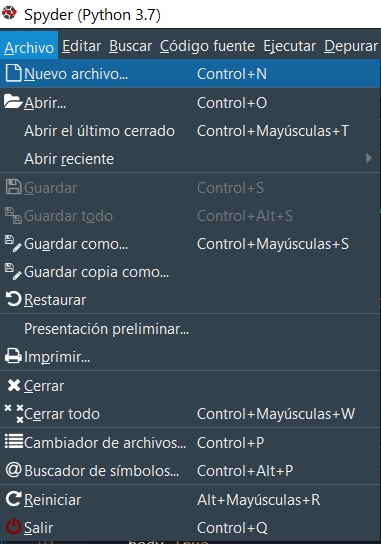


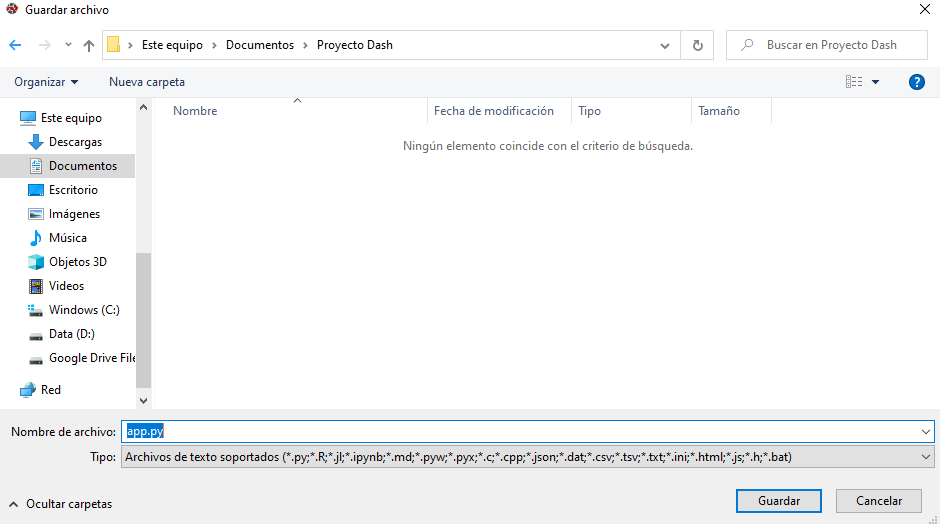
Se muestra que predijo que es de la clase 1

2.Instalamos Spyder desde el Anaconda Navigator.



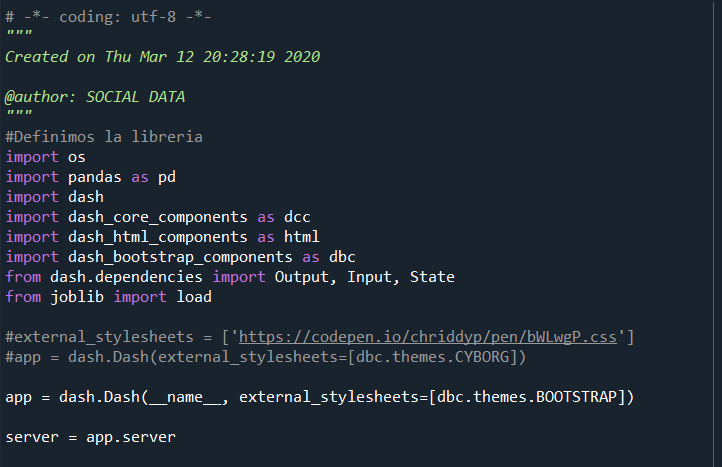
3.Creamos un nuevo archivo “.py”





4.El código lo dividimos en 4 secciones

A) Importación de librerías y definimos el DASH.

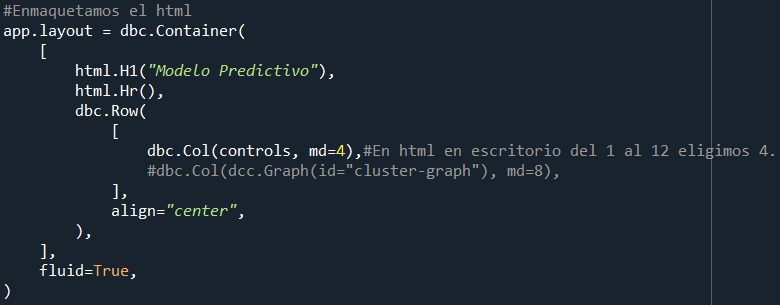


B) Definimos el html del formulario

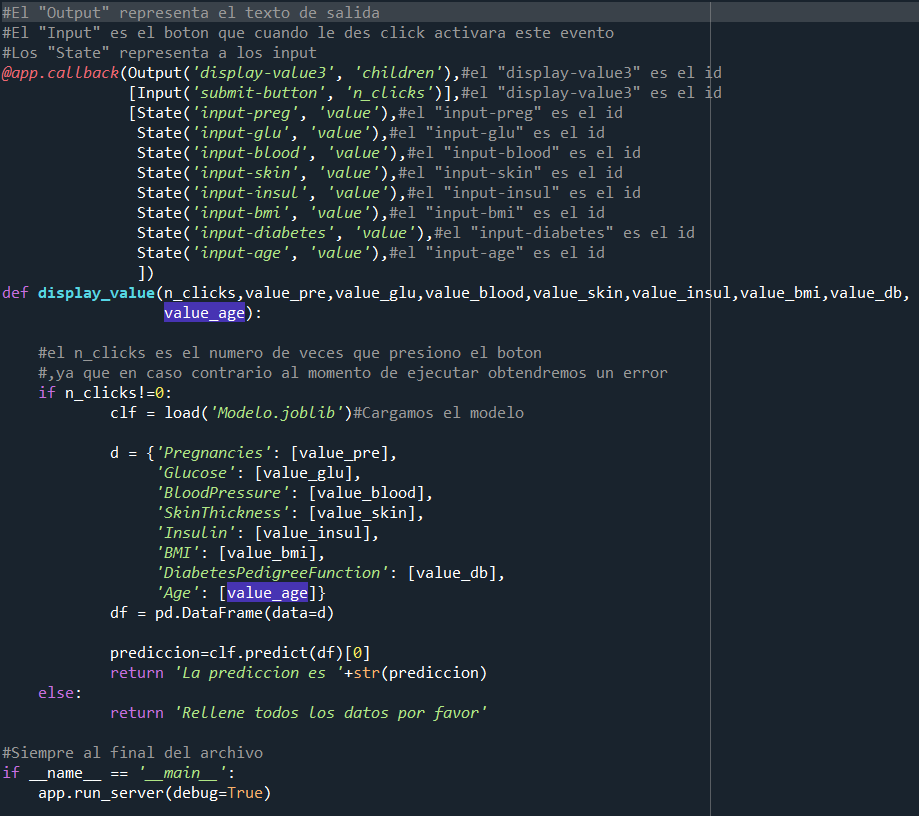




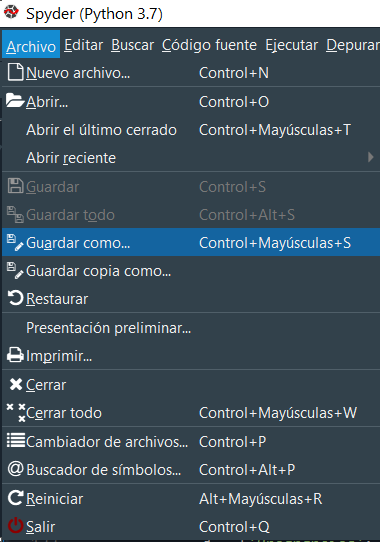
C) Definimos el html en general



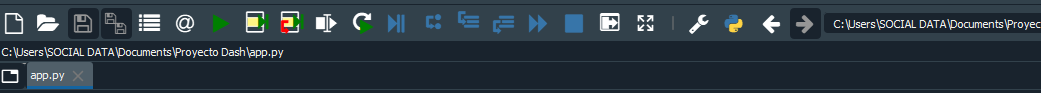
D) Definimos el evento del Botón de Predecir.



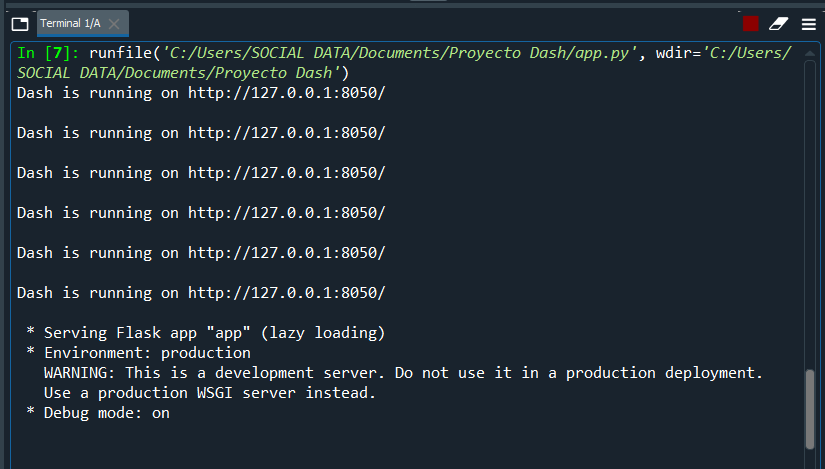
5.Guadamos el nombre del archivo con el nombre “app.py” dando click al botón “Guardar como”



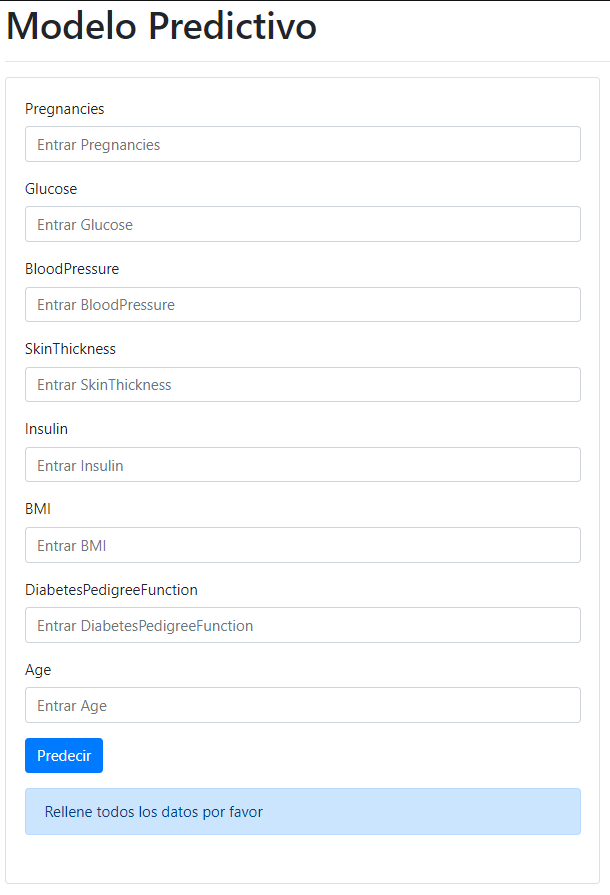
6.Le damos click al botón  “Ejecutar Archivo”



7.Nos debería mostrar en el terminal la url

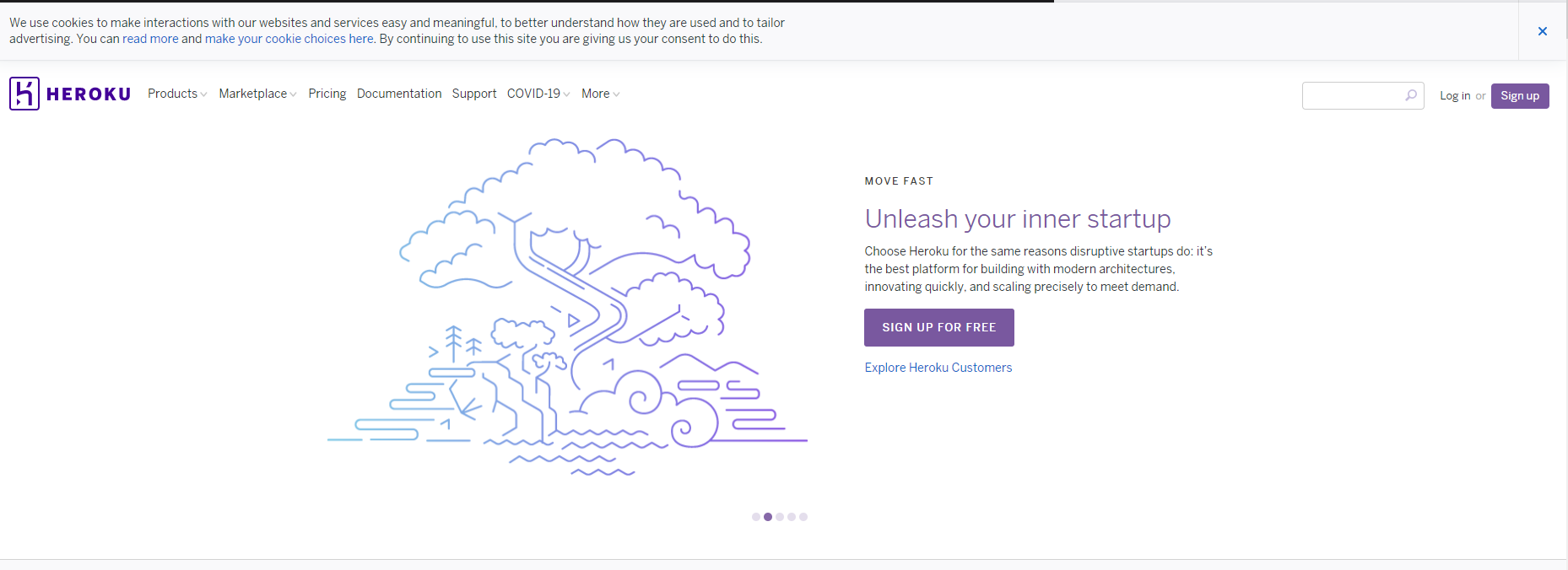


8.Nos muestra el siguiente formulario para rellenar y obtener la prediccion

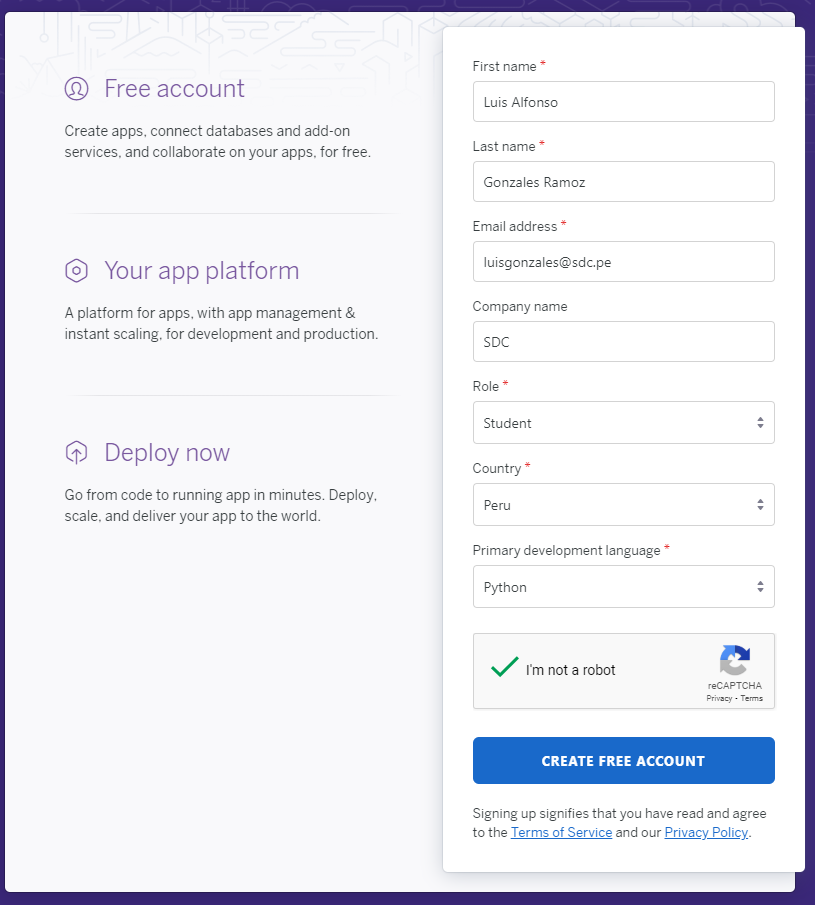


# Registrar cuenta de Heroku

1.Registrar una cuenta en Heroku, mediante el siguiente link <https://www.heroku.com>, dando click al botón “SIGN UP FOR FREE”



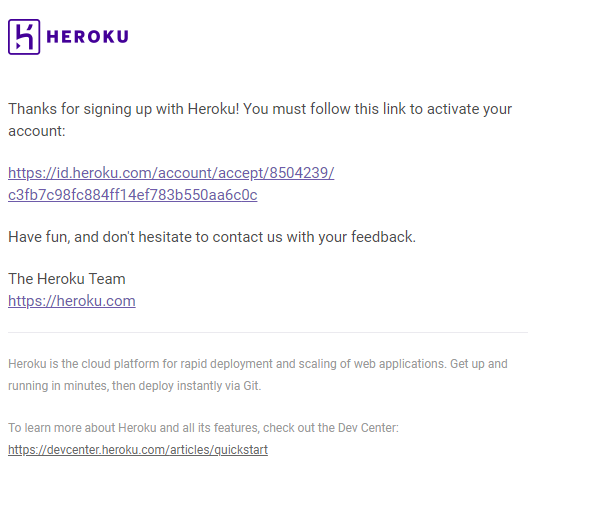
2.Registramos nuestros datos, chequeamos el de “No soy un robot” y le damos a “Create Free Account”



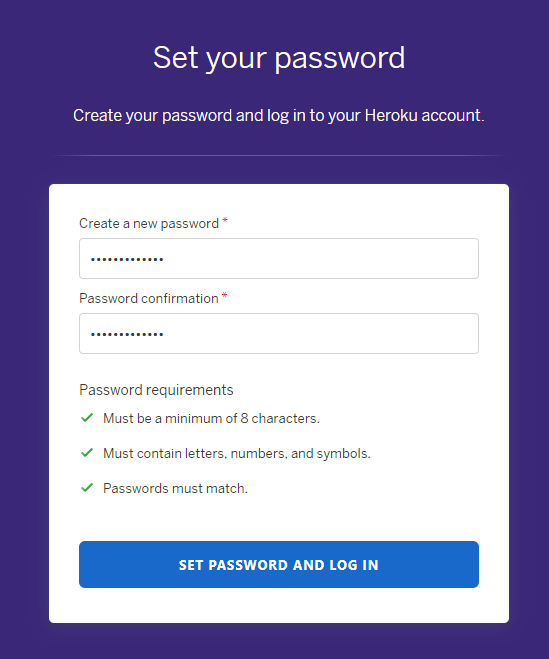
3.Recibira un email para confirmar la cuenta de Heroku.



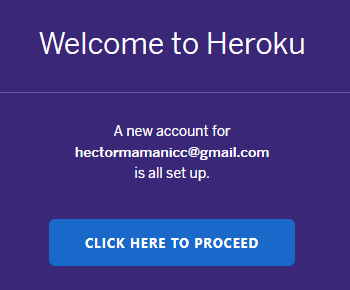
4.Al abrir el mensaje encontrara un url, al dar click nos abrirá una pagina.



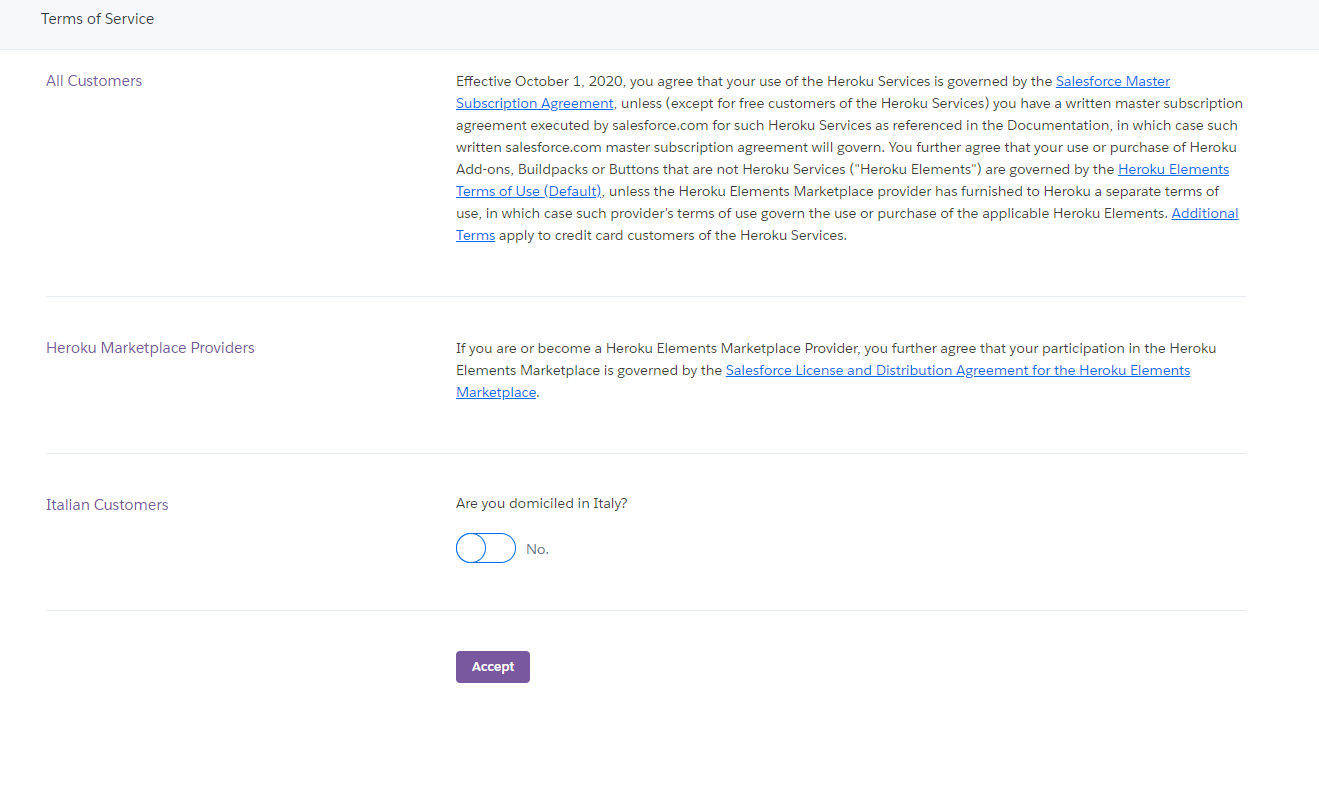
5.Ingresamos una contraseña que tenga mayúsculas, mínimo 8 caracteres y signos,



6.Nos confirma la creación de la cuenta y le damos click al botón “CLICK HERE TO PROCEED”

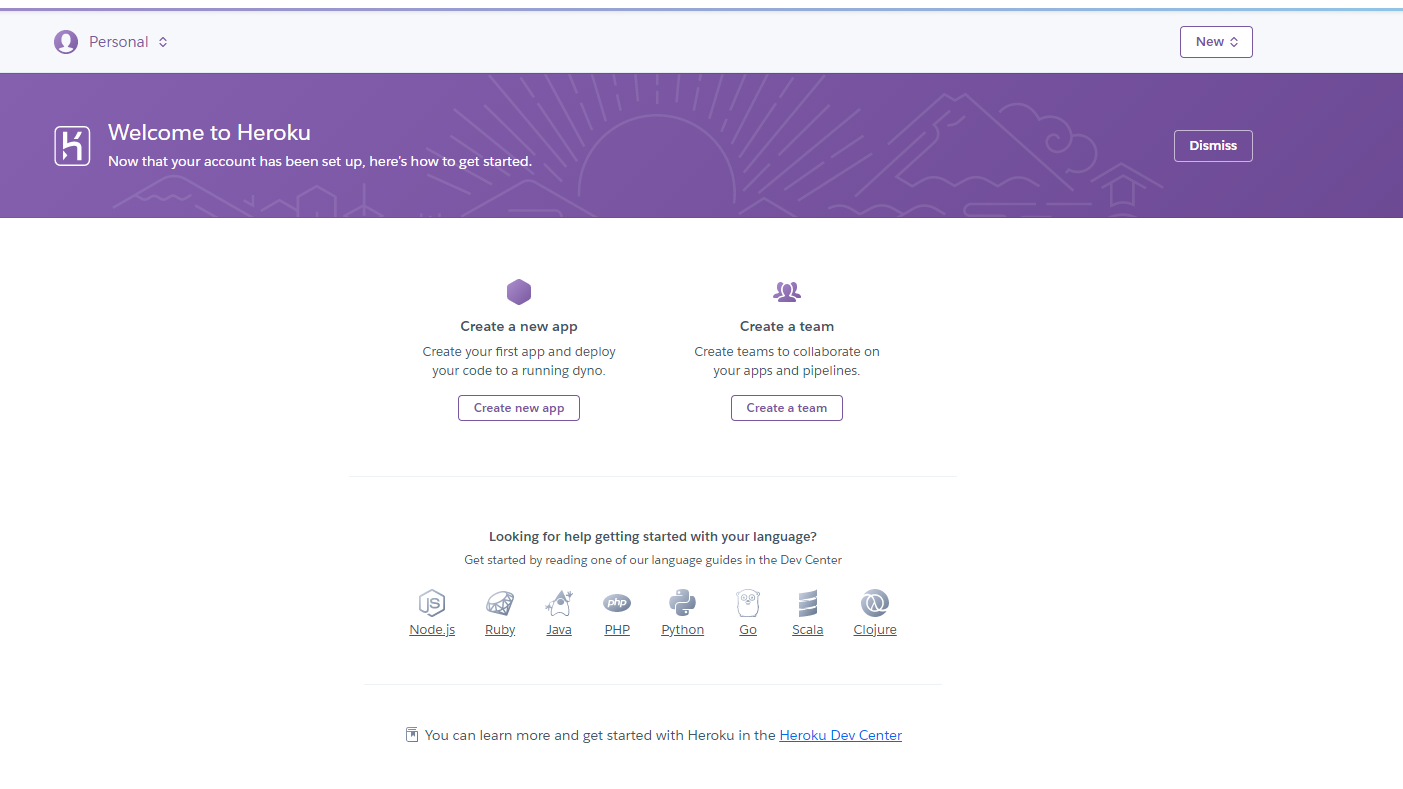


7.Aceptamos los términos de servicio dando click al boton “Accept”

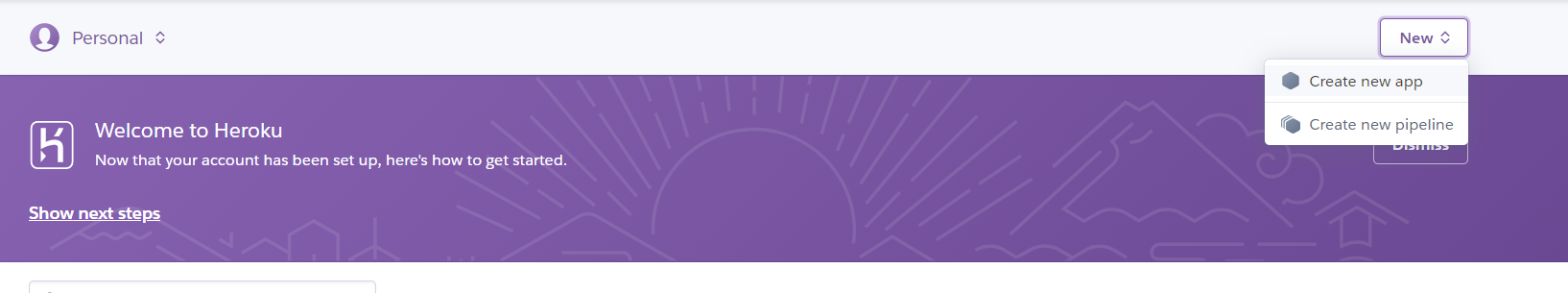


# Crear aplicación en Heroku

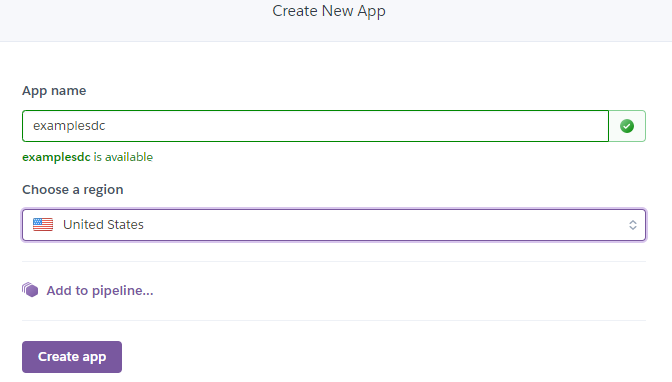
1.En el siguiente link “<https://dashboard.heroku.com/apps>” podemos crear el nuevo proyecto dando click en el boton “New”.



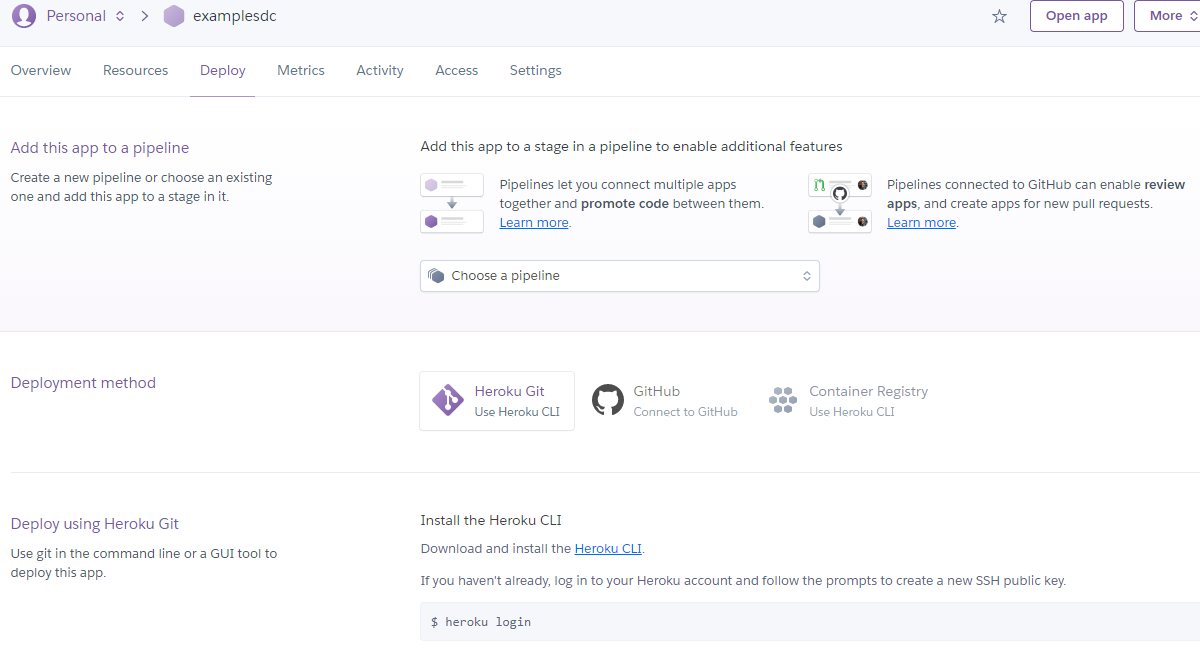
Luego damos click en el boton “Create New App”.



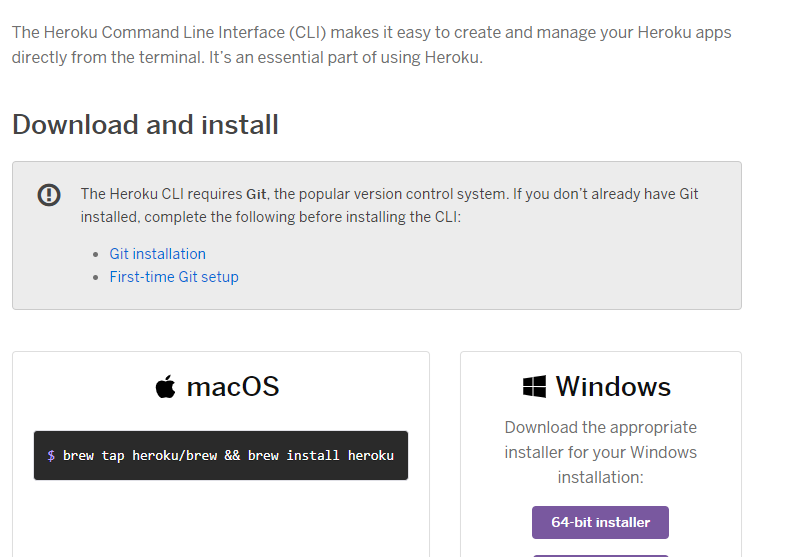
2.Ingresamos el nombre de nuestra aplicación y seleccionamos la región United States, para confirmar le damos click al botón “Create app”.



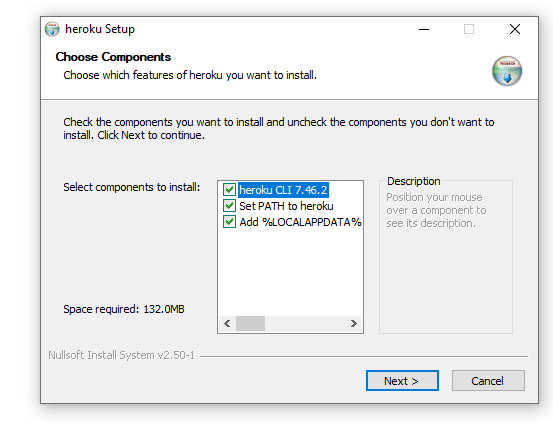
3.En el tablero “***Deploy***” del proyecto, seguimos las instrucciones del “***Deploy using Heroku Git***”,Y le damos click al vinculo.



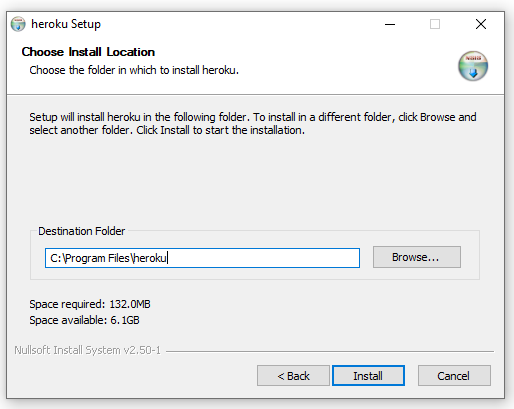
4.Descargamos el Heroku dando click al botón “64-bit installer”



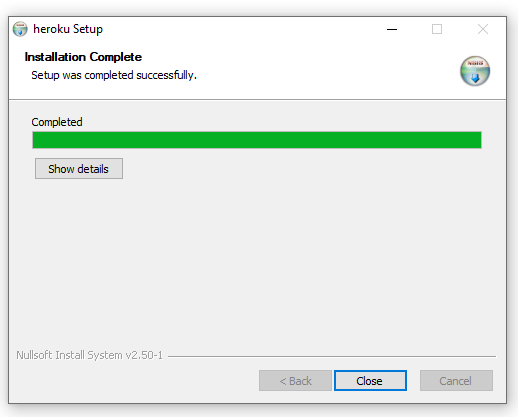
5.Iniciamos el instalador, seleccionamos las 3 opciones le damos a siguiente.



6. Definimos la ruta donde queremos instalar, en mi caso lo pongo predeterminado.



7.Una vez completado la instalación, le damos click al boton “Close”.

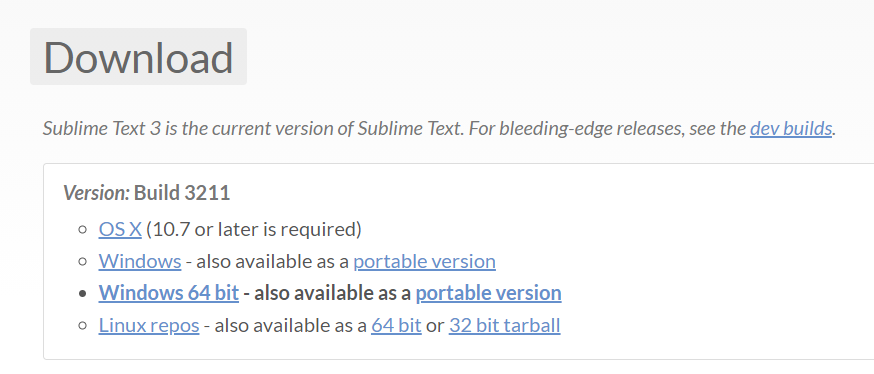


8.Creamos la carpeta del proyecto.

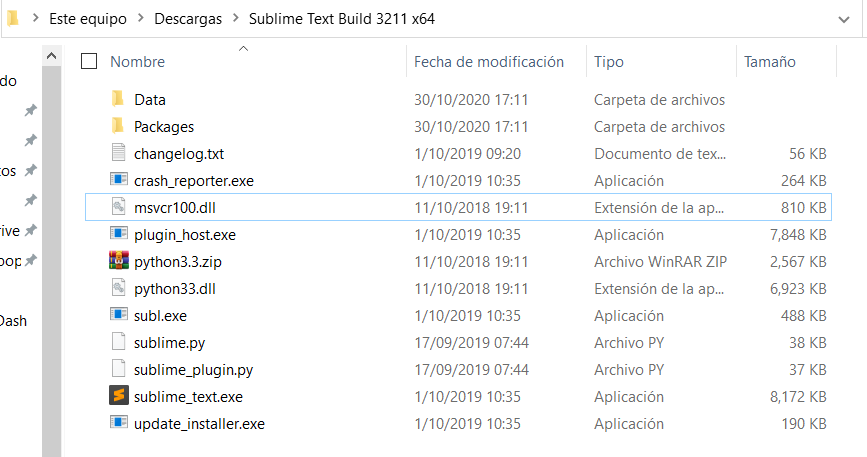


# Generar archivos necesarios para el proyecto

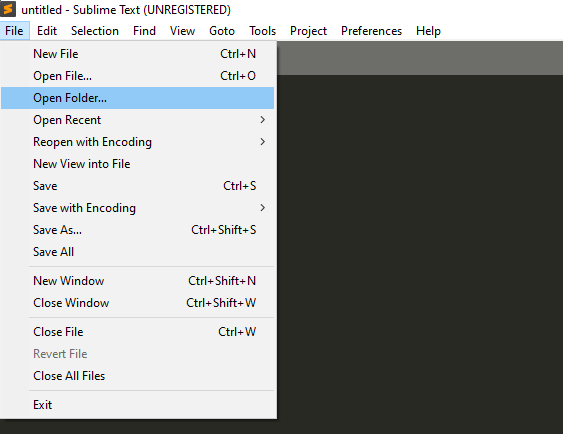
1.Ingresamos al siguiente link “<https://www.sublimetext.com/3>” y descargamos la versión de “Windows 64 bit” la versión portable



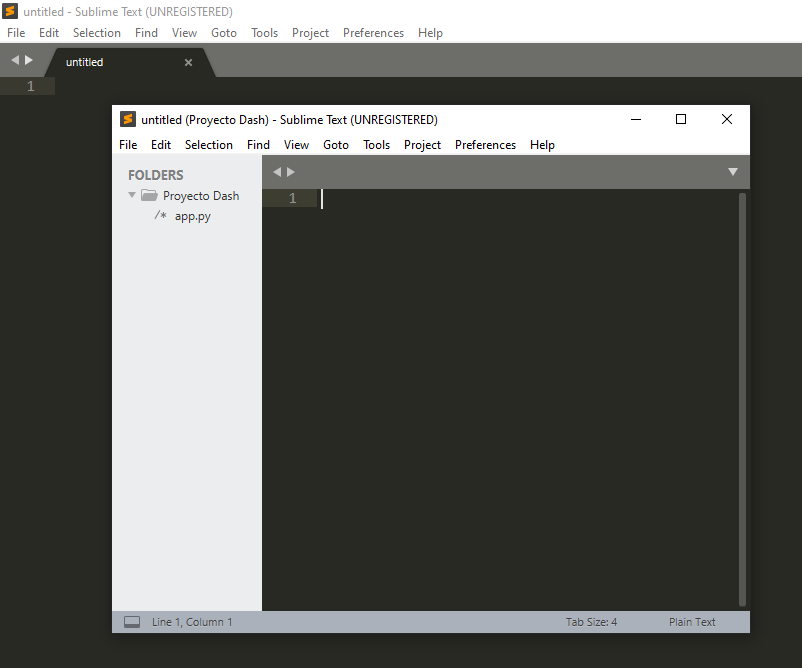
2.Lo descomprimes en una carpeta y le damos click al “sublime\_text.exe”



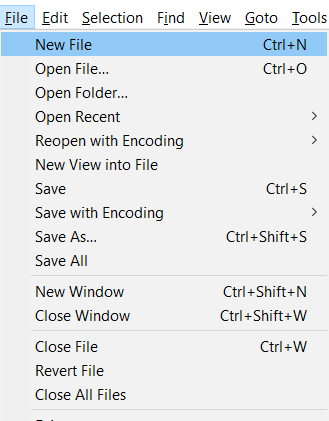
3.Le damos click a File > Open Folder y buscamos la carpeta donde esta “Proyecto Dash”.



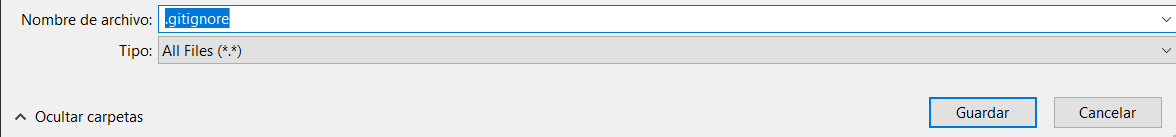
4.Como se muestra solo existe un archivo hasta ahora y nos abrió una nueva ventana



5.Necesitamos crear los archivos “.gitignore, Procfile” en esta misma carpeta



Y una vez escribiendo los dos archivos. Le damos click a “Save As…” con su respectivo nombre



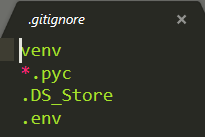
A) .gitignore

venv

\*.pyc

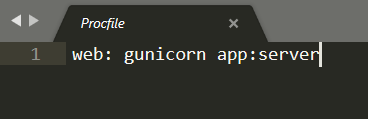
.DS\_Store

.env



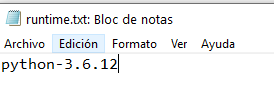
B)Procfile

web: gunicorn app:server



C)runtime.txt

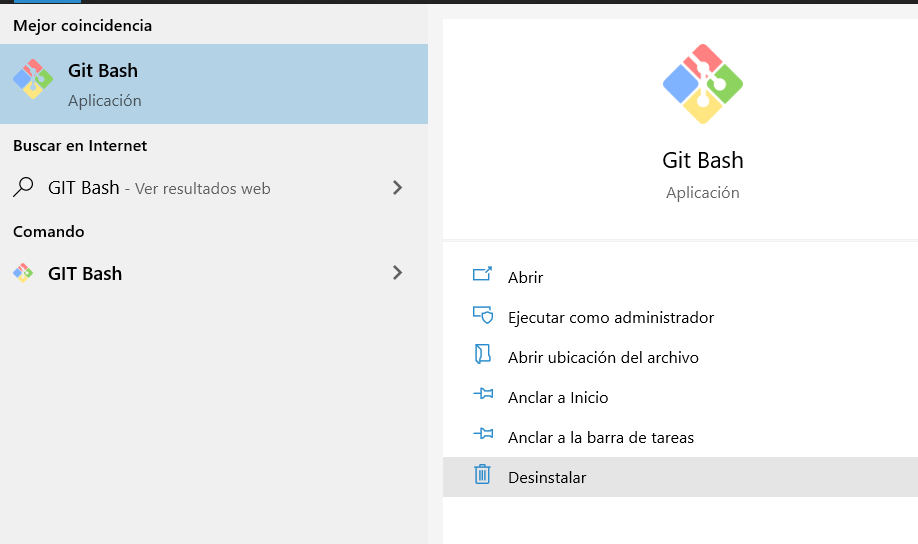
python-3.6.12



Nota: ¿Por qué es necesario especificar la versión? Heroku instala la última versión de Python 3.9. Y hay paquetes que entran en conflicto con aquella versión.

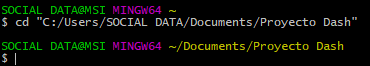
# Desplegar el modelo mediante Git Bash

1.Buscamos en nuestro el Git Bash.



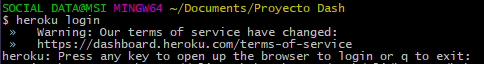
Nota:A partir de acá seguir el código mostrado en las imágenes no copiar y pegar desde el Word al Git ya que da errores de formato o la doble – parece uno solo.

2.Buscamos el Proyecto Dash mediante la línea de código cd “C:/Users/SOCIAL DATA/Documents/Proyecto Dash”

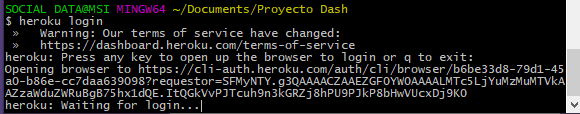


***Nota:En algunos casos si no funciona con doble comilla intente con comilla simple.Pasarlo a bloc de notas por que el Word tiene otro tipo de fuente que no soporta el Git***

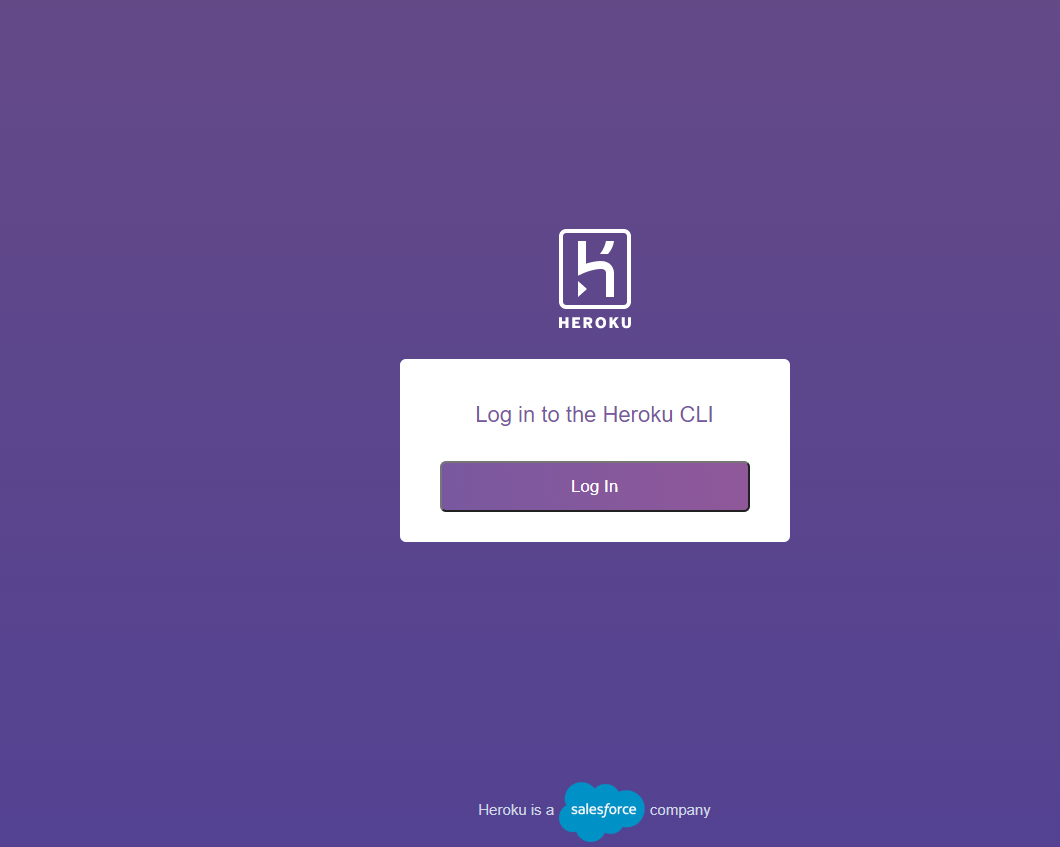
3.Ingresamos a nuestro heroku mediante el “heroku login” y presionamos ENTER



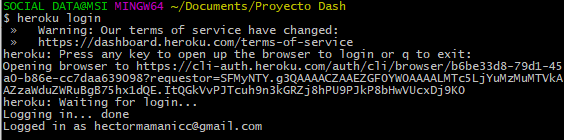
4.Nos pedirá que esperemos



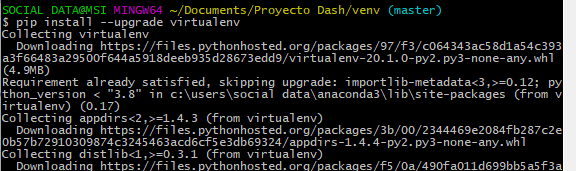
5.Nos abrirá una ventana en nuestro navegador le damos click al botón “Log In”, para confirmar.



En nuestro Git Bash, ya deberia aparecernos que ya iniciamos sesión si no vuelve a aparecer el signo de abajo del dólar para continuar presione “Control+C”



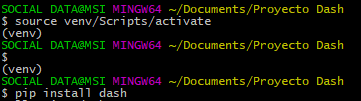
6.Antes crear nuestro entorno para este proyecto. Actualizamos con el comando “pip install –upgrade virtualenv”



7.Creamos el entorno mediante el siguiente comando “python -m venv venv”



8.Activamos el entorno “venv” mediante el comando “source venv/Scripts/activate”



Nota: El (venv) arriba de SOCIAL DATA@MSI nos muestra que efectivamente se activó el entorno.

9.Tambien instalamos todos las librerías que requiere el proyecto como pandas,dash,etc. Se recomienda que instale en el orden que se presenta para que no haiga conflictos

pip install numpy==1.19.3

pip install scipy==1.5.3

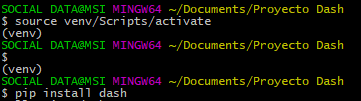
pip install pandas==1.1.4

pip install dash

pip install dash-bootstrap-components

pip install -U scikit-learn

pip install gunicorn



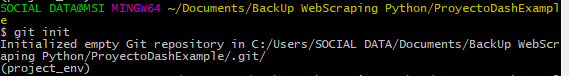
Nota: En algunos casos es necesario especificar las versiones de los diferentes paquetes, ya que Heroku no tiene todas las versiones.

10.Es necesario crear un archivo donde esta colocado las versiones de las librerías que hemos instalado para este proyecto mediante el siguiente comando.

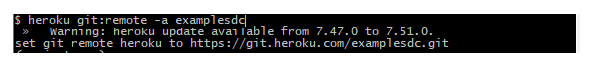
pip freeze > requirements.txt



11. Inicializamos el git



12.Enlazamos el git con heroku



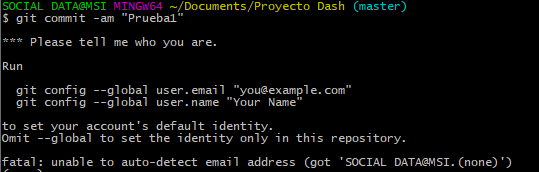
13.Ahora añadimos los archivos mediante el:

git add .



14.Ahora definimos el nombre de nuestro primer commit mediante el:

git commit -am “Prueba1”



Pero ocurre que necesitamos definir el Gmail y el nombre mediante los comandos siguientes

git config --global user.email “luisgonzales@sdc.pe”

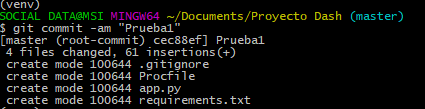


Git config –global user.name “Luis”

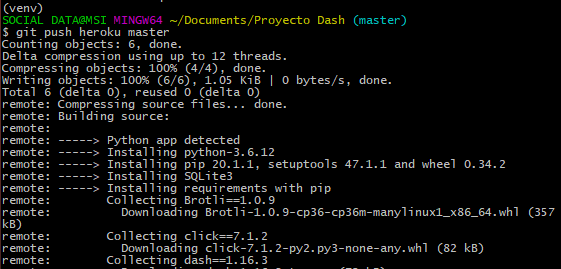


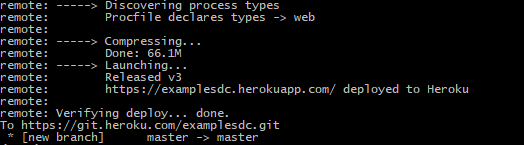
Volvemos a intentar y ya no nos aparece el error de antes





15. Realizamos el despliegue del modelo mediante el comando “git push heroku master”



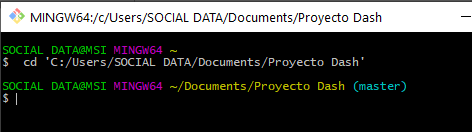


Y si todo sale correcto nos mostrara el url del proyecto desplegado en mi caso seria en <https://examplesdc.herokuapp.com/>

## Realizar actualizaciones

Nota: En algunos casos es necesario volver a iniciar sesion el heroku login

1.Ingresamos al proyecto mediante el comando “cd”



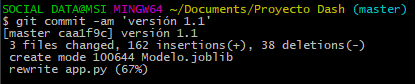
2.Selecciona los archivos del Proyecto mediante el

“git add .”



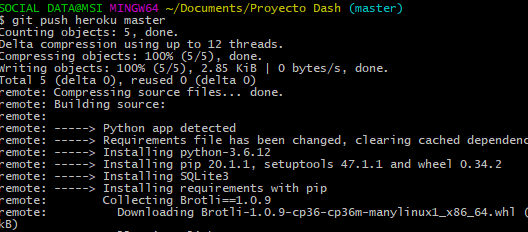
3.Define el nombre del título de nuestro cambio mediante el

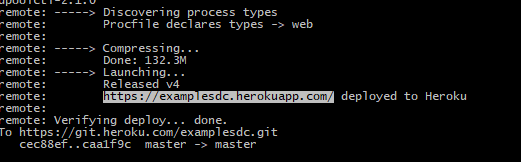
git commit -am “versión 1.1”



4.Subimos los cambios mediante el

“git push heroku master”





5.Como Resultado obtendríamos nuestro modelo desplegado del url

