
	PROTOTIPO DE SISTEMA PARA EL ANÁLISIS DE OPINIONES BASADO EN TWEETS. CASO DE ESTUDIO: METRO DE QUITO	VERSIÓN: 1.0
		FECHA: 03 de abril del 2023


MANUAL DE INSTALACIÓN

**ANÁLISIS DE OPINIONES BASADO EN TWEETS.
CASO DE ESTUDIO: METRO DE QUITO**

	MANUAL DE INSTALACIÓN METRO DE QUITO - EPN	VERSIÓN: 1.0
		FECHA: 03 de abril del 2023

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS	2
OBJETIVO	3
REQUERIMIENTOS	3
GUÍA DE INSTALACIÓN	3
1. Instalación Python 3.....	3
2. Instalación Pip	3
3. Instalación Virtualenv	3
4. Instalación Flask	4
5. Instalación Dash	4
6. Instalación Plotly	4
7. Instalación Pandas	4
8. Instalación Wordcloud	4
9. Instalación spacy	5
10. Instalación nltk	5
11. Instalación servidor	5
11.1. Archivo profile	5
11.2. Archivo requirements	5
11.3. Instalación de Heroku en Windows.....	5
11.4. Login Heroku	6
11.5. Cargar archivos hacia Heroku	6
11.6. Deploy en Heroku.....	9

	MANUAL DE INSTALACIÓN METRO DE QUITO - EPN	VERSIÓN: 1.0
		FECHA: 03 de abril del 2023

ÍNDICE DE FIGURAS

Figura 1: Archivo procfile para ejecución del prototipo de sistema para el análisis de opiniones basado en tweets.....	5
Figura 2: Archivo requirements para la instalación de librerías.	5
Figura 3: Instalación de Heroku en Windows.....	6
Figura 4: Comando login hacia Heroku	6
Figura 5: Interfaz de login Heroku.....	6
Figura 6: Mensaje de login exitoso.....	6
Figura 7: Directorio del prototipo de sistema para a el análisis de opiniones basado en tweets	6
Figura 8: Visualización de archivos del directorio.	7
Figura 9: Crear e inicializar repositorio	7
Figura 10: Visualizar los cambios en los archivos	8
Figura 11: Agregar comentario de confirmación de la versión cargada.....	9
Figura 12: Crear repositorio dentro de Heroku	9
Figura 13: Cargar y ejecutar los archivos locales en el ambiente de Heroku.	10

OBJETIVO

Documentar el correcto proceso de instalación del prototipo de sistema para el análisis de opiniones basado en tweets. caso de estudio: metro de quito

REQUERIMIENTOS

- Python 3
- Pip
- virtualenv
- Flask
- Dash
- dash_core_components
- dash_html_components
- dash-bootstrap-components
- Plotly
- Pandas
- wordcloud
- spacy
- gunicorn

GUÍA DE INSTALACIÓN

1. Instalación Python 3

Se está utilizando el sistema operativo Linux con distribución Ubuntu 16.0. puede instalarse bajo los siguientes comandos.

```
$ sudo apt-get update  
$ sudo apt-get install python3.10
```

2. Instalación Pip

Para instalar pip en Linux, ejecute el siguiente comando

```
$ apt install python3-pip
```

Para instalarlo en distribuciones como CentOS o redhat (RHEL) hay que habilitar el repositorio EPEL para poder instalarlo

```
$ yum install epel-release  
$ yum install python-pip
```

3. Instalación Virtualenv

Para instalar un ambiente virtual usar el siguiente comando.

```
$ apt-get install python-virtualenv
```

Para activar el ambiente virtual utilizar el siguiente comando.

```
$ venv\scripts\activate
```

4. Instalación Flask

Para instalar los paquetes se recomienda utilizar pip para obtener la última versión.

```
$ pip install flask
```

Para visualizar los paquetes que fueron instalados de manera adicional, usar el siguiente comando.

```
$ pip freeze
```

Como salida se obtiene lo siguiente

```
Flask ==2.2.2  
Jinja2==3.1.2  
MarkupSafe==2.1.2  
Werkzeug==2.2.2
```

5. Instalación Dash

Para instalarlo hay que ejecutar el siguiente comando.

```
$ pip dash
```

Dash permite agregar componentes como html para el uso de etiquetas y componentes de Bootstrap, para instalarlos ejecutar el siguiente comando.

```
$ dash-bootstrap-component
```

6. Instalación Plotly

Para ser instalado se ejecuta el siguiente comando.

```
$ pip install plotly
```

7. Instalación Pandas

Para ser instalado se ejecuta el siguiente comando.

```
$ pip install pandas
```

8. Instalación Wordcloud

Para ser instalado se ejecuta el siguiente comando.

```
$ pip install wordcloud
```

9. Instalación spacy

Para ser instalado se ejecuta el siguiente comando.

```
$ pip install spacy
```

10. Instalación nltk

Para ser instalado se ejecuta el siguiente comando.

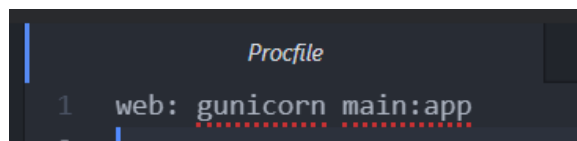
```
$ pip install nltk
```

11. Instalación servidor

El prototipo de sistema para el análisis de opiniones basado en tweets será alojado en el hosting de Heroku, haciendo uso del servidor web Flask.

11.1. Archivo procfile

Para realizar el deploy del prototipo de sistema para el análisis de opiniones basado en tweets se requiere un archivo llamado "Procfile". Procfile hace la llamada al archivo de inicio de la app (main.py) y la instancia de la app (app=Flask)

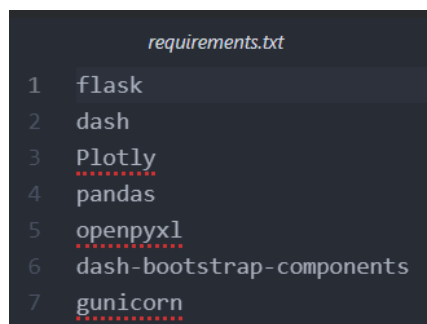


```
Procfile
1 web: gunicorn main:app
```

Figura 1: Archivo procfile para ejecución del prototipo de sistema para el análisis de opiniones basado en tweets.

11.2. Archivo requirements

El archivo permite instalar todas las librerías que utiliza el del prototipo de sistema para el análisis de opiniones basado en tweets de manera recursiva



```
requirements.txt
1 flask
2 dash
3 Plotly
4 pandas
5 openpyxl
6 dash-bootstrap-components
7 gunicorn
```

Figura 2: Archivo requirements para la instalación de librerías.

11.3. Instalación de Heroku en Windows

Ingresa a la consola de Windows o CMD y ejecutar el siguiente comando.

```
> npm uninstall -g Heroku-cli
```

```
C:\Users\Jorge>npm uninstall -g heroku-cli  
up to date in 103ms
```

Figura 3: Instalación de Heroku en Windows

11.4. Login Heroku

Ejecutar el siguiente comando en la consola de Windows.

```
> heroku login
```

```
C:\Users\Jorge>heroku login
```

Figura 4: Comando login hacia Heroku

Automáticamente el comando abrirá su navegador predeterminado hacia el hosting de Heroku. Ingresar las credenciales y el código de seguridad.

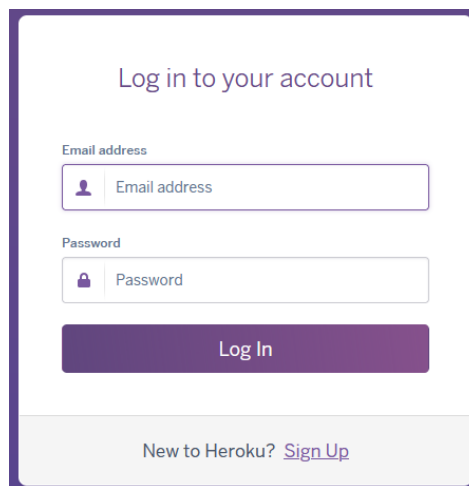
The image shows the Heroku login web interface. At the top, it says "Log in to your account". Below this, there are two input fields: "Email address" with a person icon and "Password" with a lock icon. A purple "Log In" button is positioned below the password field. At the bottom, there is a link that says "New to Heroku? Sign Up".

Figura 5: Interfaz de login Heroku

Luego retorne al terminal de Windows para visualizar que el login ha sido exitoso.

```
C:\Users\Jorge>heroku login  
» Warning: Our terms of service have changed: https://dashboard.heroku.com/terms-of-service  
heroku: Press any key to open up the browser to login or q to exit:  
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/4c33adff-7980-45a2-ae9c-5cb7d30924f4?requestor=SFMyN  
Logging in... done  
Logged in as djjorgemix46@gmail.com
```

Figura 6: Mensaje de login exitoso.

11.5. Cargar archivos hacia Heroku

Ejecutar el siguiente comando para dirigirse a la carpeta que contiene los archivos del proyecto.

```
> cd C:\Users\Jorge\Documents\tesis
```

```
C:\Users\Jorge>cd C:\Users\Jorge\Documents\tesis
```

Figura 7: Directorio del prototipo de sistema para a el análisis de opiniones basado en tweets

Visualizar los archivos del directorio.

```
> dir
```

```
C:\Users\Jorge\Documents\tesis>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: D852-51B9

Directorio de C:\Users\Jorge\Documents\tesis

30/03/2023  21:58  <DIR>          .
30/03/2023  21:58  <DIR>          ..
30/03/2023  20:05  <DIR>          dash_application
01/02/2023  22:46                3.103 main.py
30/03/2023  21:59                24 Procfile
30/03/2023  20:04                74 requirements.txt
01/02/2023  23:14  <DIR>          static
01/02/2023  23:10  <DIR>          templates
01/02/2023  23:14  <DIR>          venv
                3 archivos          3.201 bytes
                6 dirs  388.229.046.272 bytes libres
```

Figura 8: Visualización de archivos del directorio.

Crear un nuevo repositorio con el siguiente comando.

```
> git init
```

```
C:\Users\Jorge\Documents\tesis>git init
Initialized empty Git repository in C:/Users/Jorge/Documents/tesis/.git/
```

Figura 9: Crear e inicializar repositorio

Añadir los archivos de manera recursiva con el siguiente comando.

```
> git add .
```

Visualizar el estado de los archivos y del entorno de trabajo con el siguiente comando.

```
> git status
```



```
C:\Users\Jorge\Documents\tesis>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Procfile
    new file:   dash_application/candidatura_barrera.py
    new file:   dash_application/candidatura_guarderas.py
    new file:   dash_application/candidatura_rodas.py
    new file:   dash_application/candidatura_yunda.py
    new file:   main.py
    new file:   requirements.txt
    new file:   static/archivos/OneDrive_1_20-3-2023.zip
    new file:   static/archivos/candidatura_Barrera.xlsx
    new file:   static/archivos/candidatura_Guarderas.xlsx
    new file:   static/archivos/candidatura_Rodas.xlsx
    new file:   static/archivos/candidatura_Yunda.xlsx
    new file:   static/images/Barrera.png
    new file:   static/images/Guarderas.png
    new file:   static/images/Rodas.png
    new file:   static/images/Yunda.png
    new file:   static/images/fondo.png
    new file:   static/images/metro_logo.png
    new file:   static/images/sad-face.png
    new file:   static/images/smile.png
    new file:   static/images/twitter.png
    new file:   static/images/word_cloud_c1.png
    new file:   static/images/word_cloud_c2.png
    new file:   static/images/word_cloud_c3.png
    new file:   static/images/word_cloud_c4.png
    new file:   templates/header.html
    new file:   templates/home.html
```

Figura 10: Visualizar los cambios en los archivos

Agregar un comentario para confirmar la versión que se está subiendo con el siguiente comando.

```
> git commit -m "comentario"
```

```

C:\Users\Jorge\Documents\tesis>git commit -m "first commit -m"
[master (root-commit) 5635e43] first commit -m
27 files changed, 1631 insertions(+)
create mode 100644 Procfile
create mode 100644 dash_application/candidatura_barrera.py
create mode 100644 dash_application/candidatura_guarderas.py
create mode 100644 dash_application/candidatura_rodas.py
create mode 100644 dash_application/candidatura_yunda.py
create mode 100644 main.py
create mode 100644 requirements.txt
create mode 100644 static/archivos/OneDrive_1_20-3-2023.zip
create mode 100644 static/archivos/candidatura_Barrera.xlsx
create mode 100644 static/archivos/candidatura_Guarderas.xlsx
create mode 100644 static/archivos/candidatura_Rodas.xlsx
create mode 100644 static/archivos/candidatura_Yunda.xlsx
create mode 100644 static/images/Barrera.png
create mode 100644 static/images/Guarderas.png
create mode 100644 static/images/Rodas.png
create mode 100644 static/images/Yunda.png
create mode 100644 static/images/fondo.png
create mode 100644 static/images/metro_logo.png
create mode 100644 static/images/sad-face.png
create mode 100644 static/images/smile.png
create mode 100644 static/images/twitter.png
create mode 100644 static/images/word_cloud_c1.png
create mode 100644 static/images/word_cloud_c2.png
create mode 100644 static/images/word_cloud_c3.png
create mode 100644 static/images/word_cloud_c4.png
create mode 100644 templates/header.html
create mode 100644 templates/home.html

```

Figura 11: Agregar comentario de confirmación de la versión cargada.

11.6. Deploy en Heroku

Crear un nombre para el repositorio dentro de Heroku con el siguiente comando.

```
> heroku create metro-quito
```

```

C:\Users\Jorge\Documents\tesis>heroku create metro-quito
Creating ⬢ metro-quito... done
https://metro-quito.herokuapp.com/ | https://git.heroku.com/metro-quito.git

```

Figura 12: Crear repositorio dentro de Heroku

Cargar el contenido de los archivos del repositorio local hacia Heroku con el siguiente comando.

```
> git push heroku
```

Finamente se puede visualizar el prototipo de sistema para el análisis de opiniones basado en tweets a través del enlace que nos aparece al final de la ejecución del comando.

```

C:\Users\Jorge\Documents\tesis>git push heroku
Enumerating objects: 34, done.
Counting objects: 100% (34/34), done.
Delta compression using up to 8 threads
Compressing objects: 100% (33/33), done.
Writing objects: 100% (34/34), 45.57 MiB | 9.03 MiB/s, done.
Total 34 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-22 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Python app detected
remote: -----> No Python version was specified. Using the buildpack default: python-3.11.2
remote: -----> To use a different version, see: https://devcenter.heroku.com/articles/python-runtimes
remote: -----> Installing python-3.11.2
remote: -----> Installing pip 23.0.1, setuptools 63.4.3 and wheel 0.38.4
remote: -----> Installing SQLite3
remote: -----> Installing requirements with pip
remote: Collecting flask
remote:   Downloading Flask-2.2.3-py3-none-any.whl (101 kB)
remote: Collecting dash
remote:   Downloading dash-2.9.2-py3-none-any.whl (10.2 MB)
remote: Collecting Plotly
remote:   Downloading plotly-5.14.0-py2.py3-none-any.whl (15.3 MB)
remote: Collecting pandas
remote:   Downloading pandas-1.5.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.0 MB)
remote: Collecting openpyxl
remote:   Downloading openpyxl-3.1.2-py2.py3-none-any.whl (249 kB)
remote: Collecting dash-bootstrap-components
remote:   Downloading dash_bootstrap_components-1.4.1-py3-none-any.whl (220 kB)
remote: Collecting gunicorn
remote:   Downloading gunicorn-20.1.0-py3-none-any.whl (79 kB)
remote: Collecting Werkzeug>=2.2.2
remote:   Downloading Werkzeug-2.2.3-py3-none-any.whl (233 kB)
remote: Collecting Jinja2>=3.0
remote:   Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
remote: Collecting itsdangerous>=2.0
remote:   Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
remote: Collecting click>=8.0
remote:   Downloading click-8.1.3-py3-none-any.whl (96 kB)
remote: Collecting dash-html-components==2.0.0
remote:   Downloading dash_html_components-2.0.0-py3-none-any.whl (4.1 kB)
remote: Collecting dash-core-components==2.0.0
remote:   Downloading dash_core_components-2.0.0-py3-none-any.whl (3.8 kB)
remote: Collecting dash-table==5.0.0
remote:   Downloading dash_table-5.0.0-py3-none-any.whl (3.9 kB)
remote: Collecting tenacity>=6.2.0
remote:   Downloading tenacity-8.2.2-py3-none-any.whl (24 kB)
remote: Collecting packaging
remote:   Downloading packaging-23.0-py3-none-any.whl (42 kB)
remote: Collecting python-dateutil>=2.8.1
remote:   Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
remote: Collecting pytz>=2020.1
remote:   Downloading pytz-2023.3-py2.py3-none-any.whl (502 kB)
remote: Collecting numpy>=1.21.0
remote:   Downloading numpy-1.24.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.3 MB)
remote: Collecting et-xmlfile
remote:   Downloading et_xmlfile-1.1.0-py3-none-any.whl (4.7 kB)
remote: Collecting MarkupSafe>=2.0
remote:   Downloading MarkupSafe-2.1.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (27 kB)
remote: Collecting six>=1.5
remote:   Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
remote: Installing collected packages: pytz, dash-table, dash-html-components, dash-core-components, tenacity, six,
python-dateutil, Plotly, openpyxl, Jinja2, pandas, flask, dash, dash-bootstrap-components
remote: Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.2 Plotly-5.14.0 Werkzeug-2.2.3 click-8.1.3 dash-2.9.2 da
dash-table-5.0.0 et-xmlfile-1.1.0 flask-2.2.3 gunicorn-20.1.0 itsdangerous-2.1.2 numpy-1.24.2 openpyxl-3.1.2 packaging-23.0 par
remote: -----> Discovering process types
remote: Procfile declares types -> web
remote:
remote: -----> Compressing...
remote: Done: 139.9M
remote: -----> Launching...
remote: Released v3
remote: https://metro-quito.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/metro-quito.git
* [new branch] master -> master

```

Figura 13: Cargar y ejecutar los archivos locales en el ambiente de Heroku.