



Máster Avanzado de Programación en Python  
para Hacking, BigData y Machine Learning

**Asignatura: Buenas practicas de**  
**Programacion**



Autor: Pedro Jesús Navarro Blázquez

Email: [petrus\\_nvm@hotmail.com](mailto:petrus_nvm@hotmail.com)

Actividad relacionada con la lección 3:

Como hemos estudiado en esta lección existen diversas herramientas que nos facilitan la tarea de documentar nuestro código. Para la actividad de esta lección se proponen dos ejercicios:

1. **Busque una herramienta alternativa a Sphinx para generar la documentación de código en Python.**

La herramienta alternativa encontrada es Mkdocs.

MkDocs es un generador de sitios estáticos rápido, simple y francamente magnífico que está orientado a la creación de documentación de proyectos. Los archivos de origen de la documentación se escriben en Markdown y se configuran con un solo archivo de configuración YAML.

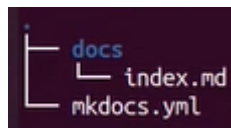
Usando este video he realizado un ejemplo básico de documentación con MkDocs

<https://www.youtube.com/watch?v=NuNj75iE8KA>

Pasos:

1. Paso-1: Se instala el paquete mkdocs con el comando:  
`pip install mkdocs.`
2. Paso-2: Se crea un directorio llamado hello-pkg con el comando:  
`mkdocs new hello-pkg`
3. Paso-3: Comprobamos que se ha creado correctamente el directorio con el comando:

`tree`



4. Paso-4: Modificamos el archivo mkdocs.yml:

```
Actividad > hello-pkg > ! mkdocs.yml
1  site_name: hello-pkg
2  site_url: https://example.com
3  nav:
4  |   - Home: index.md
```

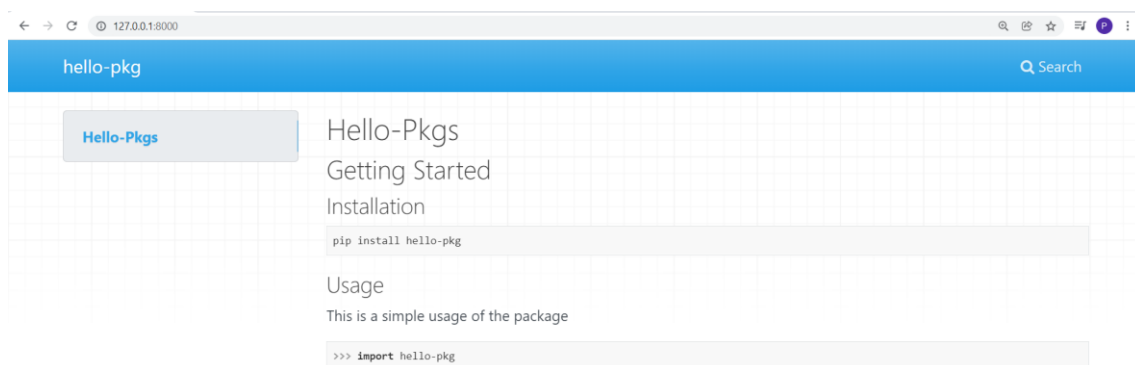
5. Paso-5: Modificamos el archivo index.md

```
Actividad > hello-pkg > docs > index.md > ## Hello-Pkgs > ### Getting Started > #### Usage
1  ## Hello-Pkgs
2
3  ### Getting Started
4
5  #### Installation
6  ```bash
7  pip install hello-pkg
8  ```
9  #### Usage
10 This is a simple usage of the package
11 ```python
12 >>> import hello-pkg
13 ```
```

6. Paso-6: Hacemos el build para mostrarlo en el navegador con el comando:

mkdocs serve

```
dirigida por la documentación\Actividad\hello-pkg>mkdocs serve
INFO - Building documentation...
INFO - Cleaning site directory
INFO - Documentation built in 0.08 seconds
INFO - [00:31:46] Serving on http://127.0.0.1:8000/
INFO - [00:31:47] Browser connected: http://127.0.0.1:8000/
```



7. Paso-7: Creamos los siguientes archivos: userguide.md, apireference.md y about.md.

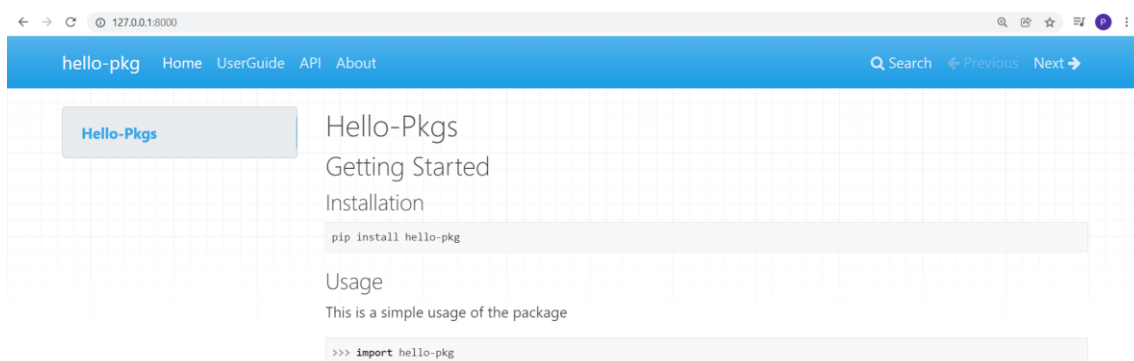
```
Actividad > hello-pkg > docs > userguide.md > ### User Guide > #### Usage
1  ### User Guide
2
3  #### Usage
4  This is how to use a package
```

```
Actividad > hello-pkg > docs > apireference.md > ### API Reference > #### API
1  ### API Reference
2
3
4  #### API
5  ***hello-pkgs***
```

```
Actividad > hello-pkg > docs > about.md > ### About Pkgs
1  ### About Pkgs
2  + An exciting hello-pkgs
```

8. Paso-8: Añadimos los archivos creados al archivo mkdocs.yml para que se puedan mostrar en la pagina web.

```
Actividad > hello-pkg > ! mkdocs.yml
1  site_name: hello-pkg
2  site_url: https://example.com
3  nav:
4    - Home: index.md
5    - UserGuide: userguide.md
6    - API: apireference.md
7    - About: about.md
```

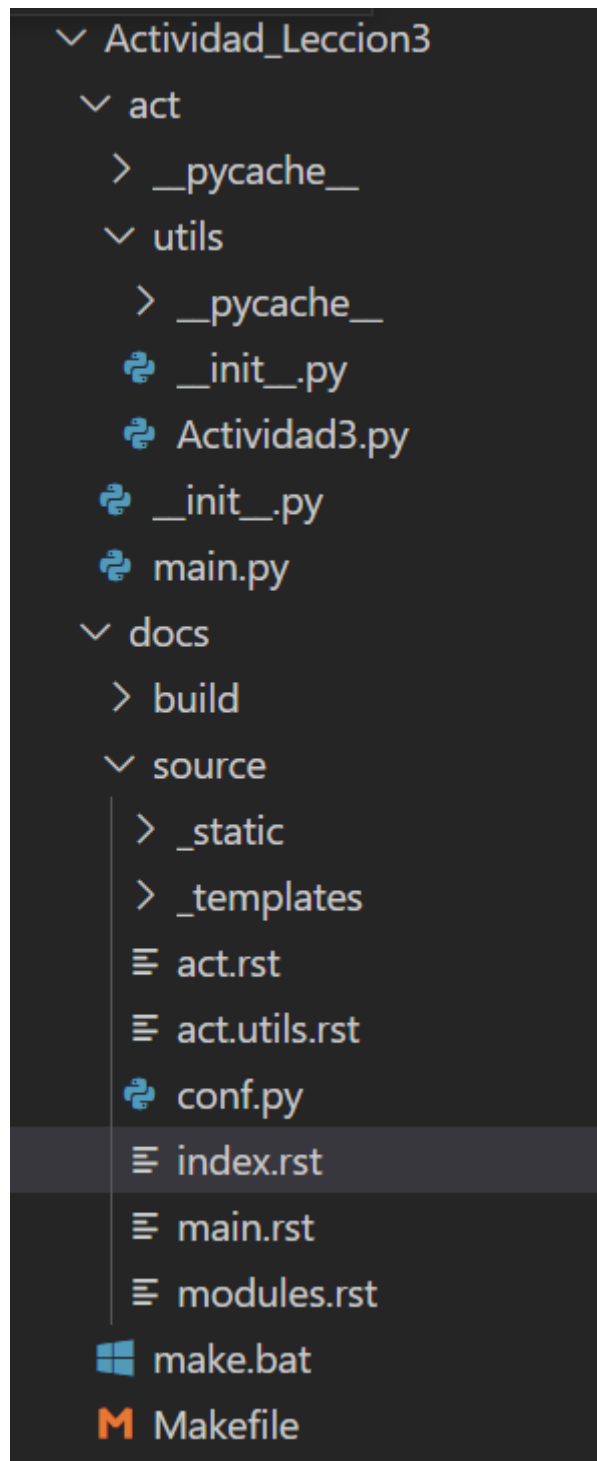


9. Paso-9: Para desplegarlo se utiliza el comando:

Mkdocs gh-deploy

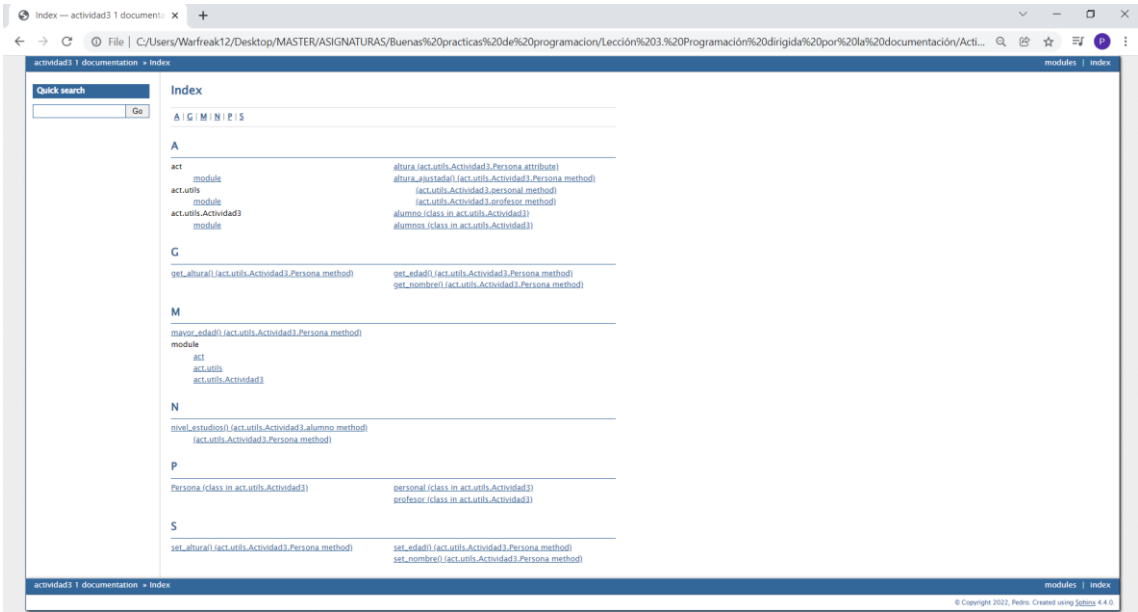
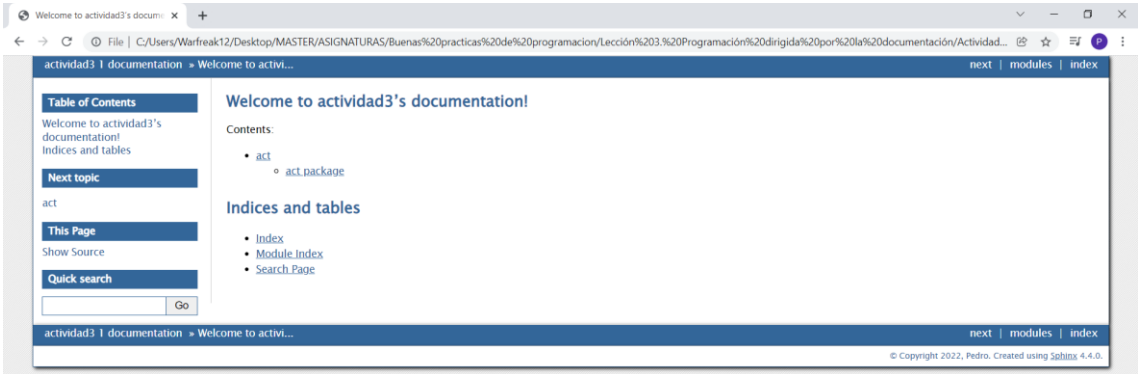
2. Realice la documentación de alguno de los códigos que ha desarrollado a lo largo de este curso. La documentación puede generarla con Sphinx o con la herramienta que ha buscado en el primer ejercicio.

A partir de la clase Persona se ha generado la documentación. Podemos apreciar tanto el árbol de los diferentes directorios como de la documentación creada en la pagina web.



## Lección-2

## Buenas Practicas de Programacion



January 19th, 2022