

# ED06\_2 - Integración de MkDocs con GitHub Pages

## 1. Paso 1: Configuración del Entorno

- crear un nuevo directorio

```
Ricar@Andres_BD MINGW64 ~/desktop (main)
$ mkdir mi-documentacion

Ricar@Andres_BD MINGW64 ~/desktop (main)
$ cd mi-documentacion

Ricar@Andres_BD MINGW64 ~/desktop/mi-documentacion (main)
$
```

- crear un entorno virtual
- activa el entorno virtual
- instalar MkDocs y el tema material

```
PS C:\Users\Ricar\Desktop\mi-documentacion> python -m venv venv
PS C:\Users\Ricar\Desktop\mi-documentacion> venv\Scripts\activate.bat
PS C:\Users\Ricar\Desktop\mi-documentacion> pip install mkdocs mkdocs-material
```

## 2. Paso 2: Crear el proyecto MkDocs

- Inicializa un nuevo proyecto MkDocs en el directorio actual
- Crea dos archivos: "mkdocs.yml" "docs/index.md"

```
PS C:\Users\Ricar\Desktop\mi-documentacion> mkdocs new .
INFO    - Writing config file: ./mkdocs.yml
INFO    - Writing initial docs: ./docs/index.md
PS C:\Users\Ricar\Desktop\mi-documentacion>
```

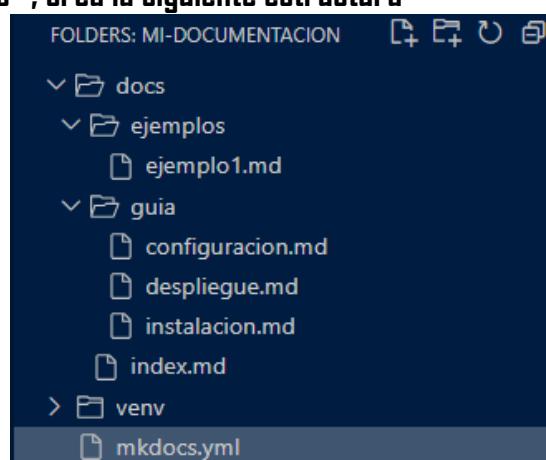
## 3. Paso 3: Configurar MkDocs

- Abre el archivo "mkdocs.yml" y configura el tema material

```
mkdocs.yml
1 site_name: Mi Documentacion
2 theme:
3   name: material
4   features:
5     - navigation.tabs
6     - navigation.sections
7     - navigation.expand
8     - content.code.copy
9
```

## 4. Paso 4: Crear la estructura de Documentación

- En la carpeta "docs" , crea la siguiente estructura

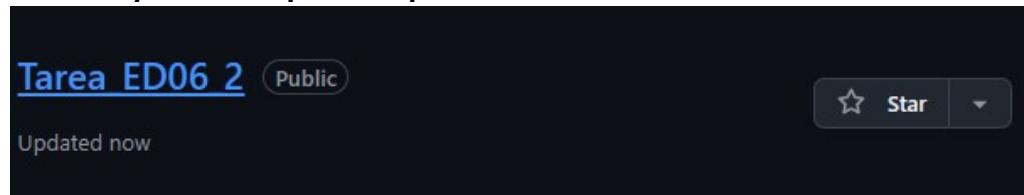


- Edita el archivo “docs/index.md”

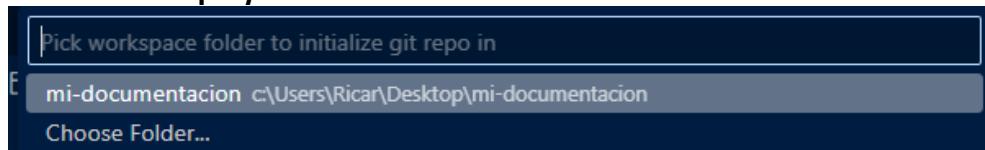
```
docs > index.md > # Bienvenido a Mi Documentacion > ## Contenido
1 # Bienvenido a Mi Documentacion
2
3 Esta es la pagina principal de mi documentacion
4
5 ## Contenido
6
7 - [Guía de instalacion](guia/installacion.md)
8 - [Configuración](guia/configuracion.md)
9 - [Despliegue](guia/despliegue.md)
```

## 5. Paso 5: Crear repositorio en GitHub

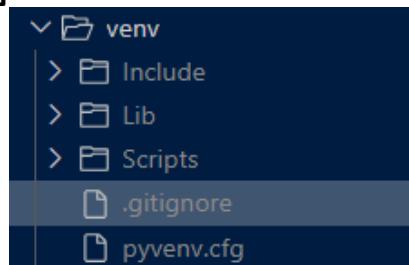
- Ve a GitHub y crea un repositorio publico



- Inicializa Git en tu proyecto local



- Crea un archivo “.gitignore”



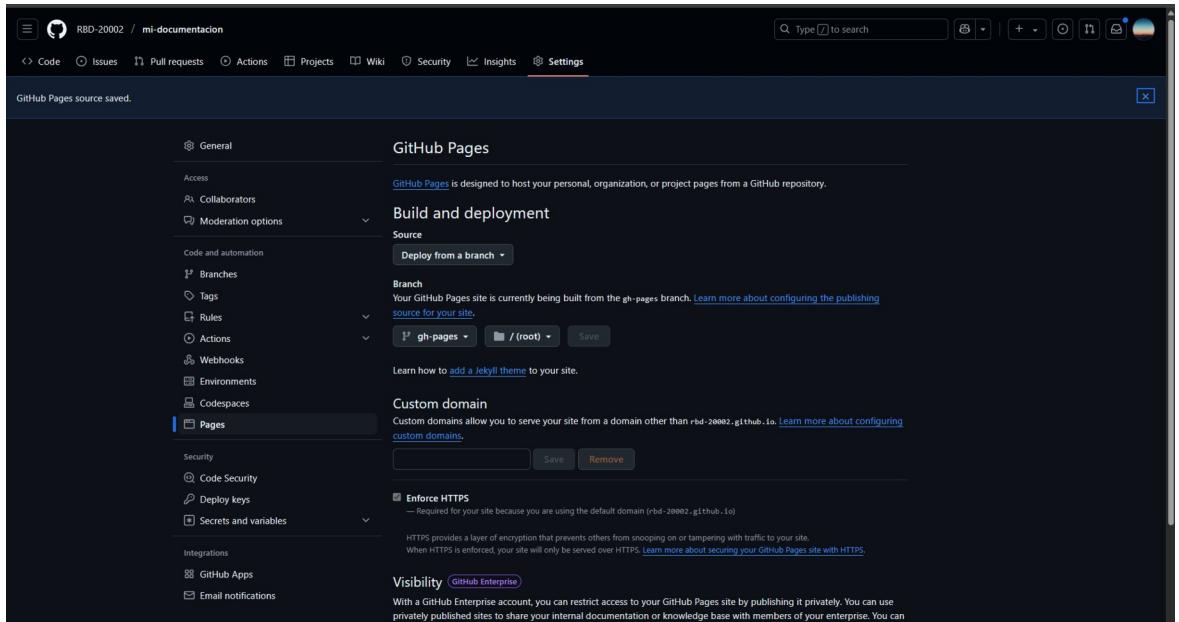
- Realiza el primer commit y vincula tu repositorio remoto

```
PS C:\Users\Ricar\Desktop\mi-documentacion> git add .
PS C:\Users\Ricar\Desktop\mi-documentacion> git commit -m "Primer Commit"
[main (root-commit) 5d3d20] Primer Commit
 6 files changed, 17 insertions(+)
 create mode 100644 docs/ejemplos/ejemplo1.md
 create mode 100644 docs/guia/configuracion.md
 create mode 100644 docs/guia/despliegue.md
 create mode 100644 docs/guia/installacion.md
 create mode 100644 docs/index.md
 create mode 100644 mkdocs.yml
PS C:\Users\Ricar\Desktop\mi-documentacion> git branch -M main
PS C:\Users\Ricar\Desktop\mi-documentacion> git remote add origin https://github.com/RBD-20002/Tarea_ED06_2
PS C:\Users\Ricar\Desktop\mi-documentacion> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 753 bytes | 125.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/RBD-20002/Tarea_ED06_2
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\Ricar\Desktop\mi-documentacion>
```

**COMENTARIO:** A partir de este punto se me complico varias veces a lo que me llevó facil varios días intentando entender el fallo y aunque lo conseguí no lo entendi del todo, porque al momento de que se ejecuta la “action” siempre me daba error pero lo solucione creando el workflow desde el GitHub y despues modificarlo en el VSC y recien la action me marco con check verde

## 6. Paso 6: Configurar GitHub Pages

- Ve a la configuración del repositorio y ve a la opción de “Pages” y en “Source” selecciona “Deploy from a branch”, y elige “gh-pages”. Luego guarda los cambios



- Crea un archivo “.github/workflows/deploy.yml” en el directorio raíz del repositorio con lo siguiente:

```
name: Deploy MkDocs
on:
  push:
    branches: [ main ]
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - uses: actions/setup-python@v2
        with:
          python-version: '3.x'
      - run: pip install mkdocs-material
      - run: mkdocs gh-deploy --force
```

## 7. Paso 8: Desplegar la documentación

- Haz commit de los cambios en el archivo `.github/workflows/deploy.yml` y realiza un push a la rama `main`

```
(venv) C:\Users\Ricar\Desktop\mi-documentacion>git add .

(venv) C:\Users\Ricar\Desktop\mi-documentacion>git commit -m "Configuracion de GitHub Pages"
[main b9af0db] Configuracion de GitHub Pages
 1 file changed, 11 insertions(+), 11 deletions(-)

(venv) C:\Users\Ricar\Desktop\mi-documentacion>git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 605 bytes | 302.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/RBD-20002/mi-documentacion
 293399b..b9af0db main -> main

(venv) C:\Users\Ricar\Desktop\mi-documentacion>
```

- Ve a la sección “Actions” de tu repositorio para ver el progreso del despliegue

The screenshot shows the GitHub Actions interface for the repository RBD-20002/mi-documentacion. The left sidebar has sections for Actions, Deploy MkDocs, Management, Caches, Deployments, and Attestations. The main area shows 'All workflows' with two runs listed:

- pages build and deployment**: Triggered by RBD-20002, completed 3 minutes ago, status: success.
- Configuración de GitHub Pages**: Triggered by a push to the main branch, completed 4 minutes ago, status: success.

- Una vez completado , la documentación estara: "<https://rbd-20002.github.io/mi-documentacion/>"

Bienvenido a Mi Documentación

Bienvenido a Mi Documentación

Contenido

- Guía de Instalación
- Configuración
- Despliegue

Table of contents

Contenido

**ENLACE A LA DOCUMENTACION: <https://rbd-20002.github.io/mi-documentacion/>**