
Repo de plantilla sphinx

Juan Bautista Talens

11 d'ag., 2025

1	Guia ràpida	3
1.1	1) Desenvolupament en local (no és necessari)	3
1.2	2) Estructura	3
1.3	3) Publicació	4
2	Personalització	5
2.1	Tema i opcions	5
2.2	Logos i favicon	5
2.3	CSS propi	5
3	Duplicar la plantilla amb l'script	7
3.1	Ús	7
3.2	Publicació	8
4	Estat del projecte	9

Benvinguda! Esta és la portada de la plantilla. Baix tens una vista ràpida i enllaços útils.

Guia ràpida **Comandes** i flux bàsic per a treballar en local i publicar.

[guia-rapida](#) Personalització Canvia tema, logo, CSS i opcions del conf .py.

[personalitzacio](#) Duplicar amb un script Com usar `scripts/nou_sphinx_repo.sh` per a clonar i
personalitzar.

[duplicar-plantilla](#)

1.1 1) Desenvolupament en local(no és necessari)

No és necessari treballar en local: la plantilla ja està preparada perquè GitHub compile la documentació i la publiqui automàticament quan faces push a main. Si vols treballar i previsualitzar en local abans de publicar, segueix les instruccions indicades:

```
# 1) Crear i activar l'entorn virtual
python3 -m venv .venv
source .venv/bin/activate

# 2) Instal·lar dependències (o usa el contenidor del workflow)
pip install -r requirements.txt

# 3) Compilar l'HTML (des de l'arrel del repo)
sphinx-build -b html docs _build/html

# 4) Obrir en el navegador (tria el teu sistema)
xdg-open _build/html/index.html      # Linux
# open _build/html/index.html        # macOS
# start _build/html/index.html        # Windows (PowerShell)
```

1.2 2) Estructura

```
docs/
  index.md
  guia-rapida.md
  personalitzacio.md
  duplicar-plantilla.md
  _static/
```

(continua a la pàgina següent)

(ve de la pàgina anterior)

```
_templates/  
conf.py
```

1.3 3) Publicació

Només cal fer *push* a main. El workflow compila **HTML** i **PDF** i els puja a gh-pages. Abans, guarda i envia els canvis amb:

```
git add .  
git commit -m "Actualització de contingut"  
git push
```


2.1 Tema i opcions

El tema està en `conf.py`:

```
html_theme = "pydata_sphinx_theme"
html_theme_options = { "show_prev_next": True, ... }
```

2.2 Logos i favicon

- `html_logo = "_static/assets/img/logos/logoJust.png"`
- `html_favicon = "_static/assets/img/logos/logo50.ico"`

2.3 CSS propi

Afig els teus CSS a `_static/assets/stylesheet/` i declara'ls en `html_css_files` (ordre de més general a més específic).

Duplicar la plantilla amb l'script

Este script automatitza la creació d'un nou repositori Sphinx:

- Clona l'origen (per defecte, la plantilla oficial),
- Reescriu títols i URL en `conf.py`,
- Ajusta el nom del PDF al workflow,
- Crea el repo a GitHub i activa GitHub Pages (`gh-pages`).

3.1 Ús

Per a duplicar la plantilla, executa el següent comandament des de la carpeta del projecte:

```
./scripts/nou_sphinx_repo.sh NOU_REPO "Títol nou" UsuariGitHub [ORIGEN] [NomPDF.pdf]
```

3.1.1 Paràmetres

- `NOU_REPO`: nom curt del nou repositori GitHub.
 - `"Títol nou"`: títol que apareixerà al lloc web generat.
 - `UsuariGitHub`: nom d'usuari o organització on es crearà el repo.
 - `[ORIGEN]` (*opcional*): URL del repo base (per defecte, la plantilla).
 - `[NomPDF.pdf]` (*opcional*): nom fix per al PDF generat.
-

3.1.2 Exemples

```
# Clonar la plantilla bàsica
./scripts/nou_sphinx_repo.sh plantilla-sphinx "Repo de plantilla sphinx" juatafe

# Clonar un repositori diferent i fixar nom del PDF
./scripts/nou_sphinx_repo.sh apunts-xarxes "Apunts de Xarxes" juatafe https://github.com/
→ juatafe/sge.git ApuntsDeXarxes.pdf
```

3.2 Publicació

Una vegada creat el repo, només cal afegir canvis i pujar-los:

```
git add .
git commit -m "Primera personalització"
git push origin main
```

GitHub compilarà automàticament **HTML + PDF** i publicarà el lloc a:

```
https://<usuari>.github.io/<repo>
```

Estat del projecte

- Tema: **pydata-sphinx-theme**
- PDF: es publica en pdf/<slug>.pdf i apareix com a icona a la barra superior.
- Build & Deploy: automàtic via **GitHub Actions** → gh-pages.