Repo de plantilla sphinx

Juan Bautista Talens

15 d'ag., 2025

Índex

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

CAPÍTOL 1

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`.

1.1 Guia ràpida

1.1.1 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 publique 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)
```

1.1.2 2) Estructura

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

1.1.3 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
```

1.2 Personalització

El fitxer `conf.py` és el cor del projecte Sphinx. Ací pots configurar el tema, els estils, els logos, les opcions de navegació, i la generació del PDF.

1.2.1 Tema i opcions visuals

El tema utilitzat és **PyData Sphinx Theme**, un tema modern i molt configurable:

```
html_theme = "pydata_sphinx_theme"
html_theme_options = {
    "show_nav_level": 1,          # Mostra des del nivell 1 del TOC
    "navigation_depth": 4,        # Profunditat màxima del menú lateral
    "collapse_navigation": False, # Evita que es col·lapse automàticament
    "secondary_sidebar_items": ["page-toc", "sourcelink", "edit-this-page"],
    "use_edit_page_button": True,
    "show_prev_next": True,
    "show_toc_level": 2,
    "navbar_end": ["theme-switcher", "navbar-icon-links"],
    "icon_links": [
        {"name": "GitHub", "url": "...", "icon": "fa-brands fa-github"},
        {"name": "Issues", "url": "...", "icon": "fa-solid fa-circle-exclamation"},
        {"name": "PDF", "url": pdf_url, "icon": "fa-solid fa-file-pdf"},
    ],
}
```

1.2.2 Logos i favicon

Es defineixen així dins del bloc HTML:

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

Pots substituir-los per les teues pròpies imatges mantenint la ruta dins de `docs/_static/...`.

1.2.3 Fulls d'estil (CSS)

El paràmetre `html_css_files` permet carregar fulls d'estil personalitzats. L'ordre és important: van del més general al més específic.

```
html_css_files = [
    "assets/stylesheets/extracsspdf.css",  # estil per a PDF
    "assets/stylesheets/customs.css",      # layout general
    "assets/stylesheets/extra.css",        # modificacions puntuals
]
```

Els fitxers han d'estar dins de `docs/_static/`. Recomanat: crea subcarpetes (`assets/stylesheets/`) per mantindre net el projecte.

1.2.4 Plantilles HTML i rutes

Si necessites modificar la distribució HTML, declara la carpeta de plantilles:

```
templates_path = ["_templates"]
```

Aquesta ruta conté fitxers `.html` que sobreescriuen parcialment els del tema. Pots personalitzar, per exemple, el layout base, els blocs de la barra lateral o la capçalera.

1.2.5 PDF: títol, fonts i llengua

La generació del PDF via LaTeX es configura amb:

```
latex_engine = "xelatex"

latex_elements = {
    "fontpkg": r"""
\usepackage{fontspec}
\setmainfont{TeX Gyre Pagella}
\setsansfont{TeX Gyre Heros}
\setmonofont{Latin Modern Mono}
""",
    "preamble": r"""
\usepackage{polyglossia}
\setmainlanguage{catalan}
""",
}
```

Aquest fragment assegura que el PDF utilitza fonts modernes i correcte llengua (valencià/català). Pots substituir les fonts si ho necessites.

1.2.6 PDF: nom fix segons el repositori

El nom del PDF es genera automàticament a partir del repositori GitHub:

```
_repo = os.environ.get("GITHUB_REPOSITORY", "")
site_slug = _repo.split("/")[-1] if _repo else slugify(project)
pdf_url = f"pdf/{site_slug}.pdf"
```

Això assegura que el fitxer `.pdf` tinga un nom coherent amb el repo (ideal per a repositoris educatius).

1.3 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`).

1.3.1 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]
```

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.

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
```

1.3.2 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>
```