



```
# Pipeline Híbrido DOCX→PDF (Local → Produção)
```

Objetivo

- Converter DOCX localmente e publicar o PDF no storage de produção, sem conversão

Fluxo Operacional

1. Origem: usar DOCX do artigo (local ou baixar do storage).
2. Conversão local: Microsoft Word (docx2pdf) ou LibreOffice headless.
3. Idempotência: calcular hash (SHA-256) do PDF; pular upload se não houver mudança.
4. Upload: enviar para storage remoto (ex.: S3) em `pdfs/artigos/<slug>.pdf` .
5. Atualização: setar `Artigo arquivo_pdf` e salvar.
6. Logs: registrar sucesso/falha; nunca derrubar o app por erro de conversão.

Checklist de Implementação

- [] Remover/confirmar remoção da conversão automática no servidor.
- [] Criar comando/endpoint para atualizar `arquivo_pdf` com segurança.
- [] PS1 orquestrador local: baixar DOCX (se preciso), converter, hash, upload, etc.
- [] Flags de settings: `ENABLE_DOCX_PDF_CONVERSION=False` em produção.
- [] Templates com guardas para `.url` em campos de arquivo.
- [] Logging `WARNING+` e sem access log do Gunicorn.
- [] Política de acesso no bucket (Get/Put/Delete restritos).

Boas Práticas

- Conversores/helpers em `Apenas_Local/` ou app `devtools` (somente com `DEBUG=True`).
- `--dry-run` no PS1 antes de publicar.
- Documentar passos e plano de rollback.

"""

```
md_path.write_text(md, encoding="utf-8")
```

```
out_pdf, md_path
```



Resultado

```
(PastaPath/1/mnt/dato/Proximo_Veneno_Dr_Albino_Marks_v2.pdf)
```

Pergunte alguma coisa

+ Pensamento estendido ▾



[Pular para o conteúdo](#)

Pular para o conteúdo

Pular para o conteúdo

[Pular para o conteúdo](#)

[Pular para o conteúdo](#)

Pular para o conteúdo

[Pular para o conteúdo](#)

Pular para o conteúdo

[Pular para o conteúdo](#)

Pular para o conteúdo

[Pular para o conteúdo](#)

Pular para o conteúdo

[Pular para o conteúdo](#)

Pular para o conteúdo

[Pular para o conteúdo](#)

