
Clase 232 — Portafolio público en GitHub Pages y presentación

Parte: 8 — Capstones · Fuentes: Chip Huyen, How to build a data science portfolio + Eugene Yan, What I learned from looking at 200 ML portfolios + Mitchell et al., Model Cards for Model Reporting (FAT* 2019). Duración estimada: 150 min.

Clase 232 — Portafolio público en GitHub Pages y presentación

Parte: 8 — Capstones · Fuentes: Chip Huyen, *How to build a data science portfolio* + Eugene Yan, *What I learned from looking at 200 ML portfolios* + Mitchell et al., *Model Cards for Model Reporting (FAT* 2019)*. Duración estimada: 150 min.

Objetivo

Empaquetar los tres capstones (229, 230, 231) en un portafolio público que un reclutador entienda en 30 segundos: sitio en GitHub Pages, demos hosted, blog técnico, deck de presentación y CV. La regla es calidad > cantidad: 3 proyectos terminados con métricas honestas valen más que 10 a medias.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Estructurar un sitio de portafolio con MkDocs Material o Quarto desplegado en GitHub Pages.
- Escribir un README de proyecto que comunica problema, approach, métricas y limitaciones sin marketing vacío.
- Publicar demos hosted (Streamlit Community Cloud, HuggingFace Spaces) y un blog técnico por capstone.
- Armar un deck de 10-15 slides que cuenta cada proyecto en formato problema → datos → enfoque → resultado → trade-offs.
- Detectar y evitar los antipatterns clásicos (10 proyectos a medias, README sin números, demo rota).

Componentes del portafolio

#	Componente	Para qué sirve
1	README pulido por proyecto	Es la primera impresión. Badge de CI, hero
2	Demos hosted (Streamlit/HF Spaces)	El reclutador clickea y prueba en 10 seg.
3	Blog técnico (1 post por capstone)	Demuestra que sabés comunicar además de co
4	Presentación (deck 10-15 slides)	Para entrevistas técnicas y meetups. Reusa
5	Video demo (2-3 min)	Loom o screencast. Único formato consumibl
6	CV técnico (1 página)	Verbo de impacto + número medible. Linkea

Definiciones y características

- Portafolio: sitio público (no PDF) que centraliza proyectos, blog, CV y links de contacto. Vive en username.github.io o subdominio propio.
- GitHub Pages: hosting estático gratis de GitHub. Build con Jekyll por defecto, pero podés deployar cualquier sitio estático (MkDocs, Quarto, Docusaurus, Hugo).
- MkDocs Material: generador estático en Python; búsqueda built-in, dark mode, navegación lateral. Lo más rápido de levantar.
- Quarto: generador científico (Posit/RStudio); ejecuta notebooks .qmd/.ipynb y renderiza con bibliografía BibTeX y matemáticas LaTeX. Ideal si tu blog tiene mucho código + plots.

- Model Card (Mitchell et al., 2019): ficha estándar de un modelo — datos de entrenamiento, métricas por subgrupo, casos de uso previstos y NO previstos, riesgos éticos.
- Hero image / GIF: imagen o screencast al inicio del README que muestra el output en 1 vistazo. Sin hero, nadie scrollea.
- Quick start de 30 segundos: bloque bash con 3-5 líneas para correr el proyecto local. Si no entra en 30 seg, perdiste al reviewer.
- Reproducibilidad (Parte 7): lock file + Dockerfile + CI verde + dataset documentado. Es el diferenciador entre "proyecto de tutorial" y "proyecto serio".

Recursos

- Herramientas: MkDocs Material, Quarto, Streamlit Community Cloud, HuggingFace Spaces, Render, Vercel.
- Templates: mkdocs-material starter, Quarto blog template.
- Inspiración: portfolios de Eugene Yan, Chip Huyen, Vicki Boykis, Hamel Husain — todos en eugeneyan.com, huyenchip.com, etc.

Ejercicios

1. MkDocs Material setup: pip install mkdocs-material, mkdocs new portfolio, editar mkdocs.yml con tema material y deployar a gh-pages con mkdocs gh-deploy. Verificar que el sitio carga en <https://<user>.github.io/portfolio>.
2. README de capstone: tomar el capstone 229 (NLP) y reescribir el README con hero image, badges (CI, license, Python version), quick start de 30 seg, sección "decisiones técnicas" (3 bullets) y "limitaciones" (3 bullets).
3. Demo hosted: subir el capstone 230 (CV) a HuggingFace Spaces con Gradio. URL pública compartible.
4. Blog post técnico: escribir 1 post de 800-1500 palabras sobre el capstone 231 (tabular) siguiendo el outline problema → datos → approach → resultado con un plot → trade-offs → próximos pasos.
5. Deck: armar 10-15 slides (Google Slides, Marp o reveal.js) presentando los 3 capstones con la regla de "1 idea por slide".

Homework verificable

Sitio en GitHub Pages con:

1. Index con foto, bio de 3 líneas, links a 3 proyectos, link a blog, link a CV PDF, link a LinkedIn/GitHub.
2. Una página por proyecto (3 capstones) con README + Model Card + link a demo + link a blog post + link a repo.
3. Blog con al menos 1 post técnico publicado.
4. CV PDF de 1 página linkeado desde el index.
5. Demos: al menos 1 de los 3 proyectos con demo hosted funcionando (Streamlit Cloud o HF Spaces).

Criterio de aceptación: un reclutador no técnico abre el sitio, en 30 segundos sabe (a) tu nombre, (b) qué haces, (c) ve un proyecto con métricas, (d) puede clicar una demo que funciona. Si falla algún paso → el portafolio no aprueba.

Errores comunes

Síntoma	Causa y cómo arreglar
README dice "implementé un modelo de ML pa	Cero métricas, cero credibilidad. Fix: pon
10 proyectos en el portafolio, todos a 30%	Cantidad sin profundidad. Fix: borrar 7, d
Demo rota (link a Streamlit que duerme y n	Streamlit free duerme tras 7 días sin uso,
Hero image inexistente — README es solo te	Sin imagen no scrollean. Fix: GIF de 5 seg
requirements.txt con pandas sin versión	Imposible reproducir. Fix: lock file (uv.l
CV de 3 páginas con responsabilidades gené	Nadie lee la pág 2. Fix: 1 página, verbos

Preguntas frecuentes

¿MkDocs Material o Quarto?

MkDocs Material si el sitio es documentación + landing. Quarto si el sitio tiene mucho notebook ejecutable y math (papers, posts técnicos largos con BibTeX). Ambos deployan a GitHub Pages igual.

¿Necesito dominio propio?

No para empezar. username.github.io funciona perfecto. Comprá dominio (tunombre.dev, ~12 USD/año) cuando el portafolio esté en versión estable y quieras una URL más memorable.

¿Y si no tengo experiencia laboral todavía?

El portafolio reemplaza la falta de experiencia laboral. 3 capstones con métricas, demo y blog post pesan más en una entrevista junior que "tomé 5 cursos en Coursera".

¿Subo el código del homework de las clases?

No al portafolio principal. Mantenelos en un repo aparte (python-data-science-program-homework). El portafolio es solo para proyectos pulidos.

¿Cuánto tiempo dedicar al portafolio antes de aplicar a trabajos?

Regla práctica: 1 semana de pulido full-time tras terminar el capstone 231. Pasada esa semana, aplicás aunque no sea perfecto — el portafolio se itera con feedback real de entrevistas.

Referencias

- Chip Huyen, How to build a data science portfolio — huyenchip.com.
- Eugene Yan, What I learned from looking at 200 ML portfolios — eugeneyan.com.
- MkDocs Material — documentación oficial.
- Quarto — sitio oficial, especialmente Websites y Blogs.
- HuggingFace Spaces — hosting gratis para demos.
- Mitchell, M. et al. Model Cards for Model Reporting (FAT* 2019) — arxiv:1810.03993.

Cierre del programa

Llegaste a la clase 232 de 232. Empezaste con print("hola mundo") y terminás con un portafolio público, tres capstones con métricas honestas, demos hosted y un blog que explica cómo pensás. El programa termina acá; tu trabajo como practicante de data science recién arranca. Lo que sigue no es otro curso — es elegir una vertical (research, applied science, MLE, product DS), buscar a la gente que está haciendo el trabajo que querés hacer, y empezar a contribuir: issues en open source, posts en tu blog, charlas en meetups, un

proyecto nuevo cada trimestre. La ventaja competitiva no es saber más algoritmos: es enviar cosas terminadas, medibles y reproducibles. Eso ya lo sabés hacer.

Gracias por llegar hasta acá. Ahora andá a romperla.

[Volver al índice general](#) · [Índice de Parte 8](#)

Apéndice: notebook (primer bloque)

Esta clase es más de producto y comunicación que de código ML. El notebook es un utilitario práctico: define el schema de un proyecto, genera los .md del portafolio, valida que cumpla mínimos, y emite los stubs de mkdocs.yml + workflow de Pages. Solo stdlib + jinja2 opcional. Seed 42. Self-contained.

```
from dataclasses import dataclass, field, asdict
from typing import Optional
import textwrap, random

random.seed(42)

try:
    from jinja2 import Template
    HAS_JINJA = True
except ImportError:
    HAS_JINJA = False
print(f'jinja2 disponible: {HAS_JINJA} (no es bloqueante, usamos f-strings de fallback)')

@dataclass
class Project:
    title: str
    slug: str
    problem: str          # 1-2 frases en lenguaje humano
    approach: str         # 2-3 frases técnicas
    metrics: dict         # {'metric_name': value}
    stack: list           # ['python', 'pytorch', ...]
    repo_url: str
    demo_url: Optional[str] = None
    blog_url: Optional[str] = None
    limitations: list = field(default_factory=list)
    hero_image: Optional[str] = None
```

Archivos complementarios

- notebook.ipynb