DÍA 13 — GitHub + README Profesional

```
Día Tema nuevo principal    Repaso integrado    Mini práctica sugerida
13  GitHub + README profesional Git avanzado, proyecto final    Subir proyecto completo, crear
README con badges e instrucciones
```

Objetivo del día

```
   Aprender a subir tu proyecto completo a GitHub de forma limpia y profesional.
   Redactar un README.md atractivo: instrucciones, badges, capturas de pantalla.
   Reforzar buenas prácticas de repositorios: ramas, commits limpios, .gitignore.
   Sentar las bases de tu portfolio online.
```

Concepto clave: GitHub = tu escaparate

```
Analogía: Tu repositorio de GitHub es como un escaparate de tienda online.
Quieres que cuando alguien lo vea, entienda qué hace tu proyecto, cómo probarlo, y vea que está
ordenado.
```

Estructura mínima de tu proyecto final

```
/proyecto_web_logica/
│
├── index.html
├── /css/
│   └── style.css
├── /scripts/
│   └── procesar.py
├── README.md
├── .gitignore
└── LICENSE (opcional)
```

Buenas prácticas para tu README.md

```
   Empieza con un título claro y un pequeño párrafo describiendo qué hace tu proyecto.
   Incluye captura de pantalla (puedes arrastrarla a tu repo en GitHub).
   Usa badges si quieres: versión, licencia, build passing, etc. (shields.io).
   Añade Instrucciones de instalación (dependencias, cómo correrlo).
   Explica cómo contribuir si quieres que otros lo usen.
   Incluye Autor / Créditos / Licencia.
```

Ejemplo básico de README.md

```
#    Proyecto Encuesta Web

Pequeña web para practicar integración de lógica Python con Frontend.

![Captura](./captura.png)

##    Estructura

/proyecto_web_logica/
│
├── index.html
├── /css/
│   └── style.css
├── /scripts/
│   └── procesar.py

## ⚙ Cómo usar

1. Clona el repo.
2. Ejecuta `python3 scripts/procesar.py` desde terminal.
3. Modifica `index.html` a tu gusto.

##    Tecnologías

- HTML5
- CSS3
- Python3

##      Autor

- Zokan Dev

##    Licencia

MIT License
```

```
   Reto guiado del día
    1.  Crea un nuevo repositorio en GitHub llamado proyecto_web_logica.
    2.  Sube todo tu proyecto (index.html, style.css, scripts/).
    3.  Crea tu README.md con toda la info clave.
    4.  Añade un .gitignore (ej: __pycache__/, .vscode/).
    5.  Haz commits claros: Inicial commit, Añadir estructura básica, Actualizar README.
    6.  Verifica que tu repositorio se vea ordenado y entendible.
```

```
   Frase para recordar

"Tu README es la puerta de entrada: si no lo entienden, nadie entrará a mirar tu código."
```

Configuración .gitignore

1. ¿Qué hacemos?
   - Creamos un archivo .gitignore en la raíz del proyecto.
   - Dentro del archivo escribimos las rutas y patrones de lo que NO queremos subir al repositorio.

⚙ 2. ¿Qué contiene?

```
__pycache__/
.vscode/
*.log
```

- __pycache__/ → Carpeta de archivos compilados de Python (basura temporal).
- .vscode/ → Configuración local del editor (preferencias personales).
- *.log → Archivos de log o registro que se generan en local.

3. ¿Qué hace exactamente?

Git ignora estos archivos y carpetas:
- No se añaden al repositorio.
- No se suben a GitHub.
- Se quedan solo en tu máquina local.

4. Clave para recordar

.gitignore es como una lista de 'NO subir' para proteger tu repo de basura y datos sensibles.

Con esto mantienes tu proyecto limpio, seguro y profesional.