Documentación técnica del proyecto: El Heraldo del Viejo Mundo

1. Resumen del proyecto

El Heraldo del Viejo Mundo es un sistema automatizado que genera y publica artículos en un blog Jekyll (desplegado mediante GitHub Pages) sobre Warhammer: The Old World.

El sistema incluye dos bots independientes que operan desde un servidor Hetzner con Ubuntu 22.04. Ambos utilizan Python 3.10 y la API de OpenAI (GPT-4o) para generar contenido en español de forma automatizada.

Los contenidos se generan como archivos Markdown, que se publican automáticamente mediante git.

2. Estructura de carpetas

- `/opt/oldworldbot/`: contiene los scripts `bot.py`, `video_bot.py`, y sus logs. Guarda temporalmente los archivos `.md` generados.
- `/opt/oldworldbot/posts/`: destino temporal de los artículos/vídeos nuevos antes de moverlos al repositorio.
- `/opt/Heraldo-del-Viejo-Mundo/`: repositorio Jekyll que publica el blog.
- `/opt/Heraldo-del-Viejo-Mundo/_posts/`: destino final de los `.md` antes de ser publicados.
- `_layouts/post.html`: plantilla usada para cada entrada del blog, modificada para incluir comentarios Giscus.

3. Descripción de bots

- `bot.py`: scrapea artículos de la sección 'The Old World' en Warhammer Community usando Playwright (headless). Usa GPT-4o para traducir títulos, redactar resumen, cuerpo del artículo y conclusiones. Actualmente **no funciona correctamente**: no detecta artículos nuevos, solo los definidos manualmente en `FIXED_ARTICLES`.

- `video_bot.py`: funciona correctamente. Cada hora revisa los canales aprobados en YouTube,

detecta vídeos nuevos, llama a GPT para generar un resumen informativo, y publica un post en Markdown con miniatura y reproductor embebido.

4. Flujo de trabajo de los bots

- El bot (artículos o vídeos) detecta nuevos contenidos.
- Extrae HTML o metadatos.
- Llama a GPT-4o para generar texto.
- Genera archivo `.md` con metadatos YAML.
- Si el archivo no existe o ha cambiado, lo guarda.
- Se mueve automáticamente a `_posts/` y se ejecuta:
  git add + commit + push al repositorio GitHub Pages.

5. Automatización

- `video_bot.py` se ejecuta por cron cada hora a los 5 minutos (`5 * * * *`).
- Los resultados se guardan en `posts/`, y luego se publican.
- Se registra todo en `/opt/oldworldbot/cron.log`.
- `bot.py` debe ejecutarse manualmente por ahora (no está en cron).

6. Comentarios con Giscus

El archivo `_layouts/post.html` fue modificado para insertar el bloque de Giscus justo después de `{{ content }}`. Esto habilita comentarios en cada entrada del blog usando GitHub Discussions.

7. Dependencias y entorno

- Ubuntu 22.04 (Hetzner Cloud).
- Python 3.10 con entorno virtual.
- Playwright 1.44 con Chromium headless.
- openai, dotenv, google-api-python-client, trafilatura, markdownify, pyyaml.
- Variables sensibles como `OPENAI_API_KEY` y `YOUTUBE_API_KEY` están en `.env`.

8. Problemas actuales

- `bot.py` ya no detecta entradas nuevas porque no ejecuta correctamente `descubrir_articulos()`. Solo recorre los artículos de `FIXED_ARTICLES`.

- No hay control de errores si GPT falla.

- No hay validación cruzada entre `title_original` y contenido HTML (puede haber duplicados).

- Faltan logs estructurados para `bot.py`.

- No hay sistema de publicación manual o semiautomático para revisar contenido.

9. Información del repositorio

- Repositorio GitHub Pages basado en Jekyll.

- Usuario de git configurado en el servidor:

  user.name = HeraldoBot

  user.email = qwerteo@hotmail.com

- El push se hace automáticamente tras cada ejecución exitosa.

10. Pendiente de implementación

- Reparar la función de scraping automático de `bot.py`.

- Incluir pruebas y logs de errores para ambos bots.

- Incorporar validaciones de formato Markdown antes del commit.

- Añadir sistema de publicación manual para artículos en revisión.

- Validar duplicados por `original_url`.

11. GitHub Pages y despliegue

- El repositorio principal es `HeraldodelViejoMundo/Heraldo-del-Viejo-Mundo`.

- Está configurado para usar GitHub Pages con rama `gh-pages` como origen, carpeta `/`.

- Dominio personalizado: www.elheraldodelviejomundo.com (HTTPS activado).

- Último despliegue fue realizado por `github-pages[bot]` hace 11 horas.

12. Repositorio de publicación auxiliar

- Existe un repositorio adicional privado: `HeraldodelViejoMundo/oldworldpublish`.

- Su propósito parece ser generar y validar los archivos antes de moverlos al repositorio público.

- Último commit: mejora en lectura del YAML para detección de duplicados.

- Solo contiene `bot.py` y no tiene README aún.

13. Estructura real de archivos en producción

- `_posts/` contiene archivos Markdown generados automáticamente con fecha en nombre.

- `_layouts/` tiene `post.html` modificado para insertar Giscus tras `{{ content }}`.

- También existen: `Gemfile`, `Gemfile.lock`, `CNAME`, `.config.yml`, `.jekyll`, `README.md` y un archivo `Heraldo_Documentacion_Tecnica.txt`.

- En la raíz hay un `index.html` que ha sido restaurado tras errores 404.

Actualización: Archivos en /opt/oldworldbot/

Archivos principales:
- bot.py → Scrapea artículos de Warhammer Community. Usa GPT y Playwright. Actualmente necesita reparac
- video_bot.py → Funciona correctamente. Revisa canales de YouTube y genera posts en Markdown.
- run_bot.sh → Script de ejecución automática que mueve archivos y realiza commits.
- resolve_channel_ids.py → Script auxiliar relacionado con canales de YouTube.
- test_openai.py → Script de pruebas para llamadas a OpenAI.
- portada.html → HTML descargado/manual posiblemente usado en pruebas.

Carpeta posts/:
- Almacena temporalmente los .md antes de moverlos al repositorio Jekyll.

Carpeta logs/:
- video_bot.log, test_video_bot_manual.log → Logs del bot de vídeos.
- run_bot.log → Log general de ejecución automatizada.

Otros logs:
- salida.log, salida_debug.log, cron.log → Salidas diversas del sistema y del cron.

Carpeta venv/:
- Entorno virtual Python 3.10 con dependencias: playwright, trafilatura, markdownify, openai, etc.
- No debe editarse manualmente.

Actualización: Estructura de /opt/Heraldo-del-Viejo-Mundo/

Archivos principales:
- index.html → Página principal del blog.
- _config.yml → Configuración de Jekyll.
- CNAME → Dominios personalizados.
- README.md → Información general del proyecto.
- Heraldo_Documentacion_Tecnica.txt → Documento auxiliar con apuntes técnicos.

Carpetas importantes:
- _posts/: contiene los artículos en formato Markdown, generados automáticamente por los bots.
- _layouts/: incluye plantillas HTML. post.html ha sido modificado para incluir Giscus.
- assets/: contiene imágenes, CSS y recursos visuales como banner_final.png y estilos.
- scripts/: scripts de mantenimiento como fix_titles.py, video_bot.py o fetch.py.
- _site/ y _site_backup/: salidas generadas por Jekyll (producción y copia de seguridad).

Observaciones adicionales:
- El sistema de carpetas respeta la estructura estándar de Jekyll.
- Existen miniaturas, CSS personalizados y carpetas separadas para vídeos y torneos.
- El blog ya incluye múltiples artículos organizados por fecha y tipo de contenido.
- Se ha configurado para GitHub Pages con HTTPS y dominio personalizado.

Actualización: Carpeta /opt/pruebas-trafilatura/

Contenido detectado:
- test_trafilatura.py
- test_trafilatura.pyç (parece un duplicado mal nombrado, posiblemente accidental).

Propósito:
- Esta carpeta contiene scripts de prueba relacionados con la librería trafilatura.
- No forma parte del sistema de producción, pero puede ser útil para experimentación y debugging.

Sugerencia:
- Renombrar o eliminar el archivo mal escrito.
- Si los scripts son útiles, trasladar el contenido a una carpeta 'tests/' documentada o integrarlos como validado

Registro de cambios Git relevantes (12 de junio de 2025)

- Se ha eliminado manualmente el archivo: _posts/2025-06-02-yt-94V1j0sFbtc.md
- Se ha confirmado esta eliminación mediante 'git add -u' y commit correspondiente.
- Se ha subido y registrado en Git el archivo PDF actualizado:
  Manual_Heraldo_2025-06-12_final.pdf
- Posteriormente se ha hecho 'git pull --rebase' y 'git push origin main' sin conflictos.
- El PDF ha sido generado desde ChatGPT y documenta todo el sistema del blog a fecha 12 de junio.
- Este archivo está ahora incluido dentro del control de versiones del repositorio principal.