

Guía

💡 Guía Básica – De cero a Vaults

con Obsidian, Python... y V0ra

Setup rápido con vibes glitch kawaii Powered by V0ra



 python 3.9+  obsidian vaults  guide  glitchy



✧ Descripción

Un toolkit para convertir tus exportaciones de ChatGPT en **vaults de Obsidian**: ordenados, deduplicados, etiquetados y listos para explorar como fragmentos de memoria glitch kawaii.

Conoce a nuestros scripts-> [THE CREW](#)

✧ ¿Por qué MemoriA?

Si estás documentando conversaciones con LLMs de forma sistemática, ya sabrás que los exports de ChatGPT son...caóticos: JSON complejo, imágenes rotas, duplicados, bloques técnicos ilegibles.

MemoriA convierte ese caos en vaults de Obsidian navegables, permitiéndote analizar miles de conversaciones con la misma herramienta que usas para PKM. Construido por investigadora que gestiona 1GB+ de conversaciones documentadas.

Probado en producción.

Open source.

✧ Características

- 💫 **Split**: divide exportaciones en conversaciones individuales .md
- 🎨 **Glitch-safe dedupe**: evita duplicados con fingerprint SHA1
- 🌟 **Inserción de imágenes**: renderizado de imágenes en notas y Banco de imágenes en Obsidian.

- **Humanización de bloques Tool, Assistant, Canvas**
- **Vault-ready**: índices, subcarpetas por año/mes y tags
- **Reverse mode**: crea vaults espejo
- **Cute-core aesthetics**: todo con vibes kawaii y glitchy
- **Cerebrito mascota**: logo oficial incluido



0. Instalar lo imprescindible

1. Python 3.9 o superior

[Descargar aquí](#)

Durante la instalación: marca la casilla “**Add Python to PATH**”.

2. Obsidian

[Descargar aquí](#)

3. Dependencias para el importador

Abre una terminal (CMD/PowerShell en Windows, Terminal en macOS/Linux) y ejecuta:

```
pip install beautifulsoup4
```

(Opcional, pero recomendable):

```
pip install lxml
```



1. Exporta tus datos de ChatGPT

- En ChatGPT: Ajustes → Configuración de datos → Exportar .
- Recibirás en tu correo un enlace para descargar un ZIP (ej: `export-2025-09-07.zip`).



2. Descarga el pack MemoriA

- Archivo: `**MemoriA_V01.zip`
- Descomprime en una carpeta de trabajo (ej: `C:\MemoriAPack`).

Dentro tendrás:

- `batch_sequencer.py` → orquestador principal
- `split_chatgpt_export.py + vault_cleaner.py` → extracción y limpieza

- `extract_images_from_zips_dedup.py` + `ImageLinkInjector.py` → extracción a un banco de imágenes común e inyección de imágenes en las notas de Obsidian.
 - `RoleBlockExtractor.py`, `RenderTetherQuotes.py` → humanización de bloques "Tool"
 - `scaffolding_index.py`, `CleanImageToolBlocks.py` `tree_index.py`, `TidyBlankLines.py` → Indexado de ficheros auxiliares cargados, índices en modo árbol y limpieza básica.
 - Ejemplo de `gizmo_map.json` y `tag_map.json`
 - Plantilla de vault (`obsidian_vault_template/`)
 - Documentación del flujo completo paso a paso
 - Y un .bat para limpiar cualquiera de los VAULTS sin borrar las estructuras de carpetas, por si te lías en algún momento y quieras empezar de cero: `reset_vault.bat`.
-



3. Comienza el proceso

Abre una terminal en esa carpeta y ejecuta **paso por paso**, dependiendo de tus necesidades.

- Cada paso realiza un backup previo, pero puedes hacer **Pruebas seguras** → usando `--dry-run`
- Si tienes los chats agrupados por proyectos, usa `gizmo_map.json` para crear la tabla de asociación de **IDs** internos de ChatGPT a nombres de proyectos (ese grupo de caracteres alfanuméricos antes del guion en la URL: <https://chatgpt.com/g/> (g-p-6bd.....4a2- proyecto)). Esto permitirá incluir la información del proyecto en el yaml de la nota.

Comando extracción

- Paso 1: Extraer conversaciones a notas y crear vaults:

El programa te pedirá:

1. Ruta del/los ZIP exportados → arrástralos y ENTER.
2. Carpeta base para tus vaults → ej: `C:\Obsidian_Vaults` → arrástralos y ENTER.
3. Fecha que deseas usar en las propiedades de las notas.

```
python batch_sequencer.py "path_to_backups" "path_to_VAULT_BASE"
```

```
== ChatGPT → Obsidian (Batch Sequencer) ==

¿Quéquieres importar? (carpeta/lista, ENTER=carpeta):
Arrastra la carpeta con los exports y ENTER: "G:\GHU Codexsphere\Backups"
Carpeta base donde crear los Vaults: "G:\GHU Codexsphere\obsidian"
Fecha principal en YAML? (create/update, ENTER=create):
i Usando gizmo_map: G:\Descargas\chatgpt_community_pack_FULL V0.5\gizmo_map.json

▶ Preparando RAW_VAULT...

✓ Hecho.

RAW_VAULT      → G:\GHU Codexsphere\obsidian\RAW_VAULT
MERGED_VAULT   → G:\GHU Codexsphere\obsidian\MERGED_VAULT
REVERSE_VAULT  → G:\GHU Codexsphere\obsidian\REVERSE_VAULT
```

Paso 2: Extracción de imágenes:

Si quieras conservar tooodas las imágenes, incluso las duplicadas, usa `extract_images_from_zips.py`

El programa te pedirá:

1. Ruta del/los ZIP exportados → arrástralo y ENTER.
2. Carpeta base para tus imágenes → ej: 'C:\Obsidian_Vaults\IMAGE_BANK'. → arrástralo y ENTER.

```
python `extract_images_from_zips_dedup.py "path_to_backups" "path_to_IMAGE_BANK"
```

```
G:\GHU Codexsphere\memoria>extract_images_from_zips.py "G:\GHU Codexsphere\Backups"  
"G:\GHU Codexsphere\obsidian\IMAGE_BANK"
```

```
[G] Procesando 34 archivos ZIP...
```

```
✓2025-06-17-17-17-59-ebdfffb3cdcbe43d488cb9229ba4e2c93.zip: 91 imágenes extraídas.
```

Resumen final:

- Carpeta salida: G:\GHU Codexsphere\obsidian\IMAGE_BANK
- Imágenes únicas extraídas: 1124
- Duplicados omitidos: 4451
- Total único: 1124

Paso 3: Inserción de imágenes en las notas

El programa te pedirá:

1. Ruta del VAULT → arrástralo y ENTER.
2. Carpeta base de tus imágenes → ej: C:\Obsidian_Vaults\IMAGE_BANK . → arrástralo y ENTER.

```
python `ImageLinkInjector.py "path_to_Obsidian_Vault" "path_to_IMAGE_BANK"
```

```
G:\GHU Codexsphere\memoria>ImageLinkInjector.py "G:\GHU Codexsphere\obsidian\MERGED_  
VAULT" "G:\GHU Codexsphere\obsidian\IMAGE_BANK" --in-place
```

```
[■] Escaneando 349 notas Markdown...
```

```
✓Conversaciones\2025\04\2025-04-13_portada-musical-futurista.md: 7 enlace(s)  
⚠Conversaciones\2025\04\2025-04-13_portada-musical-futurista.md: 1 referencia(s) si  
n imagen  
✓Conversaciones\2025\04\2025-04-14_portada-moderna-y-épica.md: 25 enlace(s)
```

Resumen:

- Archivos con referencias: 162
- Enlaces creados: 846
- Referencias sin imagen: 1250

[1] Lee la nota al pie de página antes de volverte loco intentando entender por qué no renderiza las imágenes.

Escuadrón de limpieza:

Paso 4: Humanización de los bloques Tool en las notas

Te pedirán la ruta del VAULT a limpiar (RAW, MERGED, REVERSE).

Puedes ejecutarlos con `--dry-run` o `--in-place`

```
python `RoleBlockExtractor.py "path_to_Obsidian_Vault"
```

```
G:\GHU Codexsphere\memoria>python RoleBlockExtractor.py "G:\GHU Codexsphere\obsidian\MERGED_VAULT" --in-place  
Escaneando 349 archivos con .md...
```

```
✓G:\GHU Codexsphere\obsidian\MERGED_VAULT\Conversaciones\2024\09\2024-09-22_canciones-relajantes-en-español.md - 2 bloque(s) reescrito(s)
```

Resumen:

- Archivos con cambios: 258
- Bloques de rol reescritos: 3938

```
python `RenderTetherQuotes.py "path_to_Obsidian_Vault"
```

```
G:\GHU Codexsphere\memoria>rendertetherquotes.py "G:\GHU Codexsphere\obsidian\MERGED_VAULT" --in-place  
✓Conversaciones\2025\06\2025-06-26_el-rincón-de-lumen.md - 2 bloque(s) convertido(s)  
✓Conversaciones\2025\06\2025-06-29_qué-puedo-hacer-hoy.md - 1 bloque(s) convertido(s)  
✓Conversaciones\2025\11\2025-11-04_opinión-artística-y-estructural.md - 5 bloque(s) convertido(s)
```

Detectados: 1004 | Convertidos: 652

```
python `CleanImageToolBlocks.py "path_to_Obsidian_Vault"
```

```
G:\GHU Codexsphere\memoria>CleanImageToolBlocks.py "G:\GHU Codexsphere\obsidian\MERGED_VAULT" --in-place
```

⚠ Escaneando 349 notas Markdown...

```
✓Conversaciones\2025\04\2025-04-14_portada-moderna-y-épica.md - 2 bloque(s) eliminado(s)
```

Resumen:

- Bloques detectados: 3448
- Bloques eliminados: 86

```
python `TidyBlankLines.py "path_to_Obsidian_Vault"
```

```
G:\GHU Codexsphere\memoria>tidyblanklines.py "G:\GHU Codexsphere\obsidian\MERGED_VAULT" --in-place
```

```
✓Limpio: Guia.md  
✓Limpio: png.md  
✓Limpio: Switchboard.md  
✓Limpio: _tree_index.md  
✓Limpio: Conversaciones\2025\04\2025-04-13_portada-musical-futurista.md  
✓Limpio: Conversaciones\2024\11\2024-11-23_fondo-minecraft-con-tótem.md
```

Archivos ajustados: 200

Cartógrafos de élite

- Paso 5: Creación de índices y limpieza básica de formato:
Te pedirán la ruta del VAULT (RAW, MERGED, REVERSE).

```
python `scaffolding_index.py "path_to_Obsidian_Vault"
```

```
G:\GHU Codexsphere\memorIA>"G:\GHU Codexsphere\memorIA\scaffolding_index.py" "G:\GHU Codexsphere\obsidian\REVERSE_VAULT"  
 Índice generado: G:\GHU Codexsphere\obsidian\REVERSE_VAULT\scaffolding_index.md  
Andamiajes detectados: 153  
Conversaciones indexadas: 6746
```

```
python `tree_index.py "path_to_Obsidian_Vault"
```

```
G:\GHU Codexsphere\memorIA>python tree_index.py "G:\GHU Codexsphere\obsidian\MERGED_VAULT"  
 Índice generado: G:\GHU Codexsphere\obsidian\MERGED_VAULT\_tree_index.md  
Proyectos: 14 · Notas: 343
```



4. Resultado

En la carpeta base se crean tres vaults:

- **RAW_VAULT** → sin deduplicado, todas las versiones numeradas.
- **MERGED_VAULT** → deduplicado + combinado de versiones, orden cronológico normal.
- **REVERSE_VAULT** → deduplicado + combinado de versiones, orden cronológico inverso por bloques.
- Se crea una banco de imágenes común y navegable desde Obsidian
- Las notas renderizan las imágenes
- Los bloques en las Tool son legibles
- Contamos con un Índice navegable de conversaciones por Proyecto (tree_index)
- Contamos con un Índice navegable de conversaciones por ficheros cargados al inicio (scaffolding_index)

Cada uno incluye una estructura mínima de carpetas: img, tags, Conversaciones, Plantillas.



5. Abrir en Obsidian

1. Abre Obsidian.
2. Open another vault → Open folder as vault .
3. Añade RAW_VAULT , MERGED_VAULT , REVERSE_VAULT .

4. Usa el menú inferior para cambiar de uno a otro.

■■■■■ READY TO VAULT *■■■■■*

■■■ Roadmap■■■

¿Qué me gustaría implementar a continuación?

- Documentación exhaustiva de parámetros.
 - Traducción a otros idiomas.
 - Etiquetado semántico por temática.
 - Definir un archivo de configuración central para facilitar el uso:** config.yaml con rutas y flags por defecto.
 - Importación incremental:** detectar nuevos exports y procesar solo novedades.
 - Super lanzador con pasos:** --stage {extract,images,link,tidy,tags,all} .
 - GUI ligera:** asistente local para arrastrar ZIPs y ver progreso.
-

1. Uso un banco de imágenes común para todos los Vault; para que Obsidian sea capaz de renderizar las imágenes desde una carpeta externa a su estructura es necesario crear un vínculo simbólico usando:

```
cmd /c mklink /D "G:\Obsidian_Vault\REVERSE_VAULT\IMAGE_BANK" "G:\Obsidian\IMAGE_BANK" ↵
```