**Investigación de Herramientas para Cifrado de Archivos y Carpetas**

**Objetivo de la Investigación**

Analizar y justificar el uso de herramientas de cifrado de archivos para carpetas locales, enfocadas en criptografía simétrica realizable en Python, con énfasis en su implementación práctica.

**Herramientas Investigadas**

1. **Fernet** (biblioteca **cryptography**) - Fernet es un sistema de cifrado simétrico que utiliza AES‑128‑CBC + HMAC‑SHA‑256, con claves de 32 bytes, IV aleatorio y verificación de integridad.
   1. **Ventajas**:
      1. - Asegura confidencialidad, integridad y autenticidad con una API sencilla.
      2. - Ideal para cifrar bytes de archivos completos.
   2. **Justificación**: Combina robustez (AES+CBC+HMAC) con facilidad de uso en Python, apropiado para proteger información local en contextos académicos y profesionales.
2. **PBKDF2** (derivación de clave por contraseña) - **PBKDF2** es una función estándar (RFC 2898, PKCS #5 v2.0) que transforma contraseña + sal + múltiples iteraciones en una clave simétrica segura.
   1. **Ventajas**:
      1. - Añade una capa de seguridad al convertir contraseñas en claves robustas.
      2. - Ideal para usar con Fernet en escenarios donde la clave deriva de una contraseña de usuario.
   2. **Justificación**: Fortalece sistemas basados en password, evitando ataques de fuerza bruta.
   3. Video: <https://www.youtube.com/watch?v=fKw4L-mmv0c> [PBKDF2 tutorial in Python – Python Basics](https://www.youtube.com/watch?v=fKw4L-mmv0c)
3. **Tkinter** - GUI estándar de Python, multiplataforma, ideal para prototipos rápidos.
   1. **Usos en proyecto:** 
      1. - Ventana principal
      2. Selección de archivos (filedialog)
      3. Entradas de clave y botones de acción
      4. Mensajes emergentes con messagebox
   2. **Justificación**: No requiere dependencia externa, permite construir interfaces funcionales sin complicaciones.
   3. **Video**: <https://www.youtube.com/watch?v=ouO_QRJtvNE> - [Open Files Dialog Box – Python Tkinter GUI Tutorial #15](https://www.youtube.com/watch?v=ouO_QRJtvNE)
4. **PyInstaller** Genera ejecutables (e.g. .exe) de scripts Python.
   1. **Ventajas**:
      1. Facilita distribución sin exigir Python en el sistema.
      2. Compatible con GUIs como Tkinter.
   2. **Justificación**: Ideal para entregar aplicaciones a usuarios finales sin entornos de desarrollo.
5. **logging (Python estándar) -** Biblioteca para registrar eventos: errores, advertencias, información.
   1. **Implementación**: Errores durante cifrado/descifrado se guardan en logs/error.log
   2. **Justificación**: Mejora trazabilidad de errores y soporte técnico, manteniendo calidad y profesionalismo en el sistema.

**Comparativa de herramientas**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Herramienta | Cifrado seguro | Interfaz | Portable | Requiere instalación | Programable |
| Python + Fernet | ✅ AES‑128/CBC | ✅ GUI | ✅ .exe | ❌ Python o exe | ✅ |
| VeraCrypt | ✅ Muy fuerte | ✅ GUI | ❌ | ✅ Instalación | ❌ |
| 7‑Zip + AES | ✅ Medio | ❌ CLI | ✅ | ✅ Instalación | ❌ |
| WinRAR con clave | ❌ Menos fuerte | ✅ GUI | ✅ | ✅ Instalación | ❌ |

**Conclusión**

La solución combinada:

1. **Fernet + PBKDF2:** cifrado fuerte y seguridad de clave
2. **Tkinter**: interfaz accesible sin dependencias externas
3. **PyInstaller**: ejecutable para usuarios finales
4. **logging**: calidad profesional y seguimiento de errores es una plataforma robusta, portable y adecuada para uso académico, personal y empresarial.