

Guía de Instalación - APEDIF IA

Análisis de Evidencia Digital Forense con Inteligencia Artificial

Versión: 1.0

Autor: René Alejandro Parra Almirón

Fecha: Septiembre 2025

📋 Tabla de Contenidos

1. [Requisitos del Sistema](#)
 2. [Instalación de Dependencias](#)
 3. [Configuración del Entorno](#)
 4. [Instalación de APEDIF IA](#)
 5. [Verificación de la Instalación](#)
 6. [Primer Uso](#)
 7. [Solución de Problemas](#)
 8. [Configuración Avanzada](#)
-

💻 Requisitos del Sistema

Requisitos Mínimos

- **Sistema Operativo:** Windows 10/11 (64-bit)
- **Procesador:** Intel i3 o AMD equivalente (4 núcleos)
- **Memoria RAM:** 8 GB mínimo
- **Almacenamiento:** 10 GB libres
- **Conexión a Internet:** Para descarga inicial de modelos

Requisitos Recomendados

- **Procesador:** Intel i5-12400F o superior (6+ núcleos)
- **Memoria RAM:** 16-32 GB
- **GPU:** NVIDIA RTX 3070 8GB o superior (recomendado para IA)
- **Almacenamiento:** SSD con 20+ GB libres

Configuración Utilizada en Desarrollo

El prototipo APEDIF IA fue desarrollado y probado en:

- **CPU:** Intel i5-12400F (6 núcleos, 12 hilos)
- **RAM:** 32 GB DDR4
- **GPU:** NVIDIA RTX 3070 (8 GB VRAM)
- **Almacenamiento:** SSD NVMe 1 TB

Nota sobre GPU: Las tarjetas RTX (3070, 3080, 4070, etc.) ofrecen ventajas significativas para modelos de IA debido a:

- **Tensor Cores:** Aceleración específica para operaciones de machine learning
- **VRAM abundante:** 8+ GB permite cargar modelos completos en memoria GPU
- **Arquitectura CUDA:** Optimizada para frameworks de IA como PyTorch
- **NVENC/NVDEC:** Codificadores de hardware para procesamiento multimedia

Instalación de Dependencias

Paso 1: Instalar Python 3.11

1. Descargar desde [python.org](https://www.python.org)
2. Ejecutar el instalador y marcar "Add Python to PATH"
3. Verificar instalación:

```
cmd  
python --version  
pip --version
```

Paso 2: Instalar Git

Descargar desde git-scm.com y ejecutar el instalador.

Paso 3: Instalar Ollama (Obligatorio para IA)

1. Descargar desde ollama.ai
2. Ejecutar el instalador
3. Reiniciar el sistema
4. Verificar instalación:

```
cmd
```

```
ollama --version
```

⚙️ Configuración del Entorno

Paso 1: Crear Estructura de Directorios

```
cmd  
  
mkdir C:\ForensicAI  
mkdir C:\ForensicAI\logs  
mkdir C:\ForensicAI\reportes  
mkdir C:\ForensicAI\casos_reales\casos_procesados  
mkdir C:\ForensicAI\casos_reales\imagenes_forenses
```

Paso 2: Descargar Modelo de IA

```
bash  
  
ollama pull llama3.1:8b
```

Nota: Esta descarga puede tomar 10-30 minutos dependiendo de la conexión.

Paso 3: Crear Entorno Virtual de Python

Navegar al directorio del proyecto:

```
cmd  
  
cd C:\ForensicAI
```

Crear entorno virtual:

```
cmd  
  
python -m venv apedif_env
```

Activar entorno virtual:

```
cmd  
  
apedif_env\Scripts\activate
```

Instalación de APEDIF IA

Paso 1: Descargar Archivos del Proyecto

Coloca los siguientes archivos en `C:\ForensicAI\`:

- `forensic_data_generator.py`
- `forensic_analyzer.py`
- `ai_forensic_assistant.py`
- `forensic_gui_app.py`

Paso 2: Instalar Dependencias de Python

Con el entorno virtual activado, ejecuta:

```
cmd  
  
pip install --upgrade pip  
pip install customtkinter==5.2.0  
pip install matplotlib==3.7.1  
pip install pandas==2.0.3  
pip install numpy==1.24.3  
pip install Pillow==10.0.0  
pip install seaborn==0.12.2
```

Paso 3: Verificar Estructura Final

Tu directorio debe verse así:

```
C:\ForensicAI\  
├── apedif_env\          # Entorno virtual  
├── logs\                # Logs del sistema  
├── reportes\            # Reportes generados  
├── casos_reales\  
│   ├── casos_procesados\ # Casos JSON  
│   └── imagenes_forenses\ # Imágenes forenses  
├── forensic_data_generator.py # Generador de datos  
├── forensic_analyzer.py    # Motor de análisis  
├── ai_forensic_assistant.py # Asistente IA  
└── forensic_gui_app.py    # Interfaz gráfica
```

Verificación de la Instalación

Paso 1: Verificar Ollama y Modelo

```
cmd
```

```
ollama list
```

Debe mostrar `llama3.1:8b` en la lista.

Paso 2: Verificar Python y Dependencias

```
cmd
```

```
python -c "import customtkinter, matplotlib, pandas, numpy; print('Todas las dependencias instaladas correctamente')"
```

Paso 3: Prueba Básica del Sistema

Activar entorno:

```
cmd
```

```
C:\ForensicAI\apedif_env\Scripts\activate
```

Navegar al directorio:

```
cmd
```

```
cd C:\ForensicAI
```

Probar generador de datos:

```
cmd
```

```
python forensic_data_generator.py
```

Probar analizador:

```
cmd
```

```
python forensic_analyzer.py
```

Probar asistente IA:

```
cmd
```

```
python ai_forensic_assistant.py
```

Primer Uso

Ejecutar APEDIF IA

1. Activar entorno virtual:

```
cmd  
C:\ForensicAI\apedif_env\Scripts\activate
```

2. Ejecutar interfaz gráfica:

```
cmd  
cd C:\ForensicAI  
python forensic_gui_app.py
```

3. Primer análisis:

- La aplicación se abrirá con un ID de caso generado automáticamente
- Selecciona tipo de caso: "employee_data_theft"
- Mantén "Sintético" seleccionado como tipo de evidencia
- Haz clic en " Análisis" para generar y analizar un caso
- Una vez completado, haz clic en " IA" para análisis inteligente
- Finalmente, " Reporte" generará reportes completos

Solución de Problemas

Error: "Ollama no encontrado"

Síntomas: El sistema reporta que la IA no está disponible.

Solución:

1. Verificar que Ollama esté instalado:

```
cmd  
ollama --version
```

2. Si no está instalado, repetir instalación de Ollama

3. Asegurar que el modelo esté descargado:

```
cmd
```

```
ollama pull llama3.1:8b
```

Error: "ModuleNotFoundError"

Síntomas: Python no encuentra módulos como customtkinter.

Solución:

1. Verificar que el entorno virtual esté activado
2. Reinstalar dependencias:

```
cmd
```

```
pip install --force-reinstall customtkinter matplotlib pandas numpy Pillow seaborn
```

Interfaz gráfica no se abre

Síntomas: Error al ejecutar forensic_gui_app.py

Solución:

1. Verificar que no hay problemas con el sistema de ventanas de Windows
2. Reinstalar dependencias gráficas:

```
cmd
```

```
pip install --force-reinstall customtkinter Pillow
```

Rendimiento lento de IA

Síntomas: El análisis IA toma más de 3-5 minutos.

Solución:

1. Con GPU NVIDIA RTX (Recomendado):

- Las RTX 3070/3080/4070 ofrecen el mejor rendimiento para Llama 3.1
- Verificar CUDA disponible:

```
cmd
```

```
nvidia-smi
```

- Instalar PyTorch con soporte CUDA (opcional para Ollama):

```
cmd
```

```
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
```

2. Con GPU NVIDIA GTX (Limitado):

- GTX 1660/1070/1080 funcionan pero con menor velocidad
- Considerar usar modelo más pequeño si es muy lento

3. Solo CPU (Lento pero funcional):

- El rendimiento lento es normal en CPU
- Para mejor experiencia, usar modelo más pequeño:

```
cmd
```

```
ollama pull llama3.1:7b
```

Tiempos esperados de respuesta IA:

- **RTX 3070:** 30-75 segundos por análisis
- **GTX 1660:** 60-120 segundos por análisis
- **CPU moderno:** 120-300 segundos por análisis

Errores de memoria insuficiente

Síntomas: "Out of memory" durante análisis de IA.

Solución:

1. Cerrar otras aplicaciones
2. Usar modelo más pequeño
3. Reducir tamaño de casos sintéticos en el código

Configuración Avanzada

Cambiar Directorio Base

Editar en todos los archivos Python la variable:

```
python  
  
# Cambiar de:  
base_dir = "C:/ForensicAI"  
# A:  
base_dir = "/tu/nuevo/directorio"
```

Configurar Modelo de IA Alternativo

En `ai_forensic_assistant.py`, línea ~45:

```
python  
  
# Cambiar modelo por defecto  
def __init__(self, model_name: str = "llama3.1:7b"): # Modelo más pequeño
```

Ajustar Niveles de Logging

En cada archivo Python, modificar:

```
python  
  
logging.basicConfig(  
    level=logging.DEBUG, # Más detallado: DEBUG, INFO, WARNING, ERROR  
    # ...  
)
```

Personalizar Tipos de Casos

En `forensic_data_generator.py`, línea ~35:

```
python  
  
self.case_types = [  
    "employee_data_theft",  
    "intellectual_property_theft",  
    "malware_infection",  
    "tu_nuevo_tipo_caso" # Agregar tipos personalizados  
]
```

Archivos de Configuración Adicionales

requirements.txt (Opcional)

Crear archivo `requirements.txt` con:

```
customtkinter==5.2.0  
matplotlib==3.7.1  
pandas==2.0.3  
numpy==1.24.3  
Pillow==10.0.0  
seaborn==0.12.2
```

Instalar con:

```
cmd
```

```
pip install -r requirements.txt
```

Script de Inicio Rápido

Crear `start_apedif.bat` en `C:\ForensicAI\`:

```
batch
```

```
@echo off  
cd /d C:\ForensicAI  
call apedif_env\Scripts\activate  
python forensic_gui_app.py  
pause
```

Uso del script:

1. Guardar el contenido anterior como `start_apedif.bat` en `C:\ForensicAI\`
2. Hacer doble clic en el archivo para ejecutar APEDIF IA directamente
3. El script activará automáticamente el entorno virtual y lanzará la aplicación

sos Soporte y Documentación

Logs del Sistema

Revisar logs en caso de errores:

```
C:\ForensicAI\logs\  
├── gui_app.log      # Log de la interfaz  
├── analyzer.log     # Log del analizador  
├── ai_assistant.log # Log del asistente IA  
└── data_generator.log # Log del generador
```

Información del Sistema

Para reportar problemas, incluir:

```
cmd
```

```
python --version  
ollama --version  
pip show customtkinter matplotlib pandas numpy seaborn Pillow
```

Contacto del Desarrollador

- **Autor:** René Alejandro Parra Almirón
 - **Institución:** Universidad Católica de Salta
 - **Proyecto:** Trabajo Final de Licenciatura en Ciencia de Datos
-

⚠ Notas Importantes

1. **Uso Académico:** APEDIF IA es un prototipo educativo, no validado para uso en investigaciones reales.
 2. **Datos Sintéticos:** El sistema está diseñado para trabajar con datos sintéticos seguros.
 3. **Privacidad:** Todo procesamiento es local, no se envían datos a servicios externos.
 4. **Licencia:** Código abierto para uso académico y educativo.
 5. **Actualizaciones:** Verificar periódicamente actualizaciones de Ollama y dependencias.
-

✓ Checklist Final de Instalación

- Python 3.11 instalado y verificado
- Git instalado (si es necesario)
- Ollama instalado y funcionando
- Modelo llama3.1:8b descargado
- Estructura de directorios creada
- Entorno virtual creado y activado
- Dependencias Python instaladas
- Archivos del proyecto copiados
- Prueba básica del generador exitosa
- Prueba básica del analizador exitosa
- Prueba básica del asistente IA exitosa
- Interfaz gráfica abre correctamente
- Primer análisis completo realizado

¡Instalación completa! APEDIF IA está listo para uso académico y demostraciones.