1. Extracción de los correos y whatsapps (Cabeceras, bases de datos…).

2. Validación de los datos. Adveración probatoria.

3. Detección de manipulación. (generación automática, edición de bases de datos, correlación entre el backup local y la nube, cambios de estilo en la escritura (stylometry), anomalías temporales, determinar si son consistentes los archivos multimedia)

4. Garantía de integridad (sello de integridad).

**Blackboard:** es una memoria compartida o una base de datos temporal que permite a los diferentes módulos del software de análisis forense digital comunicarse entre sí y almacenar los resultados de sus análisis

**DKIM:** mecanismo de autenticación de correo electrónico fundamental para contrarrestar la suplantación de identidad y el phishing.

¿Cómo me imagino que es el trabajo?

* Se han presentado unas pruebas en WhatsApp o correo.
* El juez ha solicitado una validación a un perito informático.
* Asumo que ya tengo los datos de la conversación en un archivo que pueda leer.
* La idea es verificar que esas pruebas no han sido manipuladas y que son verídicas.
* Generar un informe con sello de integridad.

1. Preparación de los datos   
   Extraer las cabeceras y contenido de los emails. Investigar el formato de los WhatsApps y coger las imágenes también.
2. Primera validación  
   Emails: DKIM, cadena de mensajes, congruencia de fechas.  
   WhatsApp: timeline, copias de la nube y el backup local, metadatos congruentes.
3. Detección de manipulación  
   - Tiempo: mirar si hay algún gap en el tiempo que no cuadre, saltos dispersos o mensajes demasiado bien enviados (signo de automatización)  
   - Edición de bases de datos: en el whatsapp existen unas bases de datos que se podrían editar al sacar la evidencia del chat.  
   - Stylometry: comprobar que no se ha hecho con una IA, cambios de estilos de letra, o de estilo escribiendo (formal,informal).  
   - Media: revisar que el EXIF es congruente, thumbnails desincronizados o fotos que no pertenecen al contexto de la conversación.
4. Sello

Generar un informe en Autopsy con una firma o certificado.

Intuyo que para hacer esto necesitaré una base de datos con correos y conversaciones de whatsapps manipulados y no manipulados para que se puedan entrenar los modelos de detección. O investigar si esto ya existe e implementarlo en Autopsy o si ya está creado, pero no en Autopsy.

Hacerlo en Python

Determinar el scope de si meter Android y iOS, por saber si me daría tiempo.

Notas:  
determinar que formato de mail me tienen que pasar el mail: pst, eml 🡪 El formato de correo ideal sería el **EML** (Electronic Mail Format), porque conserva íntegramente el mensaje con sus cabeceras, cuerpo y adjuntos en un archivo plano. Esto facilita la **extracción de metadatos** como remitente, destinatario, fechas, IPs y rutas de servidores SMTP. Además, EML es un estándar ampliamente soportado por clientes de correo y librerías forenses en Python.

Separar muy bien entre el módulo de Whatsapp y email.

A mi programa le debería dar igual si es de iOS o Android.   
Mirar que verificación tiene Whatsapp.

TAREAS:

Hello World de Autopsy con mi módulo de Python. HECHO

Investigar de Whatsapp que contenido de la base de datos que me podría servir para decir que esto es autentico y no manipulado.

#semana 1

Investigar cómo usar Autopsy, cómo cargar módulos, cómo usar Python en Autopsy (Jython).

Instalar Autopsy 4.22.1\_v2 en la VM y sleuthkit-java\_4.14.0-1\_amd64. He tenido que instalar también JDK11.

Crear jython de helloWorld

Crear repositorio de Github.

EN LINUX

1. Para lanzar Autopsy:
   1. cd Downloads/﻿autopsy-4.22.1
   2. bash ./bin/autopsy
2. Los módulos los metemos en ﻿/home/kali/.autopsy/dev/python\_modules/

EN WINDOWS

1. C:\Users\usuario\AppData\Roaming\autopsy\python\_modules

#semana 2

Created two modules showing that “hello world” shows a message and other “hello world artifact” that creates artifacts for each file that is exploring.

#semana 3

Email module:

<https://sleuthkit.org/autopsy/docs/user-docs/4.19.3/email_parser_page.html>

<https://peritoinformaticocolegiado.es/blog/peritaje-informatico-forense-con-autopsy/>

Otros enlaces + docs desacargados:

<https://www.camerfirma.com/como-enviar-whatsapp-certificado/>

<https://alejandria.poligran.edu.co/handle/10823/2136>

<https://itscience-indexing.com/jurnal/index.php/CNAPC/article/view/2286>

Mi modulo del email va a calcular un hash del contenido del mail. Porque lo que van a hacer es que si se presenta otro correo poder comparar los hashes. Si cambia por una coma el hash cambia. De momento esto solo sirve para comprobar que no cambia. O para comprobar con un backup que tenga.

No sirve si me dan un email ya manipulado.

Registro de email como evidencia:

[ <https://www.peritos-judiciales-perytas.com/blog/autenticidad-correo-electronico/> ]

- Preservar la cadena de custodia

- Crear una copia del correo original para trabajar sobre ella, para evitar posibles alteraciones durante el peritaje

- Nada mas descargar la evidencia sacar el hash tanto del correo como de los adjuntos.

- Analizar las cabeceras y los metadatos.

- Verificación de mecanismos de autenticación de correo (SPF, DKIM, DMARC) para comprobar si el correo fue realmente emitido desde el dominio declarado o si podría haber sido falsificado.

- Comparación con registros DNS / WHOIS / bases de datos públicas para ver si los dominios implicados están correctamente configurados.

Con esta información, mi script tiene que:  
1. Hacer el hash del mail y de sus adjuntos. Aparecerá en Analysis Results.

2. Sacar las cabeceras del mail y submails que pueda contener. Aparecerá en Data Artifacts.

Special attention to:

* Date
* From
* To
* Subject
* Message-ID
* Received
* Return path
* DKIM signature
* validaciones SPF, DKIM y DMARC
* Cabeceras MIME (de momento NO)

1 y 2 sirven para sacar el hash y ordenar la información de las cabeceras. Ahora en los siguientes pasos tenemos que sacar las evidencias.

#semana 4

He hecho el modulo de cabeceras inconsistentes (explicado en el Word correspondiente)  
He hecho el modulo de date coherence (explicado en el Word correspondiente)

## Prueba de No Manipulación de Cabeceras Básicas

Probar la "no manipulación" de cabeceras básicas (To, From, Subject, etc.) es un concepto más sutil y se centra en la **Integridad Criptográfica** del correo.

Es **imposible** que un módulo de ingesta de Autopsy (sin claves privadas) *pruebe* la no manipulación del cuerpo del correo o de las cabeceras básicas (que no están protegidas por Received:), ya que cualquier atacante puede cambiar la cabecera Subject: y recalcular el *hash* del archivo.

Sin embargo, tu módulo puede detectar la **ruptura de la integridad criptográfica** que prueba la manipulación del *contenido*, lo que automáticamente prueba que las cabeceras básicas NO pueden ser de confianza.

### El Enfoque Criptográfico (DKIM y ARC)

El único mecanismo que prueba la integridad de las cabeceras básicas y el cuerpo del mensaje es **DKIM** (DomainKeys Identified Mail) y **ARC** (Authenticated Received Chain).

Tu módulo **no necesita validar las firmas (DKIM/ARC)** (eso requeriría resolver DNS para obtener claves públicas), solo necesita **detectar cuándo fallan**.

#### Mecanismo para probar la no manipulación:

1. **Detección de Fallo DKIM:**
   * **Original:** El extracto muestra dkim=pass en la cabecera Authentication-Results.
   * **Manipulación:** Cambia ligeramente el Subject: (e.g., de "Prueba para TFM" a "Prueba FINAL para TFM").
   * **Resultado Esperado:** Cuando el correo llega a un servidor validador (como Gmail), este recalcula la firma DKIM. Como el Subject: ha cambiado, la firma **fallará**. El servidor receptor (Gmail/Outlook) marcará en la cabecera **Authentication-Results** que dkim=fail.
2. **Detección de Fallo ARC:**
   * ARC es una cadena de sellos (ARC-Seal) que protege las cabeceras modificables (como Received:) a través de diferentes servidores. Si una cabecera Received: es alterada por un intermediario malicioso, ARC lo detectará.
   * La mejor prueba de manipulación es el **link\_break** que ya creamos en la sección 1.A, ya que una alteración tan flagrante debería hacer que el sello ARC se invalidara.

### Conclusión para tu Módulo

Para tu TFM, el mejor modo de "probar la no manipulación" (o, más bien, **demostrar la integridad**) de las cabeceras es:

1. **Módulo de Headers/Coherencia:** Confirma que **no hay reverse\_order o link\_breaks** en los Received:.
2. **Módulo de Basic Headers:** Añade al reporte un *checkbox* que indique el **estado de DKIM** y **ARC** (pass, fail, o none) de la cabecera Authentication-Results más reciente.

Si el correo pasa estas dos pruebas (coherencia temporal y DKIM/ARC *pass*), se considera **auténtico** con alta probabilidad.

#semana 5

He hecho el módulo de message-ID anómalos (explicado en el Word correspondiente)

He hecho el módulo de X-headers (explicado en el Word correspondiente)

He empezado el módulo de DKIM/SPF/ARC

base = os.path.dirname(os.path.abspath(\_\_file\_\_)) # core dir

* \_\_file\_\_ es una variable especial que contiene la ruta del script de Python que se está ejecutando actualmente.
* os.path.abspath(\_\_file\_\_) obtiene la **ruta absoluta** de ese script.
* os.path.dirname(...) extrae el **directorio** donde reside ese script.
* **Resultado:** base contendrá la ruta del directorio "principal" desde donde se inicia la búsqueda.

Instalando las librerías de python3:  
He creado una virtual environment en la carpeta asi en powershell:

& "C:\Users\usuario\AppData\Local\Microsoft\WindowsApps\python.exe" -m venv "$env:APPDATA\autopsy\python\_modules\venv\_autopsy"  
"$env:APPDATA\autopsy\python\_modules\venv\_autopsy\Scripts\Activate.ps1"

Si me da error pongo esto:

Set-ExecutionPolicy -Scope Process -ExecutionPolicy RemoteSigned  
he instalado estas librerías:

pip install dkimpy dnspython cryptography