Trabajo Final de Módulo: Construcción de un lakehouse.

Para esta práctica se debe diseñar un lakehouse en local.

Para ello, tendremos en cuenta lo siguiente:

- Tendremos tres ficheros:
  - "logs_web.csv" → El cual debemos ingestar como un fichero CSV.
  - "users.json" → Debemos ingestarlo en la colección "users" de MongoDB antes de comenzar a trabajar con ello.
  - "ip_reputation.json" → Debemos ingestarlo en la colección "ip_reputation" de MongoDB antes de comenzar a trabajar con ello.
- Debemos seguir una arquitectura medallion (bronze, silver y gold) que construiremos sobre ClickHouse.
- Todos los procesos de lectura de ficheros y MongoDB, procesamiento de los datos e ingestas en ClickHouse debemos gestionarlos con Python en local.

Se debe realizar el código para desarrollar la siguiente solución y una memoria que abarque lo siguiente:

- Explicar solución desarrollada.
- Explicar algunas opciones de mejora sobre la solución planteada.
- Elegir un cloud y, con las herramientas disponibles dentro de dicho cloud, plantear una solución de construcción de un lakehouse y sus procesos (automatización, procesamiento de datos, almacenamiento de los ficheros, gestión de tablas, herramientas de visualización, herramientas de ML, etc). Se debe explicar por qué se eligen dichas tecnologías.

Se deberá realizar una presentación del material desarrollado, para ello, se puede preparar una presentación Power Point que explique:

- Solución desarrollada.
- Opciones de mejora.
- Solución cloud propuesta y por qué.

NOTA: Esta práctica se realiza **en grupos ya establecidos en el aula virtual.**

NOTA: El fichero a entregar será un ZIP que contendrá lo siguiente:

- La memoria solicitada.
- El código desarrollado.
- La presentación Power Point.

Dicho ZIP debe tener el siguiente nombre "***GRUPO_[NUM_DE_GRUPO]-TFModulo.zip***"

Anexos

Columnas fichero “logs_web.csv”

event_id

- Identificador único del evento de log.
- Tipo: string (evt_001, evt_002…).
- Sirve como clave técnica para rastrear o depurar un request concreto.

event_ts

- Marca de tiempo en formato ISO8601 (2025-06-01T09:05:00Z).
- Tipo: datetime.
- Indica cuándo ocurrió la petición.

user_id

- Identificador del usuario autenticado que hace la petición (usr_001, usr_002…), alineado con _id de users en Mongo.
- Puede venir vacío ("") si el usuario es anónimo.
- Tipo: string.

ip_address

- IP desde la que se realiza la petición, por ejemplo 203.0.113.45.
- Tipo: string.
- Se cruza con la colección ip_reputation para añadir info de riesgo de la IP.

http_method

- Tipo de método HTTP: GET, POST, etc.
- Tipo: string.
- Útil para distinguir lecturas (GET) de operaciones de login/checkout (POST).

url_path

- Ruta de la URL (sin dominio), por ejemplo /login, /cart, /products/123.
- Tipo: string.
- Permite derivar casos de uso: login, navegación de catálogo, checkout, sección admin, APIs, etc.

status_code

- Código HTTP devuelto por el servidor:
  - 2xx → éxito (200 OK)
  - 3xx → redirección (302…)
  - 4xx → error cliente (401, 403…)
  - 5xx → error servidor (500…)
- Tipo: entero.
- Permite distinguir login fallido/exitoso, errores del sistema, etc.

bytes_sent

- Tamaño de la respuesta en bytes enviada al cliente.

- Tipo: entero.
- Puedes usarlo para analizar volumen de datos servidos, posibles anomalías (respuestas muy grandes, etc.).

response_time_ms

- Tiempo de respuesta de la petición en milisegundos.
- Tipo: entero.
- Es clave para métricas de rendimiento (latencia media, p95, etc.).

user_agent

- Cadena que identifica el cliente: navegador, SO o script (Mozilla_Windows, curl_7_68_0...).
- Tipo: string.
- Útil para detectar automatismos (bots/scripts) frente a navegadores normales.

is_suspicious

- Flag de sospecha sobre el evento: 1 = sospechoso, 0 = normal.
- Tipo: entero (puede mapearse a boolean).
- En la práctica, podéis recalcularlo en Silver/Gold combinando status, IP, URL, etc.

Columnas fichero “users.json”

_id

- Identificador único del usuario (usr_001...), que coincide con user_id en los logs.
- Tipo: string.
- Es la clave de unión principal entre logs y perfil de usuario.

username

- Nombre de usuario (login) dentro de la plataforma (alice, bob...).
- Tipo: string.

email

- Correo de contacto del usuario (name@example.com).
- Tipo: string.

role

- Rol de negocio o permisos del usuario: standard, analyst, admin...
- Tipo: string.
- Sirve para segmentar comportamiento por tipo de usuario (cliente normal, admin, etc.).

country

- País de origen principal del usuario (código ISO 2 letras: ES, IT, FR, PT...).
- Tipo: string.
- Permite análisis geográficos y reglas de seguridad/privacidad por país.

created_at

- Fecha de alta del usuario en la plataforma, formato ISO8601.
- Tipo: datetime.

is_premium

- Indica si el usuario tiene un plan de pago / premium.
- Tipo: boolean.
- Se puede usar en las tablas Gold para ver si las amenazas se concentran más en clientes premium, etc.

risk_score

- Puntuación de riesgo asociada al usuario (0–1, por ejemplo).
- Tipo: número (float).
- Podéis interpretarlo como probabilidad / índice de riesgo, similar a scores de reputación o fraude.

Columnas fichero "ip_reputation.json"

_id

- Identificador interno del registro de reputación (ip_001, ip_002...).
- Tipo: string.
- No es clave de unión; lo importante para joins es ip.

ip

- Dirección IP a la que se le asigna una reputación (203.0.113.45, 185.199.110.153...).
- Tipo: string.
- Se unirá con ip_address en logs_web.csv.

source

- Origen de la información de reputación:
  - threat_intel → feeds externos de threat intelligence
  - internal_allow_list → lista interna de IPs de confianza
  - internal_network → red corporativa interna
  - external_feed, internal_list, etc.
- Tipo: string.
- En el mundo real, muchas soluciones de seguridad etiquetan fuentes de reputación de manera parecida.Thales Documentation Portal+1

risk_level

- Nivel de riesgo de la IP:
  - low, medium, high, critical...
- Tipo: string (podrías mapear a ordinal).
- En sistemas reales de IP reputation se usan niveles / scores similares para indicar confiabilidad o peligrosidad.Abnormal AI+1

threat_type

- Tipo de amenaza asociado a la IP:
  - benign → tráfico normal
  - brute_force → intentos masivos de login
  - suspicious_login → logins anómalos
  - credential_stuffing → ataques con credenciales robadas
  - internal_traffic → tráfico interno corporativo
- Tipo: string.
- Muy útil para construir métricas Gold (por ejemplo, nº de eventos desde IPs con credential_stuffing).

last_seen

- Última fecha/hora en que se observó actividad relevante de esa IP.
- Tipo: datetime.
- Permite saber si la amenaza es reciente o antigua.