Licenciatura en ciencia de datos

Proceso ETL con Python, Pandas y SQLAlchemy

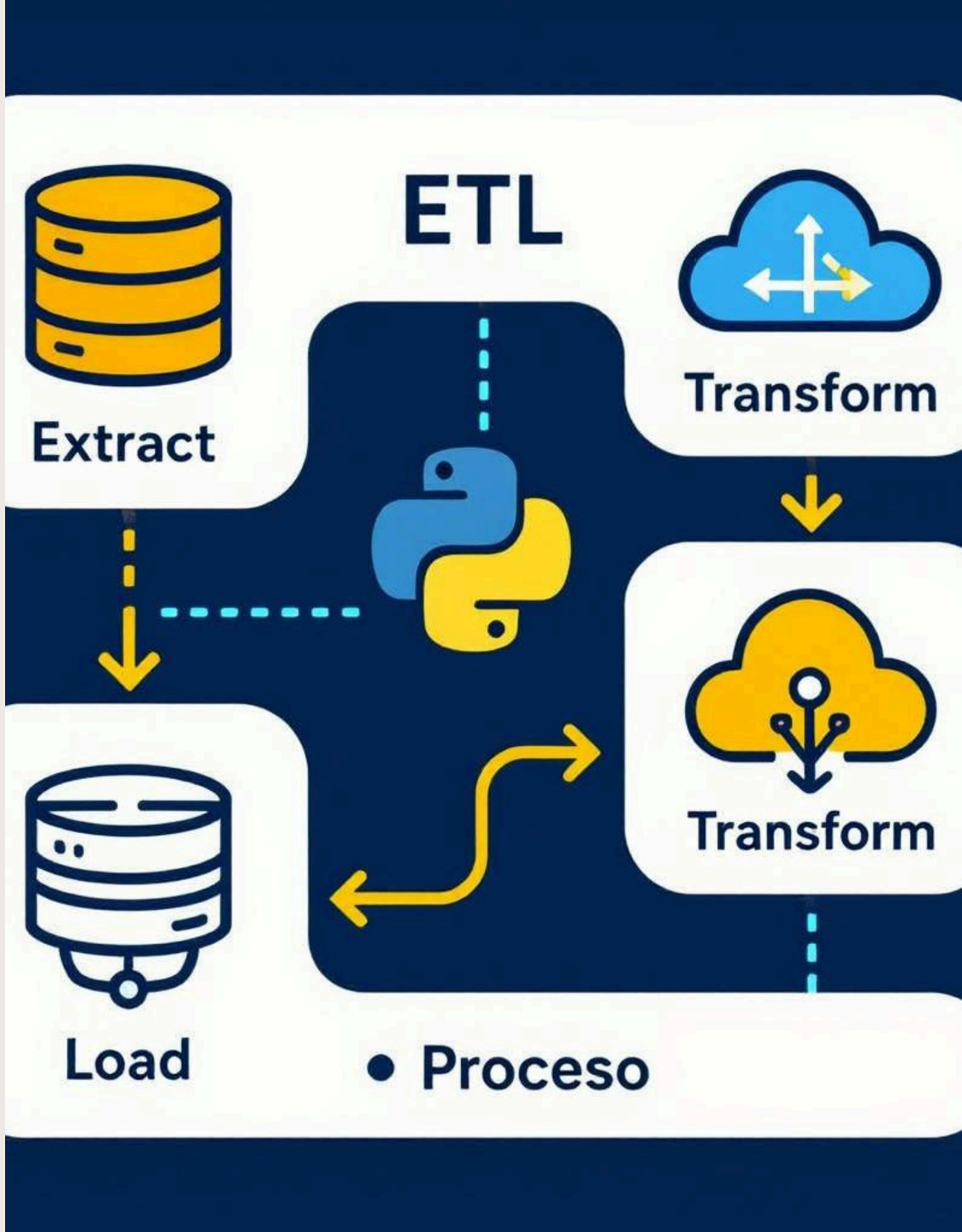
Tabla de
CONTENIDO

Definición de ETL

ETL: Extraer, Transformar y Cargar datos desde diversas fuentes para su análisis eficaz.

- **Proceso de extracción**: Consiste en obtener datos brutos desde bases, archivos o APIs para su posterior tratamiento.

- **Importancia del ETL**: Fundamental para integrar datos heterogéneos y asegurar análisis coherente y confiable en sistemas destino.



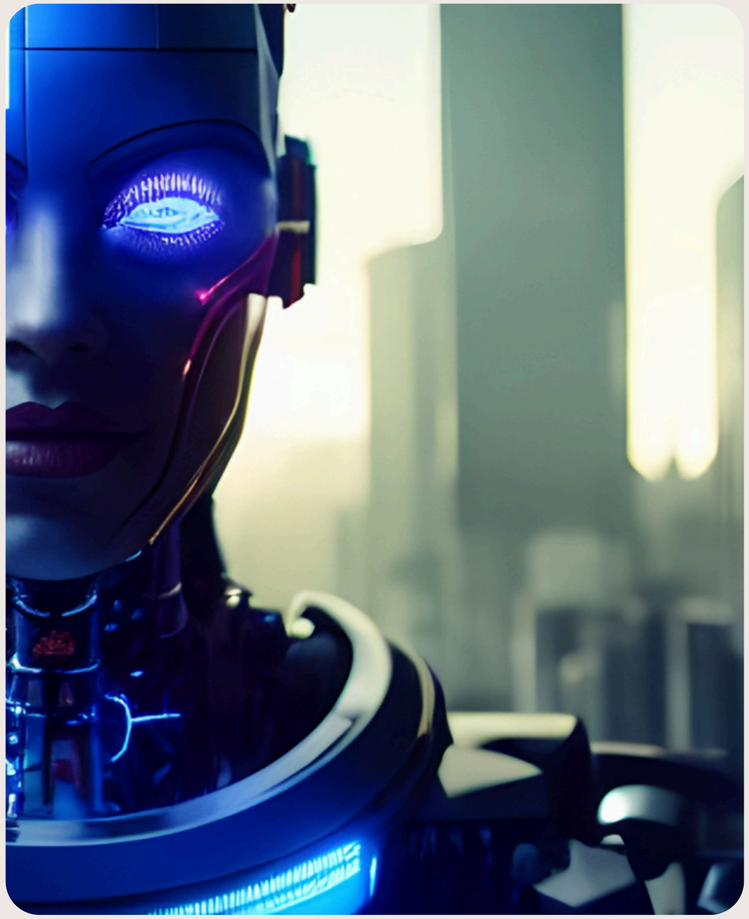
DISEÑO DE FLUJO ETL

CARACTERÍSTICAS

- **Modularidad** para reutilización: Dividir el proceso ETL en funciones claras permite mantenimientoeficiente y reutilización en distintos proyectos.

- **Escalabilidad** para grandes volúmenes: Implementar procesos que soporten crecimiento de datosmejora capacidad y evita cuellos de botella futuros.

- **Optimización** de rendimiento: Priorizar métodos eficientes en lectura, transformación y carga reduce tiempos y uso de recursos computacionales.



HERRAMIENTAS CLAVES

Pandas y SQLAlchemy

¿Por qué?

- **Manipulación eficiente con Pandas:** Pandas ofrece estructuras DataFrame para manejar datostabulares y realizar transformaciones complejas fácilmente.

- **Interacción robusta con bases de datos mediante SQLAlchemy:** SQLAlchemy posibilita modelarbases SQL como objetos Python y gestionar conexiones con múltiples motores.

- **Sinergia para ETL eficaz:** Combinar Pandas y SQLAlchemy facilita transformar datos en memoria y cargarlos rápidamente a bases relacionales.

Extracción de datos con Pandas

- **Funciones clave** para lectura de datos: read_csv, read_excel y read_json para importar datos estructurados desde archivos locales.

- **Extracción** mediante APIs y requests: Requests permite obtener datos JSON de APIs REST y convertirlos fácilmente a DataFrames con pandas.

Transformación de datos con Pandas

- **Limpieza de datos en DataFrames:** Eliminar valores nulos con dropna y duplicados con drop_duplicates para asegurar calidad consistente.
- **Renombrar y filtrar columnas y filas:** Usar rename para cambiar nombres y query o boolean indexing para filtrar filas específicas.
- **Transformaciones personalizadas con lambda y map:** Aplicar funciones lambda y map a columnaspara modificar datos de forma flexible y eficiente.

Conexión a base de datos con SQLAlchemy

- **Creación de engine con create_engine:** create_engine genera objeto conexión que gestiona conexióny sesiones hacia bases SQL.
- **Cadenas de conexión para SQLite:** Ejemplo SQLite: sqlite:///ruta_al_archivo.db conecta base local sin necesidad de usuario ni contraseña.
- **Cadenas de conexión para PostgreSQL:** PostgreSQL usapostgresql://usuario:contraseña@host:puerto/nombre_bd para acceso remoto seguro.

Carga de datos transformados

- **Método to_sql para carga**: Utiliza to_sql para insertar DataFrames en tablas SQL, integrando pandas con bases relacionales

- **Modos de inserción**: Append agrega registros sin eliminar existentes, replace sobrescribe tabla completa al cargar datos nuevos.

- **Gestión de índices y duplicados:** Controla índice con parámetro index; para duplicados, usar claves únicas y manejo previo en transformación para importar datos estructurados desde archivos locales.

Automatización del proceso ETL

- **Modularización** con funciones y clases: Organizar código en funciones y clases mejora claridad, mantenibilidad y facilita reutilización en múltiples proyectos.

- **Herramienta** schedule para tareas simples: Schedule permite programar ejecuciones periódicassencillas vía Python para tareas ETL recurrentes y regulares.

- **Airflow** para flujos y dependencias complejas: Airflow gestiona DAGs para coordinar, monitorear y automatizar pipelines ETL complejos con dependencia entre tareas.

BUENAS PRACTICAS Y ERRORES COMUNES

Guide lines

- Uso de logging para seguimiento: Implementar logging permite monitorear ejecución, identificarcuellos de botella y facilitar la depuración precisa.

- Validación de datos en cada etapa: Validar formatos y rangos en cada fase garantiza integridad y calidad antes de continuar al siguiente paso.

- Manejo robusto de excepciones: Capturar y registrar errores evita caídas del proceso y facilita análisisposterior para mejora continua.

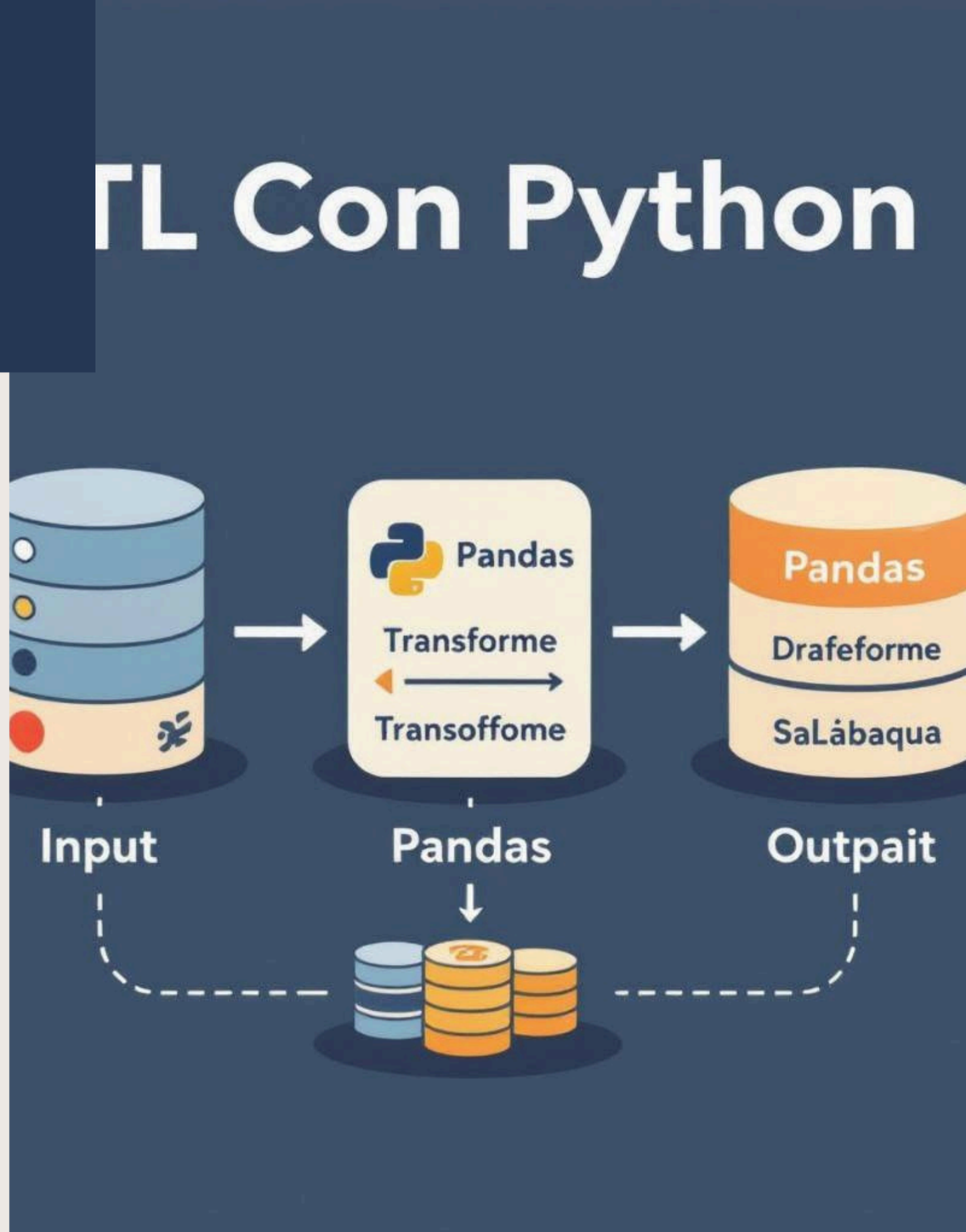
EJEMPLOS

Ejemplos

- **Integración del proceso ETL**: Código muestra extracción de CSV, limpieza de datos y carga en base SQLite con Pandas y SQLAlchemy.

- **Explicación paso a paso**: Extracción con pandas.read_csv, transformación con pandas para limpieza y filtro, carga con DataFrame.to_sql.

- **Resumen y flujo unificado**: El pipeline ETL conecta cada etapa secuencialmente, garantizando integridad y automatización del procesamiento de datos.



¡GRACIAS POR LA ATENCIÓN!