

Análisis de Caso

OBTENCIÓN DE DATOS DESDE ARCHIVOS 

Análisis de Caso

Nombre del tema: Obtención de Datos desde Archivos con Pandas

Situación inicial

Una empresa de consultoría trabaja con grandes volúmenes de datos provenientes de diferentes fuentes, como archivos CSV, Excel y tablas web. Sin embargo, han identificado problemas en la integración y análisis de la información debido a formatos inconsistentes y datos incompletos. Actualmente, el equipo depende de procesos manuales que resultan ineficientes y propensos a errores.

El equipo ha decidido utilizar Pandas para optimizar la obtención y manipulación de estos datos. Sin embargo, necesitan una estrategia eficiente para cargar la información, limpiar los registros y exportarlos en un formato estándar para su análisis posterior. Tu misión será desarrollar una solución utilizando Pandas para automatizar la obtención de datos desde diferentes archivos y garantizar que sean procesados correctamente.

Descripción del Caso

En este caso, asumirás el rol de un **analista de datos** en la empresa, encargado de diseñar un flujo de trabajo eficiente para la carga, limpieza y exportación de datos. Deberás trabajar con distintas fuentes de datos y aplicar técnicas de preprocesamiento en **Pandas** para garantizar su calidad.

Las tareas incluirán:

- Cargar datos desde archivos **CSV, Excel y tablas web**.
- Identificar y manejar **valores nulos y duplicados**.
- Convertir formatos y asegurarse de que las columnas contengan tipos de datos correctos.
- Exportar los datos procesados a formatos compatibles para su posterior análisis.

Deberás aplicar **Pandas** para estructurar y transformar los datos de manera eficiente, optimizando su calidad y asegurando su correcta integración.

Instrucciones

Para completar este caso, realiza las siguientes tareas:

1. **Carga de datos desde distintos archivos:**
 - Importa un archivo **CSV** en un DataFrame de Pandas.
 - Carga un archivo **Excel** en otro DataFrame.
 - Extrae información de una **tabla web** utilizando `read_html()`.
2. **Limpieza y estructuración de datos:**
 - Identifica valores nulos y decide si deben ser imputados o eliminados.
 - Elimina filas duplicadas si es necesario.
 - Verifica y ajusta los tipos de datos en columnas numéricas y categóricas.
3. **Transformación y optimización de datos:**
 - Selecciona las columnas más relevantes para el análisis.
 - Renombra columnas para mejorar la legibilidad.
 - Ordena los datos en función de una columna clave.
4. **Exportación de datos:**
 - Guarda el DataFrame limpio en un **archivo CSV** sin incluir el índice.
 - Exporta los datos procesados a **Excel** para su visualización.

Entregables

Los participantes deberán entregar un informe con:

1. **Código fuente en Python** con la implementación de las tareas.
2. **Explicación detallada** de cada paso realizado y su justificación técnica.
3. **Ejemplo de los datos antes y después de la limpieza.**
4. **Conclusiones** sobre la importancia de un proceso estructurado de obtención de datos.

¡Muchas gracias!

Nos vemos en la próxima lección

