

# Manual de Usuario para el Script de Conversión y Transformación de Datos

Grupo de Control y Procesamiento Digital de Señales  
Universidad Nacional de Colombia  
CHEC

April 28, 2025

## Contents

<b>1 Descripción General</b>	<b>1</b>
<b>2 Funcionalidades Principales</b>	<b>1</b>
<b>3 Bibliotecas Necesarias</b>	<b>2</b>
<b>4 Archivos Necesarios</b>	<b>2</b>
<b>5 Instrucciones de Uso</b>	<b>2</b>
5.1 Preparación del Entorno . . . . .	2
5.2 Ejecución del Script . . . . .	2
5.3 Transformación de Datos . . . . .	2
5.4 Verificación de Resultados . . . . .	3
<b>6 Consideraciones Adicionales</b>	<b>3</b>

## 1 Descripción General

Este script de Python, denominado *conversion\_datos\_csv\_pkl.py*, está diseñado para realizar la conversión y transformación de datos almacenados en archivos CSV a formato Pickle. El objetivo principal es facilitar la manipulación y análisis de grandes conjuntos de datos relacionados con eventos críticos en una red de distribución eléctrica. El script proporciona funcionalidades para leer archivos CSV, convertir tipos de datos, y guardar los datos transformados en formato Pickle, lo cual es más eficiente para la lectura y escritura de datos en Python.

## 2 Funcionalidades Principales

- **Lectura de Archivos CSV:** El script puede leer archivos CSV que contienen datos de eventos críticos.
- **Conversión de Tipos de Datos:** Convierte las columnas de los datos leídos a tipos de datos específicos (por ejemplo, fechas a *datetime*, números a *float32*).
- **Guardado en Formato Pickle:** Los datos transformados se guardan en formato Pickle, lo que permite una carga más rápida y eficiente en futuras operaciones.
- **Transformación de Datos:** Realiza varias transformaciones en los datos, como normalización de fechas, conversión de coordenadas geográficas y métricas numéricas.
- **Compatibilidad con Múltiples Archivos:** Puede manejar múltiples archivos de datos y aplicar las mismas transformaciones a cada uno.

## 3 Bibliotecas Necesarias

Para ejecutar este script, es necesario tener instaladas las siguientes bibliotecas de Python:

- **pandas:** Utilizado para manipulación y análisis de datos.
- **numpy:** Utilizado para operaciones numéricas.
- **datetime:** Utilizado para manipulación de fechas y horas.
- **matplotlib:** Utilizado para visualización de datos (aunque no se utiliza en este script, está importado).

Puede instalar estas bibliotecas utilizando pip:

```
pip install pandas numpy matplotlib
```

## 4 Archivos Necesarios

Los siguientes archivos deben existir en las rutas especificadas para que el script funcione correctamente:

- **SuperEventos\_Criticidad\_AguasAbajo.csv:** Archivo CSV que contiene datos de eventos críticos.
- **Eventos\_interruptor.pkl:** Archivo Pickle que contiene datos de eventos relacionados con interruptores.
- **Eventos\_tramo\_linea.pkl:** Archivo Pickle que contiene datos de eventos relacionados con tramos de línea.
- **Eventos\_transformador.pkl:** Archivo Pickle que contiene datos de eventos relacionados con transformadores.

## 5 Instrucciones de Uso

### 5.1 Preparación del Entorno

1. **Instalar Bibliotecas:** Asegúrese de tener instaladas todas las bibliotecas necesarias. Si alguna falta, instálela utilizando pip. 2. **Verificar Rutas de Archivos:** Asegúrese de que los archivos necesarios se encuentren en las rutas especificadas en el script. Si es necesario, modifique las rutas en el script para que coincidan con la ubicación de sus archivos.

### 5.2 Ejecución del Script

1. **Abrir Terminal o Consola:** Abra una terminal o consola en su sistema. 2. **Navegar al Directorio del Script:** Use el comando `cd` para navegar al directorio donde se encuentra el script. 3. **Ejecutar el Script:** Ejecute el script utilizando el siguiente comando:

```
python conversion_datos_csv.pkl.py
```

### 5.3 Transformación de Datos

El script realizará las siguientes transformaciones en los datos:

- Convertirá las columnas de fecha a tipo *datetime*.
- Convertirá las columnas numéricas a tipo *float32* o *float64* según sea necesario.
- Normalizará las fechas a periodos mensuales si es necesario.
- Guardará los datos transformados en archivos Pickle en las rutas especificadas.

## 5.4 Verificación de Resultados

1. **Comprobar Archivos de Salida:** Verifique que los archivos Pickle se hayan creado correctamente en las rutas especificadas. 2. **Abrir Archivos Pickle:** Puede abrir los archivos Pickle utilizando pandas para verificar que los datos se hayan transformado correctamente:

```
import pandas as pd
df = pd.read_pickle('ruta_del_archivo.pkl')
print(df.head())
```

## 6 Consideraciones Adicionales

- **Rutas de Archivos:** Asegúrese de que las rutas de los archivos en el script sean correctas y accesibles. Si los archivos se encuentran en una ubicación diferente, actualice las rutas en el script.
- **Permisos de Lectura/Escritura:** Asegúrese de tener los permisos necesarios para leer y escribir archivos en las ubicaciones especificadas.
- **Memoria Suficiente:** Asegúrese de que su sistema tenga suficiente memoria RAM para manejar los datos, especialmente si los archivos son grandes.
- **Compatibilidad de Python:** Este script está diseñado para funcionar con Python 3. Asegúrese de que su entorno de desarrollo utilice esta versión.
- **Errores de Conversión:** Si encuentra errores durante la conversión de tipos de datos, revise los datos originales para asegurarse de que estén en el formato correcto.