Te dejamos el desafío, para resolverlo **tienen dos semanas a partir de mañana** (Fecha fin de entrega 02/02/2023), donde nos deberán entregar el desafío para poder ser evaluado. Luego acordaremos una pequeña reunión donde te consultaremos dudas sobre la resolución.

Este video explica la descarga y puesta en marcha de la Herramienta.

<https://www.youtube.com/watch?v=cxvH-xNTtvY>

Si bien el video es de ayuda, se ve que esta desactualizado respecto de la página de descarga de la versión **pentaho-community-edition**por lo que para ayudarlos, les dejo el link especifico de descarga.

<https://www.hitachivantara.com/en-us/products/dataops-software/data-integration-analytics/pentaho-community-edition.html>

La versión que vamos a usar para el desafío es pdi-ce-9.4.0.0-343.zip

Una vez descargada, en el video se explica cómo correr la **IDE**, les recomiendo que lo descompriman en: c.\pentaho\pdi-version-9.4\ de esta forma luego podrán tener otras versiones de la herramienta instalada en forma simultánea y ordenada.

Enunciado del ejercicio:

Datos del Archivo.

Dado el archivo **artist\_credit.csv**

Para descargar el archivo  (<https://drive.google.com/drive/folders/1h2Aqxn6WZ7C9LNvEZjMLSXzFebTiJhzf?usp=sharing>)

Tiene los siguientes campos

**Id = identificador de registro**

**name = Nombre del Artista**

**artist\_count = Créditos**

**ref\_count = Referecias**

**created = Fecha de Creación**

Se requiere generar un programa en **Kettle**, que sea capaz de procesar el archivo  **artist\_credit.csv**, el programa debe recibir una fecha por parámetro al ejecutarlo.

Se deberán generar tres salidas.

1. Generar una tabla de salida en una base de datos (Cualquier DB), en la tabla se deberán guardar todos los artistas cuya fecha de creación (**created**) sea mayor a la fecha ingresada por parámetro.
   1. Campos de la tabla de salida

                                                    i.     ID – Auto incremental

                                                   ii.     name

                                                  iii.     created

* 1. La tabla debe ser cargada con los datos ordenados por campo **name** en forma ascendente.

1. Generar una salida en un archivo Excel, en el que se guardarán todos los artistas cuya fecha de creación (**created**) sea menor a la fecha ingresada por parámetro.
   1. Campos del archivo de salida

                                                    i.     name

                                                   ii.     created

                                                  iii.     suma = **artist\_count + ref\_count (Colocar la suma de ambos valores)**

* 1. El archivo debe ser cargado ordenado por **id**de forma descendente.

1. Generar una salida en un archivo tipo .csv en la que se deberán guardar todos los artistas donde el año de la fecha de creación sea igual al año de la fecha ingresada por parámetro.
   1. Campos del archivo de salida

                                                    i.     name

                                                   ii.     created

                                                  iii.     periodo

                                                  iv.     concat = name + periodo (Colocar la concatenación de ambos campos)

* 1. El archivo debe ser cargado ordenado por fecha (created) de forma descendiente.

1. Subir el código de la/s transformaciones/Jobs a un repositorio git público (archivos .ktr y .kjb)
2. Generar un archivo .bat o .sh que sea capas de ejecutar la aplicación kettle utilizando **pan** o **kitchen**, subir al repositorio él archivo.