# Práctica 1+3: Reconciliación

Contenido

[Práctica 1+3: Reconciliación 1](#__RefHeading__529_2017671621)

[Objetivos 1](#__RefHeading__531_2017671621)

[Propósito del curso al que apoya la práctica 1](#__RefHeading__533_2017671621)

[Actividades 1](#__RefHeading__535_2017671621)

[Ejercicios 1](#__RefHeading__537_2017671621)

[1.- Reconciliación utilizando OpenRefine 1](#__RefHeading__539_2017671621)

[2.- Implementación de reconciliación 2](#__RefHeading__541_2017671621)

[Entregables 2](#__RefHeading__545_2017671621)

[Ubicación física de los datos 2](#__RefHeading__547_2017671621)

## Objetivos

1. Entender una aplicación práctica de las técnicas de similitud: la reconciliación. Utilizar una herramienta ya existente e implementar una solución en Java.

## Propósito del curso al que apoya la práctica

Entender diversas técnicas de minería de datos, conocer sus aplicaciones y desarrollar su implementación.

## Actividades

* Reconciliar datasets.

## Ejercicios

## 1.- Reconciliación utilizando OpenRefine

1. Instala OpenRefine en tu máquina.
2. Descarga el jar de Reconcile-csv de la página: http://okfnlabs.org/reconcile-csv/
3. Define un dataset en la clase en formato CSV. El dataset debe tener un campo tipo String el cual se quiera reconciliar.
4. Define el master data en formato CSV. El master data debe tener un campo String con los datos maestros para la reconciliación y un campo con un ID único.
5. Carga los datos con el campo a reconciliar en OpenRefine.
6. Corre el servicio de reconciliación.

java -Xmx2g -jar reconcile-csv-0.1.2.jar <CSV-File> <Search Column> <ID Column>

7.- Registra en servicio de reconciliación en OpenRefine

8.- Corre la reconciliación, cuáles son los resultados?

## 2.- Implementación de reconciliación

1. De ser necesario añade o modifica los datos a reconciliar.
2. Descarga del moodle los jars necesarios para el código.
3. Abre eclipse y crea un nuevo proyecto Java.
4. Crea clases para procesar tanto los datos a reconciliar como el master data. Asegúrate de sobrecargar el método de toString.
5. Carga ambas fuentes de datos en un ArrayList de los tipos que acabas de crear.
6. Genera el ciclo principal, el cuál está compuesto por otro ciclo que habrá de comparar cada entrada con todas las cadenas del Master data.
7. Se crearán equipos de 2 o 3 personas para investigar los siguientes tipos de distancias, se darán 10 minutos para investigar y 15 para presentar al grupo una diapositiva.

A) Levenshtein

B) Jaro

C) Dice

D) Euclidean

E) Jaccard

F) Longest Common Substring

G) Monge Elkan

H) Needleman Wunch

I) Qgrams Distance

J) Simon White

K) Smith Waterman

8.- Al momento de la comparación utilizar alguna de las distancias vistas previamente con alguna heurística.

9.- Comparar con los resultados de OpenRefine.

## Entregables

* Un documento en formato Doc, Docx o PDF, que integre las respuestas a las preguntas expuestas en los ejercicios y a las preguntas de análisis posteriores a los ejercicios

### Ubicación física de los datos

Los datos se generarán en clase.