Guía para la limpieza de datos sobre biodiversidad con OpenRefine

Paula F. Zermoglio, Camila A. Plata Corredor, John R. Wieczorek, Ricardo Ortiz Gallego, Leonardo Buitrago

Versión 3.0, 2021-02-25 12:56:06 UTC

Tabla de Contenido

Colofón

Cita bibliográfica sugerida

Zermoglio PF, Plata Corredor CA, Wieczorek JR, Ortiz Gallego R & Buitrago L (2021) Guía para la limpieza de datos sobre biodiversidad con OpenRefine. Versión 3. Copenhagen: GBIF Secretariat. https://doi.org/10.15468/doc-gzjg-af18.

Autores

Paula F. Zermoglio, Camila A. Plata Corredor, John R. Wieczorek, Ricardo Ortiz Gallego & Leonardo Buitrago

Licencia

URI persistente

https://doi.org/10.15468/doc-gzjg-af18

Control de documentos

Versión 3, Febrero 2021.

Este documento se basa en dos publicaciones anteriores producidas por los mismos autores de esta Guía.

Imagen de la portada

Una manada de guanaco (*Lama guanicoe*), Lago Argentino, Santa Cruz, Argentina. Foto 2016 Diego Carús via iNaturalist Research-grade Observations, licenciada bajo CC BY-NC 4.0.

Prefacio

Objetivo

La presente guía ha sido construida con fines únicamente pedagógicos. El objetivo de esta guía es mostrar cómo utilizar algunas de las funciones de OpenRefine que pueden utilizarse para evaluar y mejorar la calidad de datos de biodiversidad.

Cómo usar esta guía

En esta guía se muestra cómo utilizar algunas funciones de OpenRefine para la evaluación y mejoramiento de la calidad de un conjunto de datos de biodiversidad. Los ejemplos de uso presentados en esta guía constituyen sólo algunas de las alternativas posibles para el tratamiento de datos en OpenRefine.

Para hacer un mejor uso de esta guía, se recomienda seguir los pasos utilizando el programa OpenRefine y el conjunto de datos modelo provisto junto con este documento. Todos los ejemplos presentados corresponden a un conjunto de datos de biodiversidad que ha sido específicamente modificado por los autores.

En la Guía se utilizan los términos 'campo' y 'columna' indistintamente.

En el texto se utiliza el siguiente formato para los `"nombres de los campos originales"`.

Software

La versión de OpenRefine utilizada para la confección de esta guía es OpenRefine 3.3. Aunque no es la última versión disponible, es la última con la cual todos los pasos explicados en esta Guía han sido probados sin presentar problemas.

Para ver detalles sobre cómo descargar e instalar OpenRefine en su computadora, ver el Apéndice 1.

Datos

El conjunto de datos modelo utilizado en esta guía puede obtenerse aquí: EjercicioModelo_OpenRefine_Datos.csv. Descargar el archivo, y no manipularlo en otros programas (e.g., MS Excel) antes de abrirlo en OpenRefine, dado que ello puede cambiar los formatos y/o codificación del archivo.

El conjunto de datos modelo fue derivado a partir de:

Williams J (2018). Colección de Herbario. Version 3.1. Facultad de Ciencias Naturales y Museo - U.N.L.P.. Occurrence dataset https://doi.org/10.15468/i9bj5r accessed via GBIF.org on 2019-04-18.

1. Primeros pasos: datos y proyectos

1.1. Carga de datos y creación de un proyecto

Para comenzar a utilizar OpenRefine debe cargar sus datos en el programa y crear un proyecto. Para ello, siga los siguientes pasos:

Paso 1

Abra la aplicación OpenRefine. Si utiliza Windows, se abrirá una ventana de comandos que mostrará las acciones que OpenRefine está realizando (Figura 1). ***No cierre esta ventana mientras esté trabajando con el programa.***

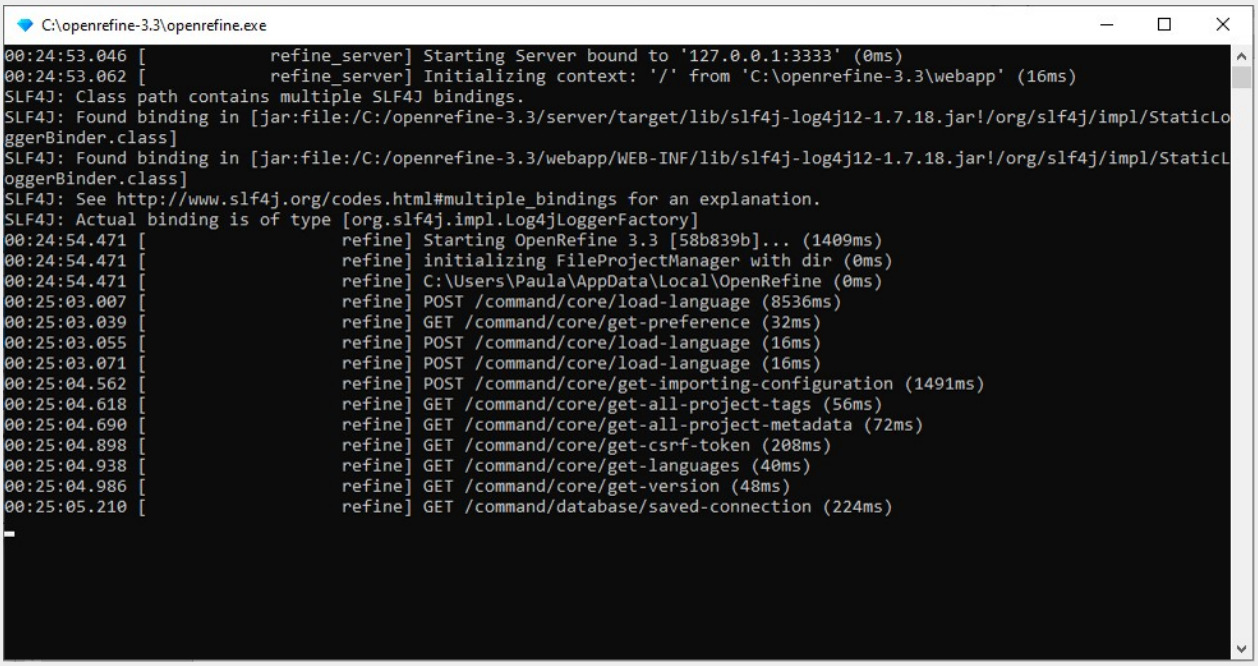

Figura 1

OpenRefine se abrirá en el navegador que usted utilice por defecto inmediatamente después de ejecutar la aplicación (Figura 2). Si OpenRefine no abre, puede acceder manualmente ingresando la siguiente URL en su navegador:

http://127.0.0.1:3333

Si bien puede ejecutar el programa utilizando cualquier navegador, nuestra recomendación es que utilice Google Chrome, pues otros navegadores suelen presentar diversos problemas, sobre todo cuando se intentan utilizar funciones más complejas.

Si bien OpenRefine utiliza la interfaz de un navegador para trabajar, todos los procesos (e.g., creación de proyectos, carga de datos, procesamiento, etc.) se realizan localmente, es decir, en su computadora, sin subir datos en línea. Por tanto, para utilizar la mayoría de las funciones del programa no hace falta una conexión a Internet. Algunas funciones, sin embargo, como las Consultas a servicios externos, sí requieren conexión.

Figura 2

En el menú de la izquierda tiene opciones para crear, abrir o importar proyectos. Si usted no tiene ningún proyecto aún, en la opción de “Abrir proyecto” verá una lista vacía.

Además puede cambiar la configuración de idioma. Para ello, haga click en “Idioma” y en la siguiente pantalla (Figura 3) seleccione el idioma preferido. Acepte los cambios. En esta guía se utilizará el idioma Español.

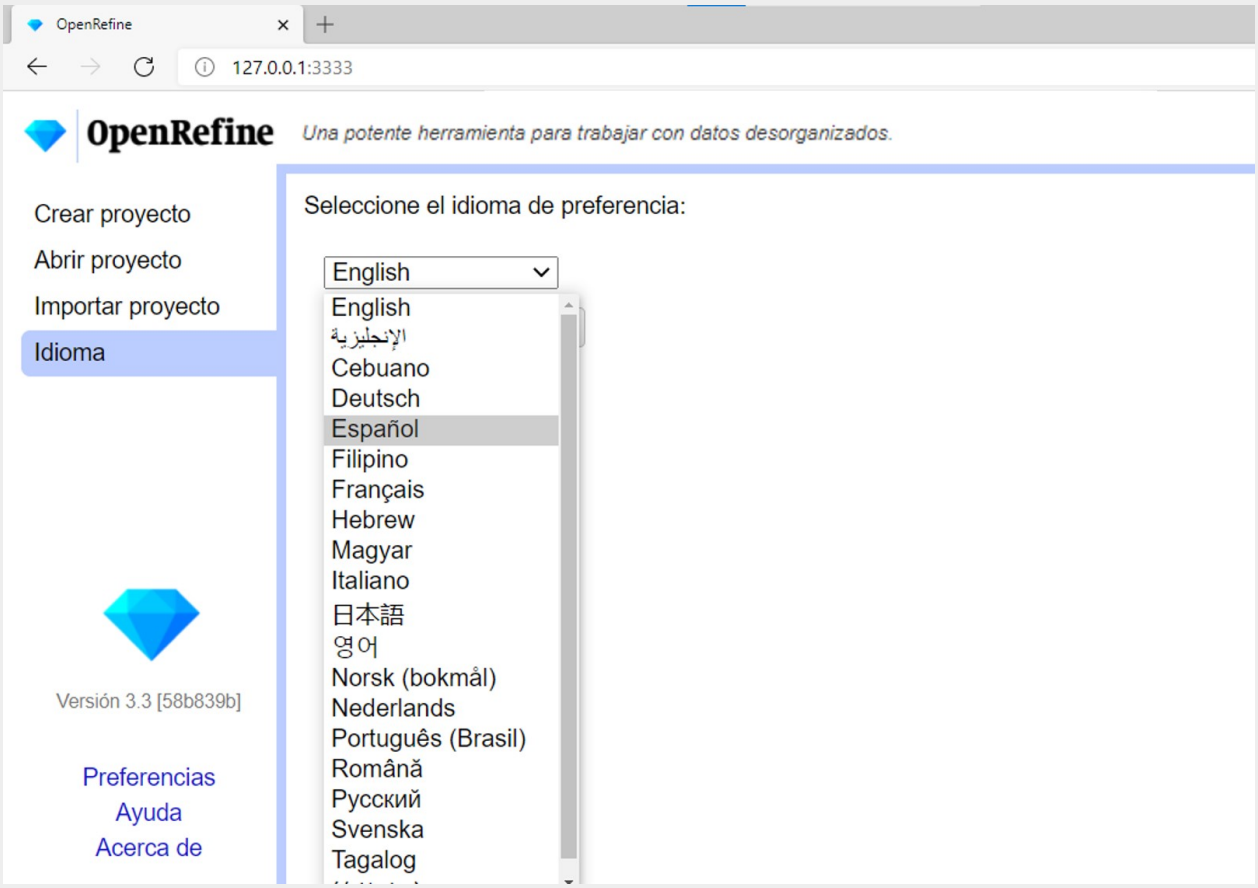

Figura 3

Paso 2

Cargue los datos (Figura 2). Dentro de la opción “Crear proyecto”, escoja el archivo que desea cargar. Note que hay varios formatos posibles de archivos que se pueden subir (tsv, csv, xls, json, etc). Haga click en “Siguiente”.

Para seguir esta guía, cargue el archivo proporcionado, al que puede acceder a través del enlace provisto en la sección Cómo usar esta guía: Datos.

Si sube archivos con formato .xls o .xlsx, tenga en cuenta que no podrá modificar la codificación, y que pueden encontrarse algunos errores en los datos (ejemplo: los tildes en las palabras se verán como símbolos raros cuando cargue los datos). Para evitarse problemas, si trabaja con MS Excel es conveniente que exporte los datos como archivo .csv (de todas formas, tenga cuidado con la codificación, ver más abajo).

Verá entonces una pantalla como la que se muestra en la Figura 4. Allí puede encontrar dos cuadros de texto arriba a la derecha, uno para indicar el nombre de su proyecto, y otro que permite asignar etiquetas (“tags”) a su proyecto. Puede asignar tantas etiquetas como desee, escribiéndolas en el cuadro. Las etiquetas le ayudarán a organizar mejor sus proyectos, y podrá verlas y editarlas a través del menú “Abrir proyecto” junto al nombre de su proyecto (ver sección “Abrir un proyecto y modificar sus metadatos” más abajo para más detalles sobre cómo agregar o cambiar etiquetas y otros elementos de los metadatos de los proyectos). Además, en esta pantalla puede ver una muestra de sus datos (tabla) y modificar varios aspectos de la carga de los datos al programa: codificación, criterio para la separación en columnas, inclusión o no de las primeras filas, etc.

OpenRefine sugiere algunas de las codificaciones más utilizadas cuando se hace click en el cuadro de texto “Codificación de caracteres”. Asegúrese de escoger correctamente la codificación. Si está utilizando el conjunto de datos de prueba proporcionado, escoja UTF-8 (Figura 5).

OpenRefine presenta la opción de “Detectar y transformar texto en números, fechas, ...”. Si esta opción es seleccionada, el programa tratará de interpretar ciertos campos transformándolos a determinados formatos. Por ejemplo, si detecta campos de fecha, tratará de colocar los valores de las celdas de ese campo en formato de fecha estándar. Dada la naturaleza de los datos sobre biodiversidad con los que solemos trabajar, estas interpretaciones pueden ser incorrectas e introducir más errores. Asegúrese entonces de desmarcar esta opción durante el paso de importación de datos.

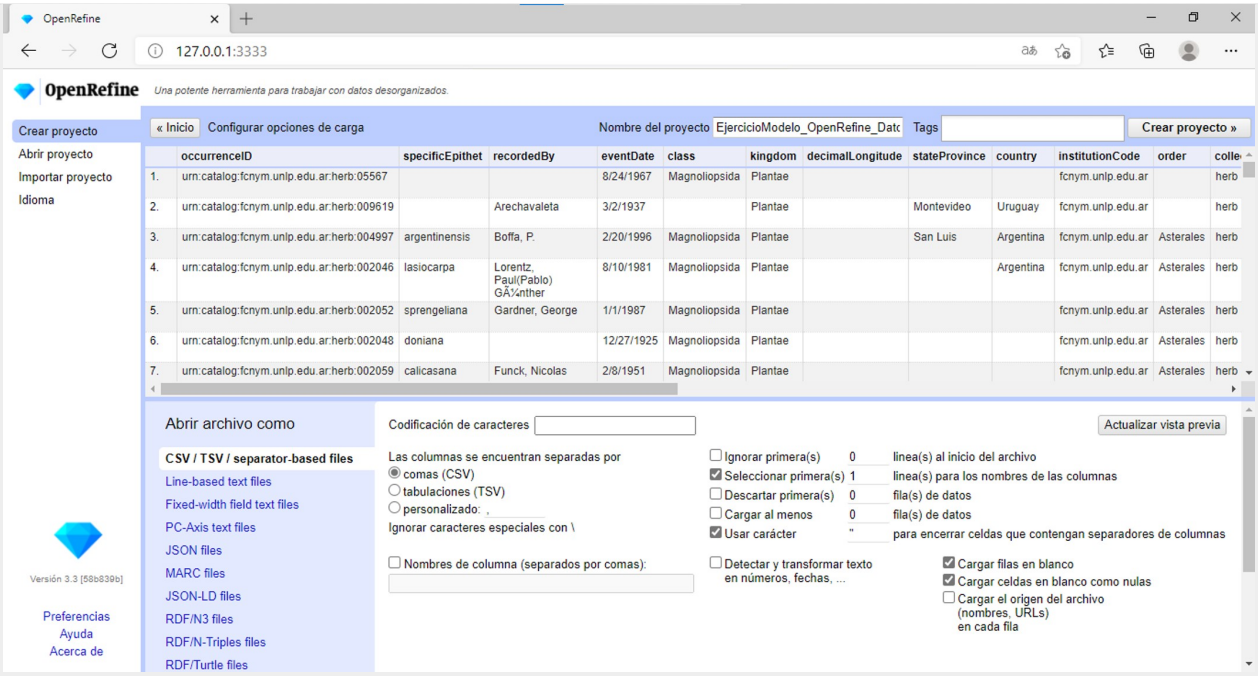

Figura 4

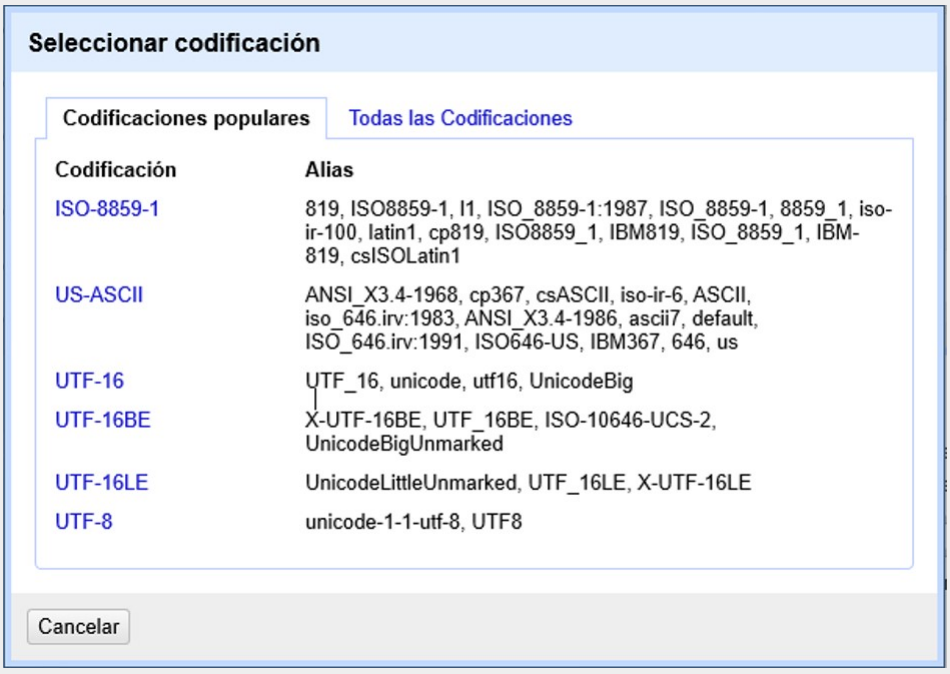

Figura 5

Paso 3

Cree el proyecto. Una vez que haya seleccionado las opciones de carga de datos, haga click en el botón “Crear Proyecto” arriba a la derecha.

Cuando se cargan datos en OpenRefine y se crea un proyecto, el programa hace una copia de los datos provistos. De esta forma, el archivo local original nunca se modifica, garantizando no perder los datos originales.

Paso 4

¡Felicitaciones! Ya tiene un proyecto (lo verá como en la Figura 6).

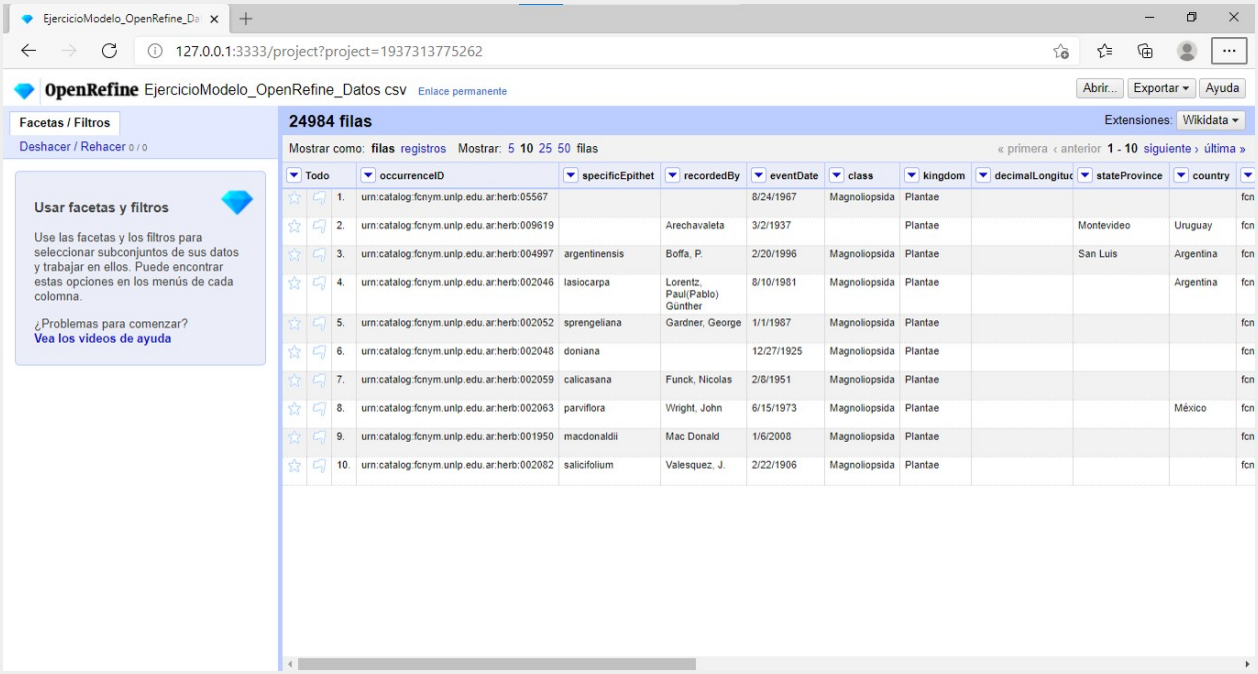

Figura 6

El número total de filas cargadas se muestra en este momento arriba de la tabla en negrita (para el caso del conjunto de datos provisto, 24984 filas). Sin embargo, verá que el número de filas mostradas en la tabla es limitado. No desespere, OpenRefine sólo muestra hasta 50 filas. Las acciones que uno pueda tomar en la aplicación, sin embargo, pueden tener efecto sobre otras filas aunque éstas no sean mostradas.

1.2. Abrir un proyecto y modificar sus metadatos

Una vez que ha creado uno o más proyectos, podrá acceder a ellos a través del menú “Abrir proyecto” (Figura 3. Cuando ingresa a este menú, verá listados todos sus proyectos, con una serie de metadatos básicos que puede utilizar para ordenar su lista (Figura 7a). Los metadatos mostrados incluyen el nombre del proyecto, la fecha de última modificación, las etiquetas asignadas (“tags”), el número de filas, etc.

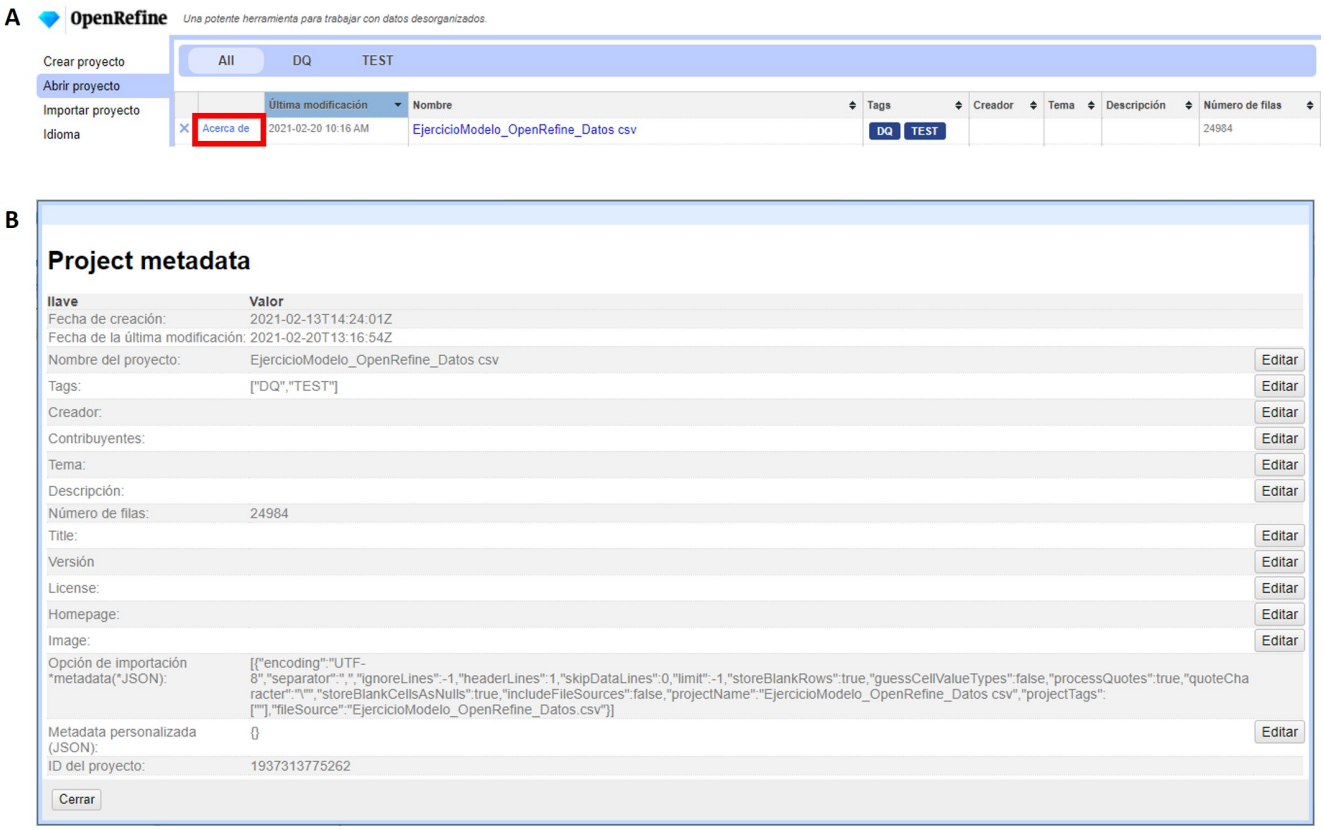

Figura 7

Si desea editar los metadatos, debe ingresar a la opción “Acerca de”, a la izquierda en la tabla de proyectos (Figura 7a). Se abrirá entonces una ventana como la mostrada en la Figura 7b. Si desea modificar los distintos elementos de los metadatos, puede utilizar el botón “Editar” sobre cada parámetro. Por ejemplo, si olvidó colocar una etiqueta al crear un proyecto, puede hacerlo más tarde desde este menú. Contar con buenos metadatos puede ayudarle a organizar sus proyectos, sobre todo lleva a cabo trabajo colaborativo.

1.3. Asignar suficientes recursos al programa

Para que OpenRefine pueda funcionar correctamente se requiere contar con suficiente memoria en la computadora asignada al programa. Especialmente si se trabajará con conjuntos de datos grandes, la memoria asignada afectará la velocidad de procesamiento e incluso podría limitar la capacidad para aplicar ciertas funciones.

Por defecto, el programa utiliza 1 gigabyte (GB, o 1024MB) de memoria, pero la cantidad de memoria asignada puede modificarse para optimizar el desempeño. Para conocer los detalles sobre cómo modificar la memoria asignada en distintos sistemas operativos, vea la documentación provista en el Manual del Usuario de OpenRefine.

2. Limpieza de datos

2.1. Manejo básico de columnas

2.1.1. Renombrar, eliminar y mover columnas

Veamos primero algunas funciones básicas que se pueden aplicar sobre los campos:

1. **Renombrar un campo.**

   Hacer click en **la ▼ azul del campo › Editar columnas › Renombrar esta columna**

2. **Eliminar un campo.**

   Hacer click en **la ▼ azul del campo › Editar columnas › Eliminar esta columna**

3. **Mover un campo.**

   Hacer click en

   ... la ▼ azul del campo › Editar columnas › Mover columna al principio

   ... la ▼ azul del campo › Editar columnas › Mover columna al final

   ... la ▼ azul del campo › Editar columnas › Mover columna a la izquierda

   ... la ▼ azul del campo › Editar columnas › Mover columna a la derecha

   Estas tres opciones pueden verse en la Figura 8.

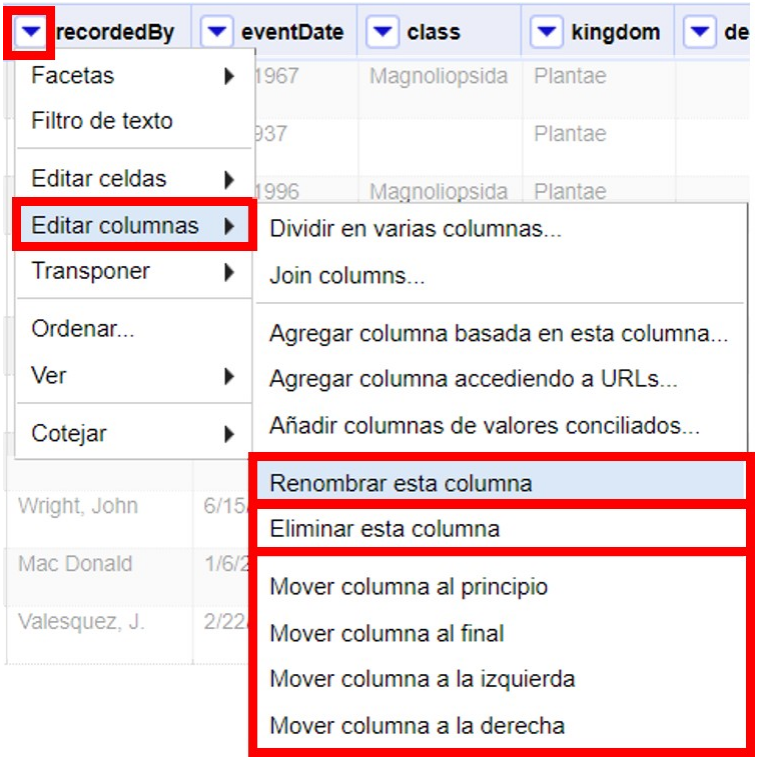

Figura 8

4. **Reordenar o eliminar varios campos a la vez.**

   Para esta función se utiliza el campo “Todo”, que se encuentra como primera columna de la tabla. Este campo no forma parte de los datos originales, es agregado por el programa para permitir llevar a cabo ciertas funciones.

   Hacer click en **la ▼ azul en el campo “Todo” › Editar columnas › Ordenar / Eliminar columnas...** (Figura 9a).

   Se abrirá entonces una ventana como la que se muestra en la Figura 9b. Allí puede ordenar los campos simplemente arrastrándolos arriba o abajo en la lista, y eliminarlos arrastrándolos hacia la parte derecha de la ventana. Una vez que termine de modificar el orden de los campos, haga click en “Aceptar”.

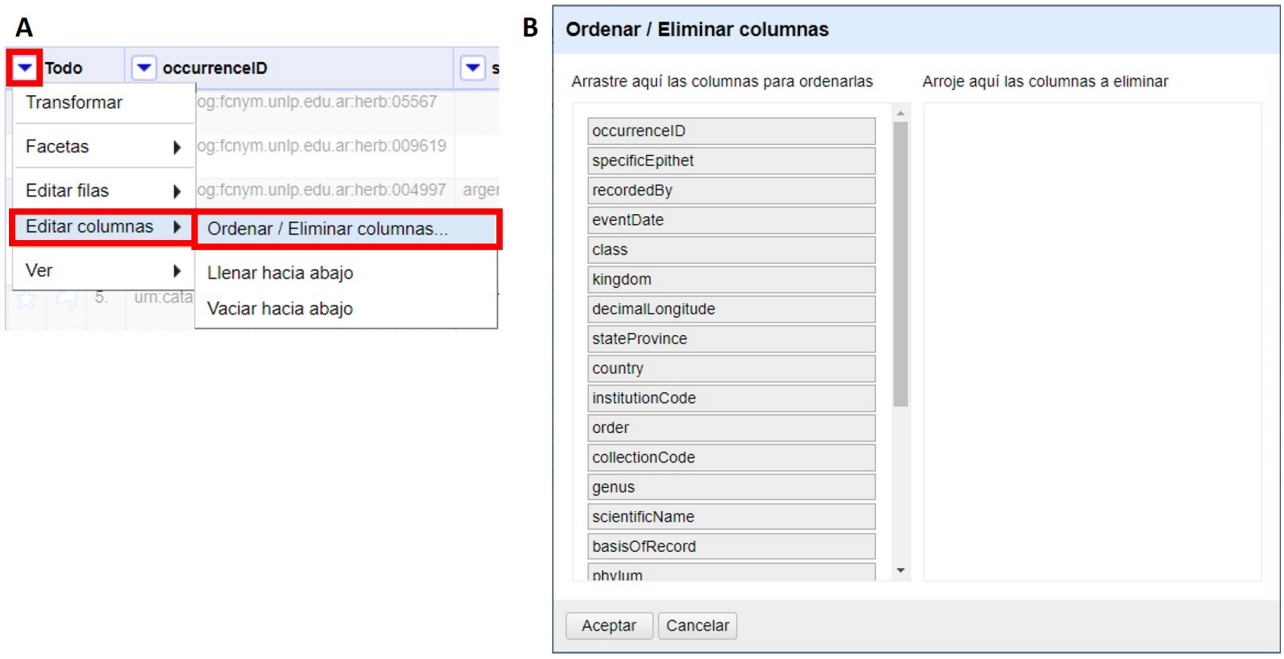

Figura 9

En OpenRefine se considera que cualquiera de los cuatro cambios descriptos anteriormente son cambios a los datos, y por ende se registran como tales en el historial de cambios (ver más abajo sección Deshacer y rehacer cambios).

2.1.2. Nuevas columnas vacías

Se pueden crear nuevos campos en base a cero, uno o más campos preexistentes.

Para crear un nuevo campo de cero, sobre cualquier columna preexistente siga la ruta:

Editar columnas › Agregar columna basada en esta columna...

Se abrirá una ventana como la que se muestra en la Figura 10.

Arriba de todo, coloque el nombre del nuevo campo.

Debe tener extremo cuidado al escoger los nombres que dará a las nuevas columnas. Considere que el nombre sea indicativo de lo que contiene (e.g., no utilice nombres tales como “Columna 1” o “Transformación 3”). OpenRefine no le dejará utilizar nombres que ya hayan sido utilizados para nombrar otros campos dentro del proyecto. Considere qué otros campos tiene en su base de datos original y no utilice nombres que ya hayan sido utilizados, se evitará así importar datos a columnas equivocadas al volver a su base de datos.

Luego, en el cuadro de texto “Expresión” escriba: `null`. Ello quiere decir que se creará un campo con valores nulos. Luego oprima “Aceptar”. Alternativamente, en vez de `null` puede colocar la expresión: `""`, y el nuevo campo tendrá valores en blanco.

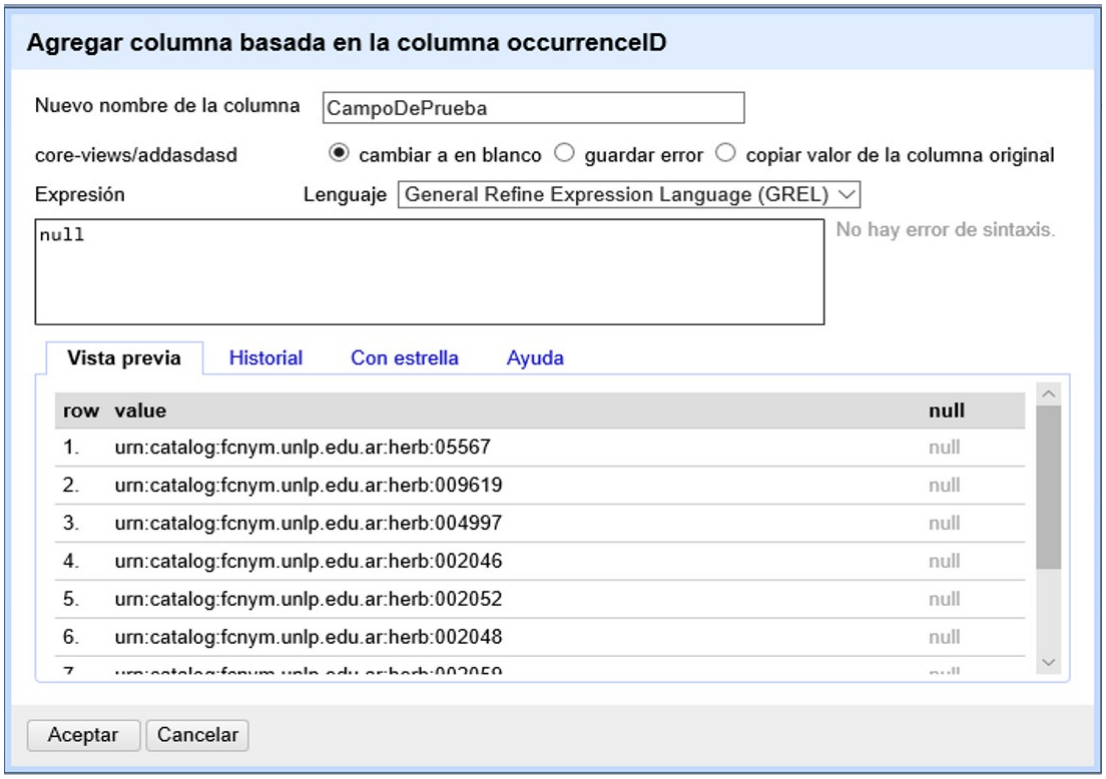

Figura 10

El nuevo campo, con el nombre que le haya dado, aparecerá a la derecha de aquel a partir del cual fue generado.

Tenga en cuenta que las columnas nuevas que cree en la aplicación no estarán en su base de datos original. Al importar los datos que han sido limpiados de regreso a su base de datos, dependiendo de cómo esté estructurada esa base de datos, es posible que estas nuevas columnas no sean importadas o que reciba un mensaje de error de importación porque el número de campos del archivo no coincide con el de la base de datos. En estos casos, debe asegurarse de agregar previamente los nuevos campos en su base de datos si desea importar todos los campos nuevos.

2.1.3. Nuevas columnas a partir transformaciones simples de otras columnas

Muchas veces no queremos modificar los datos directamente en los campos (columnas) en que se presentan, dado que queremos mantener los valores originales y/o queremos proveer información adicional basada en ciertos campos. Por ejemplo, podríamos tener como campos individuales el género y el epíteto específico y queremos agregar el campo nombre científico como concatenación de los dos; o viceversa: tenemos un único campo nombre científico y queremos mantener ese campo y proveer otros dos campos adicionales para género y epíteto, a partir de la división del anterior. Para estos casos es útil crear nuevos campos en nuevas columnas.

Veamos ahora cómo crear nuevas columnas con datos modificados a partir de columnas preexistentes.

Concatenaciones

Si desea crear un campo que sea la concatenación de otros dos campos separados puede seguir dos rutas, que se describen a continuación, de acuerdo a la versión del programa que utilice. La primera

ruta utiliza la función “Unir columnas” (o “Join columns”, como figura en el programa), y está disponible en la versión 3.3 y posteriores de OpenRefine. Las versiones anteriores del programa no tienen esta función, pero el mismo resultado puede obtenerse siguiendo la segunda ruta descripta abajo utilizando cualquier versión del programa. Para ambas rutas utilizaremos como ejemplo la concatenación de los campos `"genus"` y `"specificEpithet"`.

Ruta 1: Función “Unir columnas”

Click en **la ▼ azul del campo `"genus"` › Editar columnas › Join columns...** (Notar que el nombre de la función y los menúes que se despliegan están en inglés en esta versión del programa).

Se abrirá una nueva ventana (Figura 11), donde puede seleccionar todas las columnas que desea unir.

El primer recuadro a la derecha le permite seleccionar qué separador utilizar entre los contenido de cada columna. Por ejemplo, puede utilizar un espacio " ". Luego debe especificar qué hacer con los registros donde alguno de los campos a unir tiene valores nulos (por ejemplo, `"genus"` tiene un valor pero `"specificEpithet"` es nulo, o viceversa). Si elige la opción "Replace nulls with..." ("Reemplazar nulos con..."), puede especificar con qué reemplazar esos valores nulos (por ejemplo, algún caracter), o dejar ese recuadro vacío. Si, en cambio, escoge "Skip nulls" ("Saltear nulos"), para todos aquellos registros que tuvieran uno de los dos campos nulos no se llevará a cabo la unión.

Nota: Al escoger reemplazar los valores nulos y dejar el cuadro vacío (es decir, reemplazar por un caracter nulo), aún se hará la unión utilizando el separador indicado. Si el primer campo a unir tiene un valor no nulo y el segundo un valor nulo, el resultado será el valor del primer campo más el separador. Si el primer campo a unir tiene un valor nulo y el segundo un valor no nulo, el resultado será sólo el valor del segundo campo (sin separador). Para un ejemplo práctico, ver Tabla 1 más abajo.

Luego debe indicar si quiere los resultados sobre la misma columna sobre la que está actuando o en una columna nueva, y en ese caso, proveer un nombre para el nuevo campo. Puede llamar al nuevo campo "joint_scientificName", para indicar que se trata de la unión (note que ya hay un campo `"scientificName"` en los datos). Siempre es recomendable crear un nuevo campo, y en todo caso eliminar los campos innecesarios luego.

Por último, tiene la opción de eliminar las columnas que dieron origen a la unión ("Delete joined columns", "Eliminar columnas unidas"). Si desea conservarlas, como en este caso, asegúrese de que esa opción está desmarcada.

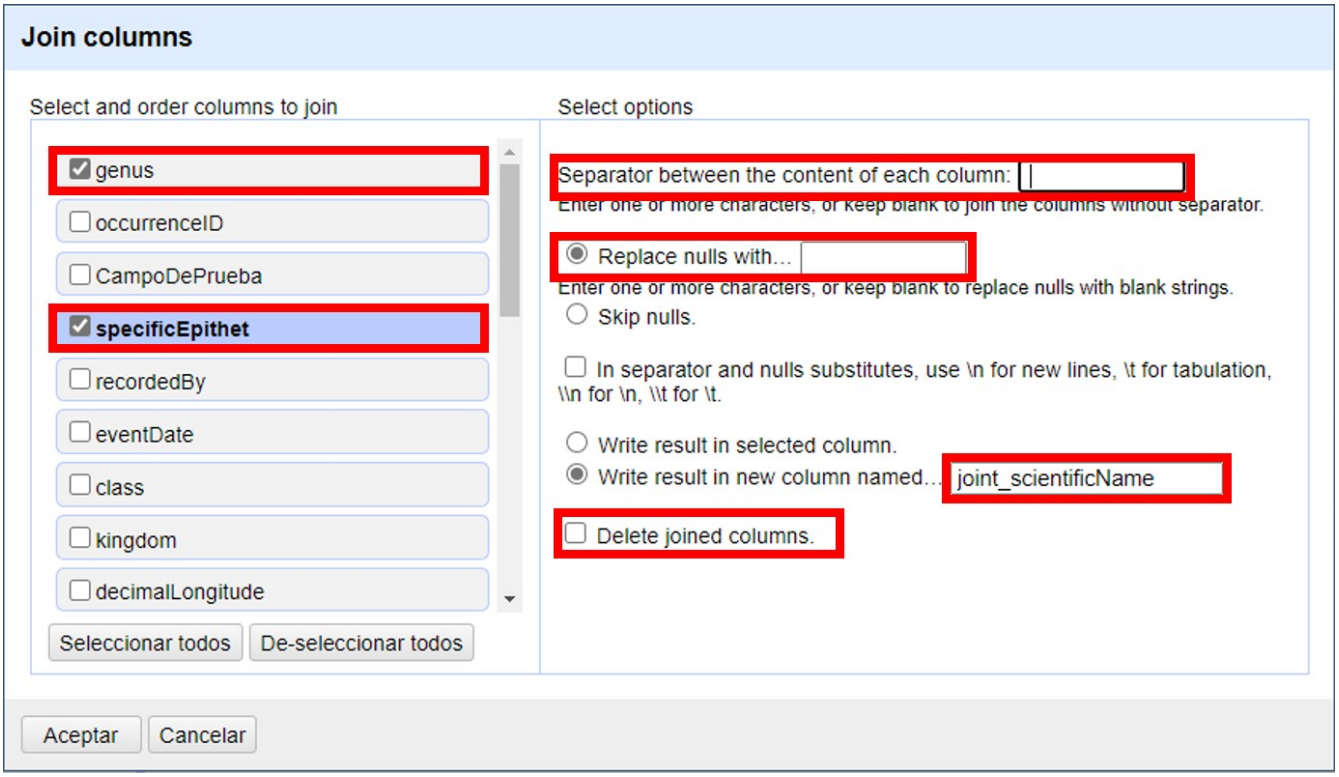

Figura 11

Los resultados esperados de acuerdo a distintos parámetros escogidos se resumen en la siguiente tabla (Tabla 1):

Tabla 1. Ejemplos de unión de dos columnas (“genus” y “specificEpithet”) en otra (“joint_scientificName), utilizando distintos separadores y tratamientos de nulos.

Separador	Tratamiento de nulos	genus	specificEpithet	joint_scientificName
" " (un espacio)	Reemplazar nulos con: "" (sin especificar)	Filago	lasiocarpa	Filago lasiocarpa
		Filago	*null*	Filago *(con un espacio extra después del género)*
		null	lasiocarpa	lasiocarpa
		null	*null*	*null*
	Reemplazar nulos con: "@"	Filago	lasiocarpa	Filago lasiocarpa
		Filago	*null*	Filago @
		null	lasiocarpa	@ lasiocarpa
		null	*null*	@ @
	Saltear nulos	Filago	lasiocarpa	Filago lasiocarpa
		Filago	*null*	*null*
		null	lasiocarpa	*null*
		null	*null*	*null*
", " (coma y espacio)	Reemplazar nulos con: "" (sin especificar)	Filago	lasiocarpa	Filago, lasiocarpa
		Filago	*null*	Filago, *(con un espacio extra después de la coma)*
		null	lasiocarpa	lasiocarpa
		null	*null*	*null*
	Reemplazar nulos con: "@"	Filago	lasiocarpa	Filago, lasiocarpa
		Filago	*null*	Filago, @
		null	lasiocarpa	@, lasiocarpa
		null	*null*	@, @
	Saltear nulos	Filago	lasiocarpa	Filago, lasiocarpa
		Filago	*null*	*null*
		null	lasiocarpa	*null*
		null	*null*	*null*

Si optamos por una opción que contiene en los resultados espacios en blanco no deseados, podemos aplicar luego una transformación en las celdas de la columna resultado del tipo "Quitar espacios al inicio y al final" (ver sección 2.2.2).

Ruta 2: Concatenación mediante expresiones regulares

Click en **la ▼ azul del campo** `"genus"` **› Editar columnas › Agregar columna basada en esta columna...**

Se abrirá una nueva ventana (Figura 12). Puede llamar al nuevo campo "concat_scientificName", para indicar que se trata de la concatenación (note que ya hay un campo `"scientificName"` en los datos).

En el cuadro de texto, pegue la siguiente expresión:

Expresión ejemplo: `cells["genus"].value + " " + cells["specificEpithet"].value` *(Expresión 1)*
Expresión general: `cells["campo1"].value + " " + cells["campo2"].value`

La expresión ejemplo concatena (`+`) los valores del campo `"genus"` (`cells["genus"].value`) y los del campo `"specificEpithet"` (`cells["specificEpithet"].value`), con un espacio entre los valores (`" "`).

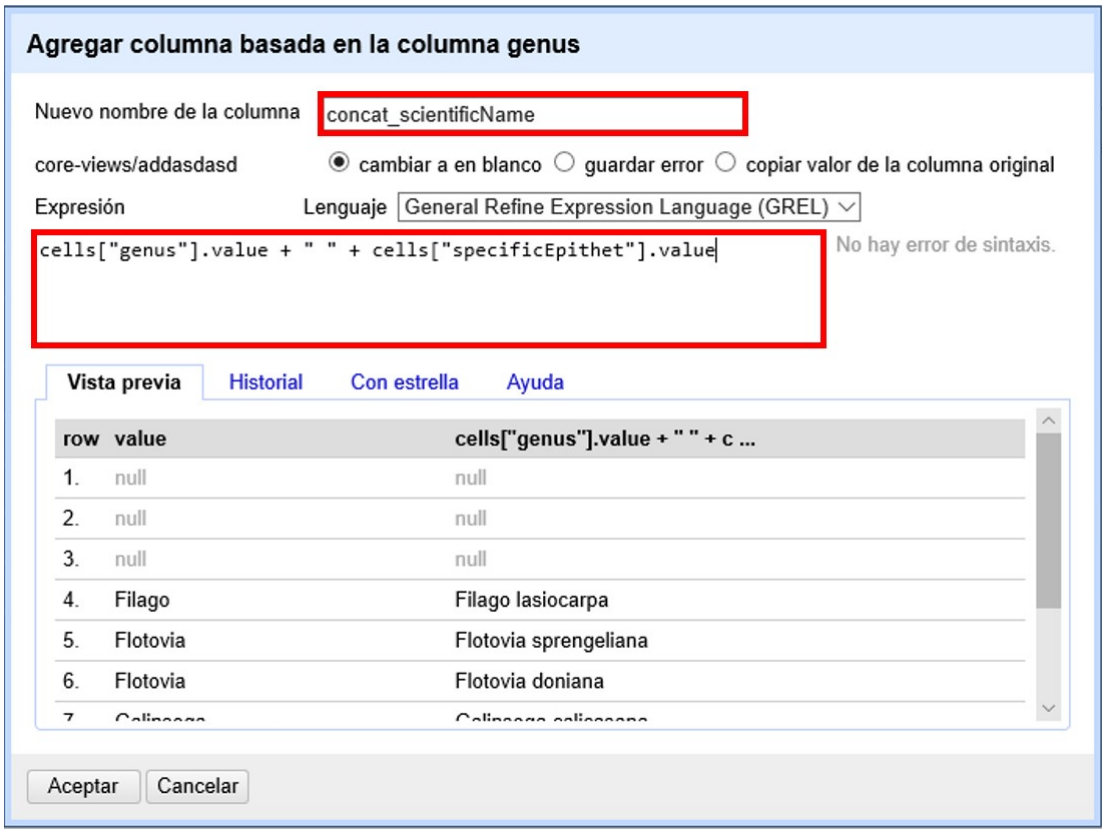

Figura 12

Note que esta expresión funciona cuando ambos campos, `"genus"` y `"specificEpithet"`, tienen valores, es decir no son nulos. Si alguno de los dos campos tiene valores nulos, entonces no se lleva a cabo la concatenación. Por ejemplo, si hay un valor para genus pero specificEpithet está vacío, el campo de concatenación aparecerá vacío. Esto se debe a que no se puede operar sobre valores nulos.

En este caso, puede sortear el problema utilizando en cambio la siguiente expresión:

```
if(isBlank(cells["genus"].value), "", cells["genus"].value) + " " + if(isBlank(cells[
"specificEpithet"].value), "", cells["specificEpithet"].value)
```

(Expresión 2)

Lo que dicha expresión significa es: concatenar (`+`) dos partes, cada una proviene de una sub-expresión `if`, separadas por un espacio (`+ " " +`). Cada una de estas sub-expresiones indica: si (`if`) el valor del campo dado es nulo (`isBlank(cells["genus"].value)`), colocar un blanco (`""`), si no (`,`), colocar el valor del campo (`cells["genus"].value`). La otra sub-expresión es lo mismo pero para epíteto específico.

Para evitar de modo más general este problema de celdas nulas, cuando importa el conjunto de datos para crear su proyecto al principio del proceso, puede asegurarse de NO seleccionar la opción “Store blank cells as nulls” (ver Figura 4).

La fórmula anterior (Expresión 2) resuelve el problema de tener valores nulos en la concatenación, pero al aplicarla, si alguno de los campos es nulo, el resultado tendrá espacios en blanco extra no deseados. Por ejemplo, si el valor de `"genus"` es nulo, el valor resultante en el campo concatenado será " epíteto", con un espacio en blanco antes del epíteto; si el valor de `"specificEpithet"` es nulo, el valor resultante será "genus ", con un espacio en blanco después del género; y si los valores de ambos son campos son nulos, el valor resultante será " ", un espacio en blanco. Para resolver este problema, se puede: 1) aplicar una transformación en las celdas de la columna resultado del tipo "Quitar espacios al inicio y al final" (ver sección 2.2.2), o 2) incluir en la expresión la quita de espacios al inicio y al final. Siguiendo la segunda opción, la expresión final sería:

```
Trim(if(isBlank(cells["genus"].value), "", cells["genus"].value) + " " + if(isBlank
(cells["specificEpithet"].value), "", cells["specificEpithet"].value))
```

(Expresión 3) donde se ha aplicado la función "Trim", que quita espacios en blanco no deseados al inicio y al final del valor de las celdas.

Los resultados esperados utilizando cada una de las tres fórmulas se resumen en la siguiente tabla (Tabla 2):

Tabla 2. Ejemplos de concatenación de dos columnas (“genus” y “specificEpithet”) en otra (“concat_scientificName), utilizando distintas expresiones (ver texto más arriba).

Expresión	genus	specificEpithet	concat_scientificName
1	Filago	lasiocarpa	Filago lasiocarpa
	Filago	*null*	*null*
	null	lasiocarpa	*null*
	null	*null*	*null*

Expresión	genus	specificEpithet	concat_scientificName
2	Filago	lasiocarpa	Filago lasiocarpa
	Filago	*null*	Filago *(con un espacio en blanco después del género)*
	null	lasiocarpa	lasiocarpa *(con un espacio en blanco antes del epíteto)*
	null	*null*	*(con un espacio en blanco)*
3	Filago	lasiocarpa	Filago lasiocarpa
	Filago	*null*	Filago
	null	lasiocarpa	lasiocarpa
	null	*null*	*null*

Divisiones

Si desea crear campos separados a partir de los valores en un único campo, siga la siguiente ruta:

Utilizaremos como ejemplo la división del campo `"eventDate"` para agregar tres campos: año, mes y día (year, month y day)

Click en **la ▼ azul del campo `"eventDate"` › Editar columnas › Dividir en varias columnas...**

Se abrirá una nueva ventana (Figura 13). Allí debe escoger si se dividirá por separador o por longitud de caracteres, y en el primer caso qué tipo de separador se utilizará (puede ser espacio, caracter de tabulación, coma, punto y coma, guión, etc.).

En este caso, si exploramos los datos del campo original veremos que año, mes y día están separados por barras oblicuas (“/”), de modo que elegiremos esta barra como separador.

Desmarque la opción “Eliminar esta columna” a la derecha. Si la deja seleccionada, perderá el campo original y sólo tendrá los tres nuevos campos.

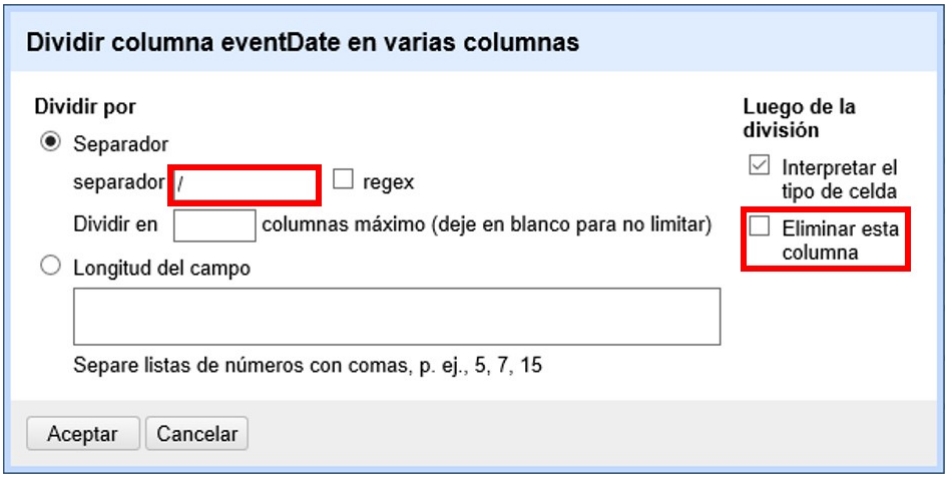

Figura 13

Una vez que oprima Aceptar, se crearán las nuevas columnas a la derecha del campo `"eventDate"`. OpenRefine las nombra automáticamente agregando números al final del nombre (en este caso: eventDate1, eventDate2 y eventDate3). Cambie los nombres de las columnas por los que corresponda (**la ▼ azul › Editar columnas › Renombrar esta columna**). En este caso, nómbrelos "year", "month" y "day" según corresponda.

Cuando efectúe este tipo de divisiones de campos utilizando como criterio ya sea separadores o longitud de caracteres, asegúrese de que en el campo original no haya distintos formatos para diferentes registros. Vea el siguiente ejemplo:

Se quiere separar un campo nombrado "coordenadas" que contiene datos de latitud y longitud separados por coma, del tipo: "-32.04588990, -54.98789901", para obtener dos campos distintos, latitud y longitud.

Si todos los campos tienen el mismo formato, obtendrá dos campos nuevos de la siguiente forma:

```
campo 1: -32.04588990
campo 2: -54.98789901
```

En cambio, si en algún registro los valores dentro del campo coordenadas no están en formato decimal, entonces tendrá problemas al dividir el campo. Suponga como ejemplo que uno o más registros tienen valores con formato "34° 20' 15,2" S, 54° 49' 13" O". En ese caso, la separación le dará 3 campos en vez de dos, con la latitud incorrectamente separada:

```
campo 1: 34° 20' 15
campo 2: 2'' S
campo 3: 54° 49´ 13´´ O
```

2.2. Uso de Facetas

La función “Facetas” es una forma de visualización de los datos, que permite el tratamiento en bloque de grupos de registros. Las facetas se pueden aplicar a celdas que contengan cualquier tipo de texto, números o fechas.

2.2.1. Facetas de texto

Ubique la columna `"kingdom"` y haga click sobre **la ▼ azul**. Dentro de “Facetas”, escoja “Faceta de texto”, como se muestra a continuación (Figura 14a). Se abrirá entonces a la izquierda una ventana con la faceta (Figura 14b).

Figura 14

En dicha ventana de faceta, puede ordenar los valores alfabéticamente (haciendo click sobre “A-Z”) o según el número de registros asociados a cada valor (haciendo click sobre “conteo”).

En la lista de valores podemos ver que hay algunos errores. Para corregirlos coloque el cursor sobre el valor que desea modificar y haga click en “editar”. Se abrirá entonces una pequeña ventana donde puede cambiar el valor (Figura 15). Para guardar el cambio haga click en “Aplicar”, ello aplicará el cambio a todos aquellos registros que tenían el valor dado.

Corrija los valores “Plante” y “Plants”. Cuando lo haga, habrá corregido todos los registros que contenían esos valores, y se modificará entonces el número de registros que tiene el valor “Plantae”.

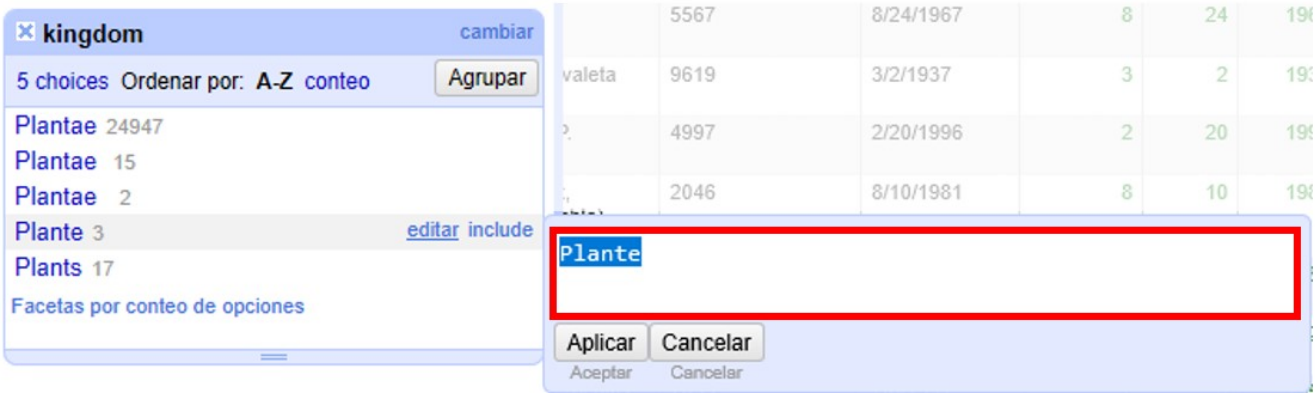

Figura 15

2.2.2. Facetas y espacios en blanco

Espacios en blanco extra al principio o al final de una cadena de texto

Una vez que haya corregido los valores en el punto anterior, notará que aún aparecen 3 valores “Plantae”, aparentemente iguales (Figura 16). Sin embargo, estos valores sí son diferentes: tienen espacios adicionales al final del valor de texto.

Figura 16

Para corregir estos errores, asegúrese de que ninguno de los valores en la faceta están seleccionados y de que el número de registros que se muestra arriba de la tabla es el total (24984). Sobre la columna `"kingdom"`, haga click sobre **la ▼ azul** y siga las siguientes opciones (Figura 17):

Editar celdas › Transformaciones comunes › Quitar espacios al inicio y final

Esta función permite eliminar espacios en blanco que puedan aparecer al principio y al final de cadenas de texto. Cuando termine este paso, los 24,984 registros deberían tener el valor “Plantae” en la columna `"kingdom"`.

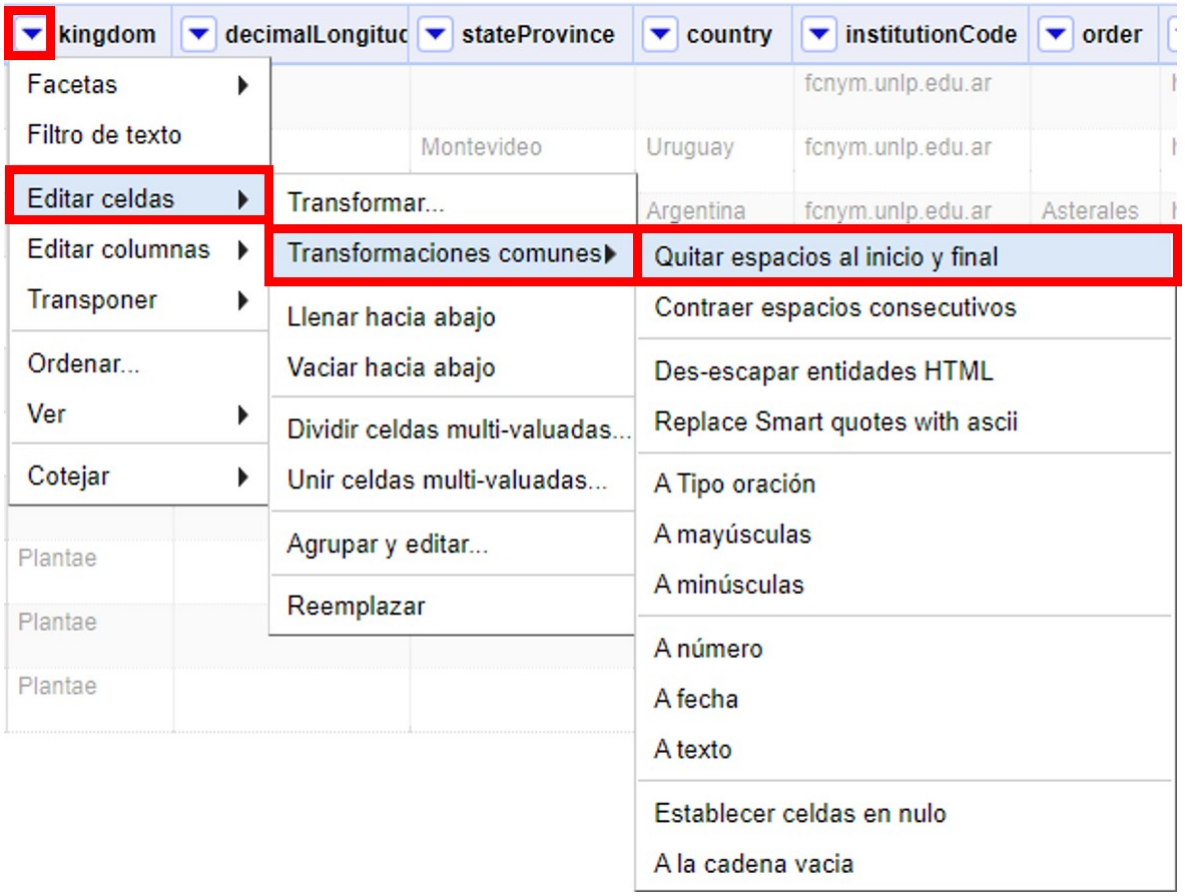

Figura 17

Espacios en blanco extra entre palabras en una cadena de texto

A veces en campos que contienen cadenas de texto con varias palabras puede haber espacios en blanco extra entre palabras. Para ver un ejemplo, ubique la columna `"stateProvince"` en el conjunto de datos. Arme una faceta de texto para dicha columna (click sobre **la ▼ azul › Facetas › Faceta de texto**). Luego, en la faceta, ordene los valores por número de registros asociados (seleccionando "conteo"). Verá entonces los valores que se encuentran en este campo como se muestra en la Figura 18.

Note que en primer y tercer lugar figura aparentemente el mismo valor, "Buenos Aires". La diferencia entre ambos valores es que uno de ellos tiene un doble espacio entre las palabras.

Figura 18

Para corregir este error, sobre la columna `"stateProvince"`, haga click sobre **la ▼ azul** y siga la siguiente ruta (Figura 19):

Editar celdas › Transformaciones Comunes › Contraer espacios consecutivos

Esta función le permite convertir múltiples espacios en blanco en un único espacio en blanco.

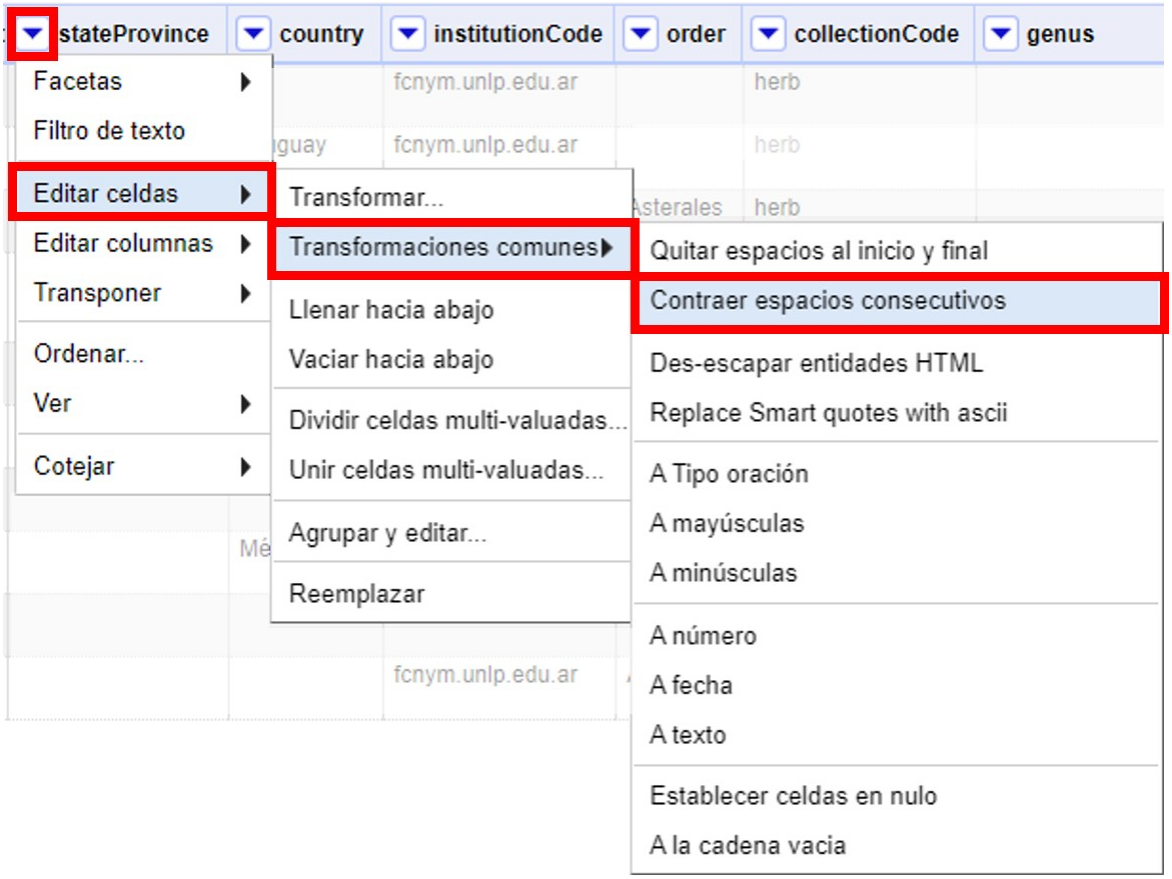

Figura 19

Una vez que haya removido los espacios en blanco extra, en la faceta sólo verá un valor para “Buenos Aires”, con un número de registros que es la suma de los valores anteriores. Tenga en cuenta que si había otros valores con el mismo problema de dobles espacios entre palabras en esta

misma columna, la modificación se aplicará a todos ellos, y no sólo a “Buenos Aires”. Puede comprobar cuántos valores se han modificado comparando el número de valores disponibles en la faceta antes y después de la transformación.

Espacios en blanco extra en todo el conjunto de datos

Habiendo visto cómo eliminar espacios en blanco extra, al principio, final o en medio de una cadena de texto, en campos determinados, existe una manera de realizar todas estas acciones al mismo tiempo sobre todos los campos de conjunto de datos. Para ello, se debe ir a la columna “Todo”, hacer click sobre "la ▼ azul > Transformar.

Se abrirá entonces una ventana, y en el cuadro de texto debe pegarse la siguiente expresión:

```
value.trim().replace(/\s+/,' ')
```

Al hacer click en “Aceptar” se eliminarán los espacios en blanco extra en todo el conjunto de datos. Los cambios serán registrados columna a columna en el historial de cambios (ver más abajo sección Deshacer y rehacer cambios).

2.2.3. Inclusión y exclusión de registros usando facetas

Inclusión de registros con valores determinados para un campo dado

Las facetas pueden utilizarse para trabajar sobre registros con uno o más valores de interés en un campo en cuestión. Para trabajar sobre un ejemplo, arme una faceta de texto sobre el campo `"phylum"` (click sobre **la ▼ azul › Facetas › Faceta de texto**). Verá que la faceta tiene varios valores. Para seleccionar solo los registros que tienen como phylum, por ejemplo, “Lycopodiophyta”, debe hacer click sobre el valor mismo dentro de la faceta o sobre la opción “include” que se muestra a su derecha (Figura 20a).

Figura 20

Al seleccionar un valor dentro de una faceta, cualquier acción que tome a continuación sólo será aplicada a los registros incluidos bajo esa selección.

Puede seleccionar tantos valores como desee dentro de una faceta, utilizando “include” sucesivamente sobre cada uno de ellos.

Exclusión de registros con valores determinados para un campo dado

Para deseleccionar registros previamente seleccionados a través de una faceta, simplemente haga click sobre el valor nuevamente, o sobre la opción “exclude” que se muestra a su derecha (Figura 20b). Puede deseleccionar tantos valores como desee utilizando “exclude” sucesivamente sobre cada uno de ellos.

En ocasiones las facetas pueden contener muchos valores de interés diferentes sobre los que quisiéramos trabajar. En estos casos, puede ser muy engorroso seleccionar todos los valores de interés uno a uno. En cambio, se puede utilizar la función “invertir” selección. Para aplicar esta función, deben seleccionarse los valores que *no* son de interés. Una vez seleccionados, en la parte superior de la faceta aparece la opción “invertir” (Figura 20b). Haciendo click el programa nos brindará la selección inversa, incluyendo entonces los valores que *sí* nos interesan.

Por ejemplo, para el campo `"phylum"` del ejemplo anterior, nos interesan todos los valores menos “Pinophyta”. Seleccionamos entonces “Pinophyta” haciendo click en “include” para este valor. Luego hacemos click en “invertir”, y como resultado habremos seleccionado todos los registros salvo aquellos que tienen el valor “Pinophyta” (Figura 20c).

Para revertir la inversión puede hacerse click en “invertir” nuevamente, volviendo entonces a los valores seleccionados inicialmente.

Selección de registros sin valores para un campo dado

En muchas ocasiones resulta muy útil poder identificar los registros que tienen un campo de interés vacío, sin valores. Utilizando facetas se pueden reconocer esos registros fácilmente, pues figuran dentro de la faceta con valor “(blank)” (ver por ejemplo este valor en la faceta compuesta para el apartado anterior, sobre el campo `"phylum"`, Figura 20) . El valor “(blank)” se puede tratar como cualquier otro dentro de la faceta, es decir, se puede incluir, excluir, y editar, facilitando la evaluación y el mejoramiento de los registros.

2.2.4. Facetas numéricas

Las facetas también pueden aplicarse a campos numéricos, y en ese caso son muy útiles para, por ejemplo, detectar valores fuera de rangos de interés.

A modo de ejemplo, armaremos una faceta numérica sobre el campo `"day"` que hemos creado más arriba. Para ello, hacer click en **la ▼ azul** del campo y seguir la ruta:

Facetas › Faceta numérica

Verá entonces una nueva ventana, la faceta, como se muestra en la Figura 21.

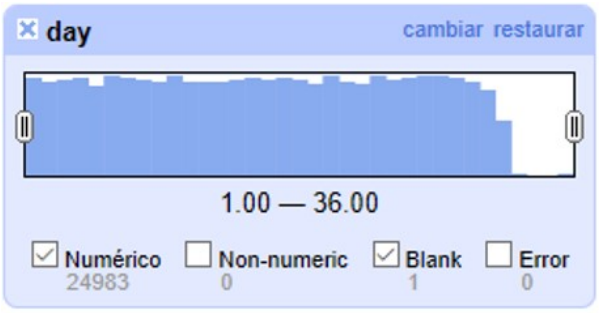

Figura 21

Allí se puede ver que el rango de días abarca desde 1 a 35 inclusive. Es decir, algunos números están fuera de rango, puesto que como máximo puede haber hasta día 31 en algunos meses.

Se pueden seleccionar los registros con los valores superiores desplazando el botón a la izquierda del rango hacia la derecha. Ello incluirá en la tabla los registros por encima del rango seleccionado y, si no desmarca la opción "Blank", también los blancos, como se muestra en la Figura 22 (en el ejemplo, tres filas en total: un caso con día 32, un caso con día 35 y un caso con día vacío). Si hubiera valores en el campo que no son numéricos, también podría verlos utilizando esta faceta.

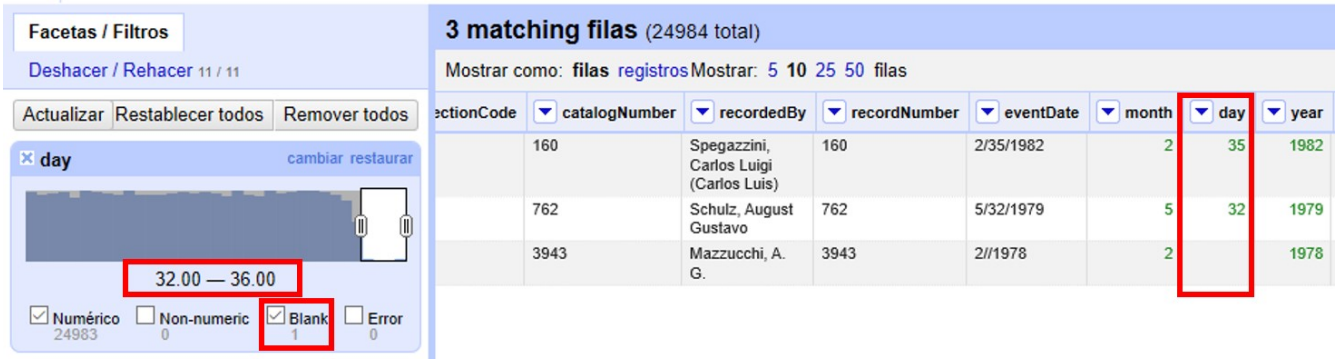

Figura 22

Los tres errores encontrados deben ser consultados con la información original de los ejemplares en la colección, y los campos de fecha estrictamente deberían quedar vacíos para estos registros. Una opción es marcar estos registros para revisar más adelante, usando estrellas o banderas (ver sección sobre uso de estrellas y banderas).

Si el campo sobre el que desea armar la faceta no es un campo con formato numérico (e.g., tiene formato texto, o fecha, etc.), la faceta numérica no le mostrará valores. En cambio, dirá que el campo no tenía valores numéricos ("No numeric value present."). Para poder armar una faceta numérica tendrá entonces primero que transformar los datos de la columna de interés a formato numérico. Para ello, siga la ruta: click sobre **la ▼ azul del campo › Editar celdas › Transformaciones comunes › A número**.

2.2.5. Facetas y duplicados

Las facetas también permiten la detección y corrección de duplicados.

Cuando hablamos aquí de duplicados, nos referimos a valores duplicados dentro de una columna, no necesariamente a registros enteros duplicados, o a duplicados en el sentido biológico/de colecciones. Por ello, tenga especial cuidado a la hora de actuar sobre estos valores duplicados, pues podrían tener efectos a diferentes niveles.

Veremos un ejemplo de duplicados en la columna `"catalogNumber"`. Para ello, haga click en **la ▼ azul** y luego siga la siguiente ruta:

Facetas › Facetas personalizadas › Faceta por duplicados

Verá entonces una ventana con la faceta, como se muestra en la Figura 23, donde “true” (“verdadero”) refiere a los valores duplicados.

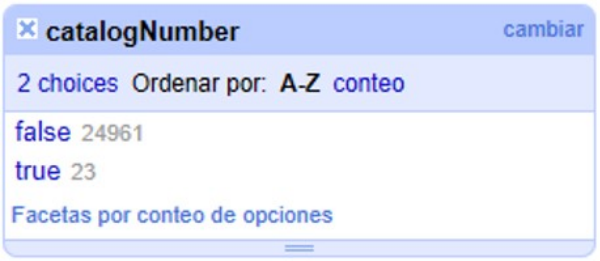

Figura 23

Si hace click en “true”, la pantalla principal le mostrará los registros que tienen número de catálogo duplicado (Figura 24). Observe por ejemplo los siguientes registros:

- el primer y quinto registros tienen el mismo número de catálogo, 5567
- el tercer registro (y otros más abajo que no son visibles entre los 25 primeros) no tiene número de catálogo (el valor nulo es lo que está duplicado).
- etc.

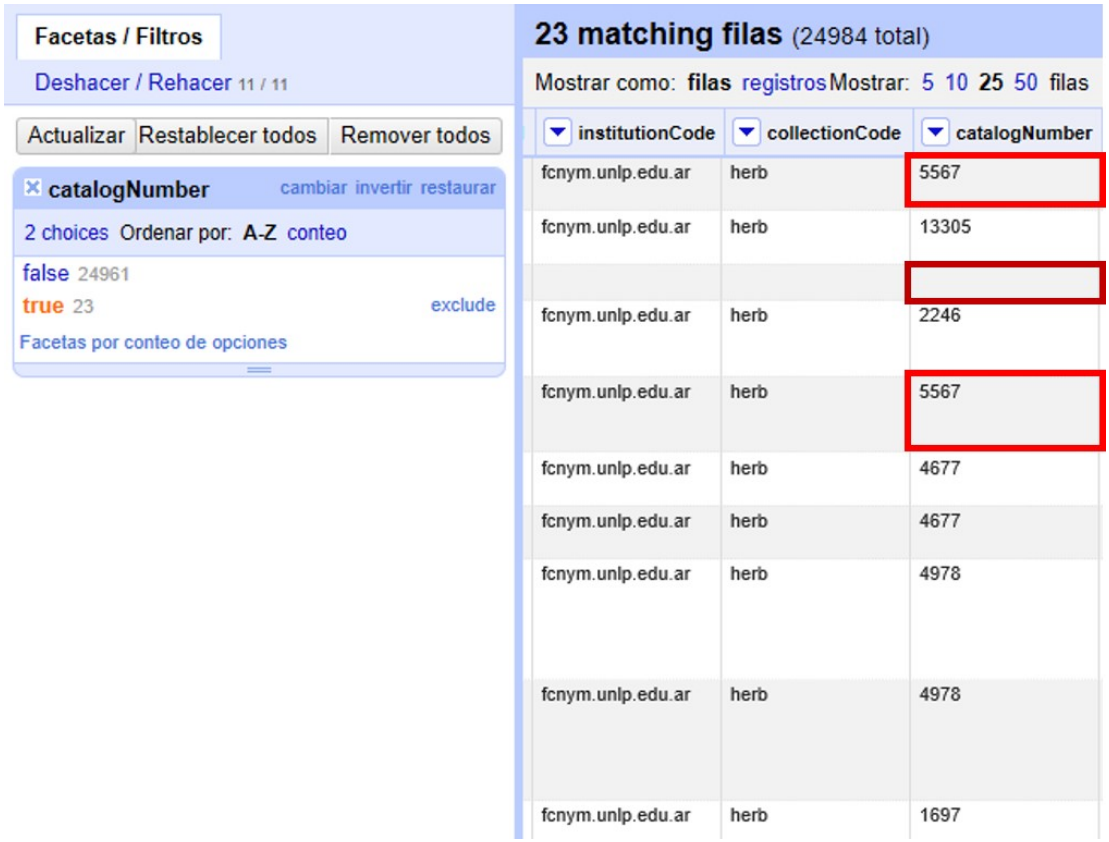

Figura 24

Corrija los números de catálogo. Para hacerlo, edite las celdas individualmente: sobre la celda haga click en el botón “editar”, modifique el valor y haga click en “Aplicar” (Figura 25).

En la práctica la corrección de los números de catálogo sólo debe hacerse una vez que los números y los datos asociados han sido comprobados con las etiquetas de los especímenes.

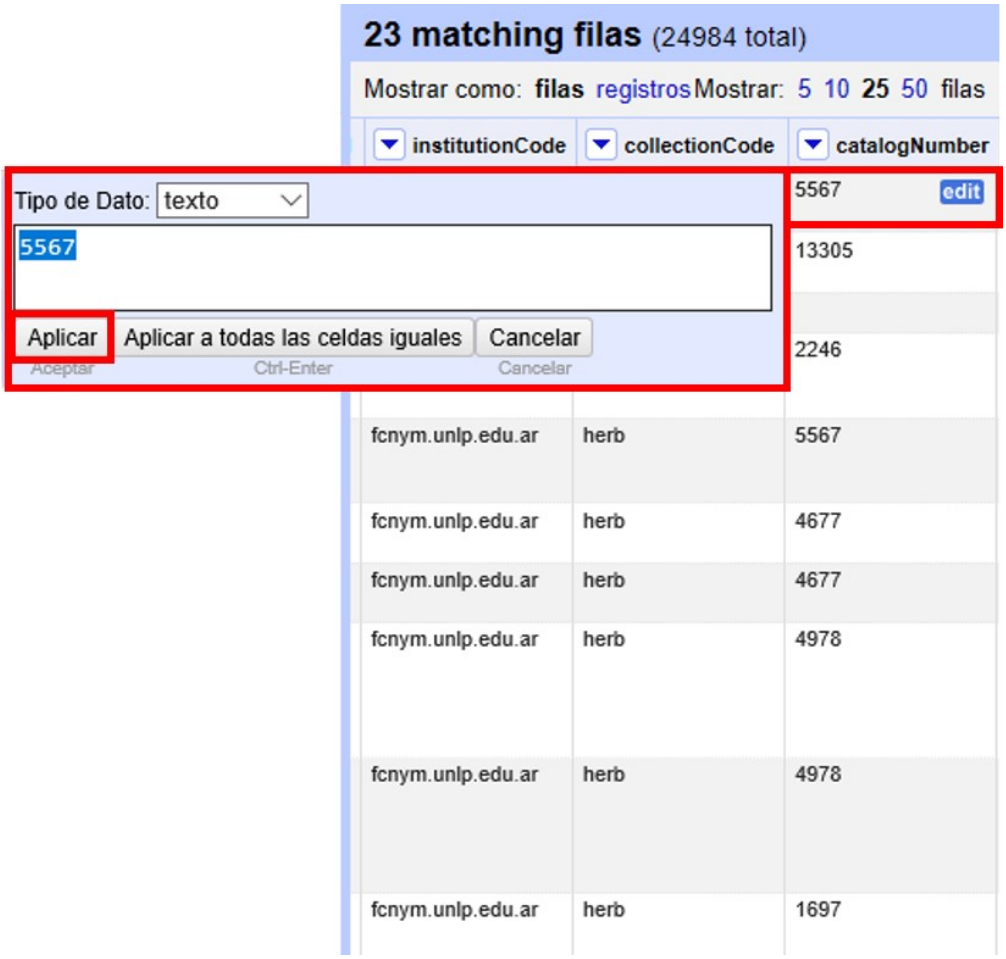

Figura 25

2.2.6. Límite en el número de opciones de las Facetas

En OpenRefine existe un límite para el número de elecciones de faceta que se muestran (“choices”). Muchas veces dicho número está pre-configurado a un valor de 2000. Ello quiere decir que sólo podrá ver 2000 opciones dentro de la faceta de interés.

Por ejemplo, si tiene configurado el valor a 2000 y trata de armar una faceta de texto en el campo `"specificEpithet"`, verá que a la derecha la faceta no muestra los valores esperados sino un mensaje que dice que hay demasiados valores para mostrar (Figura 26a).

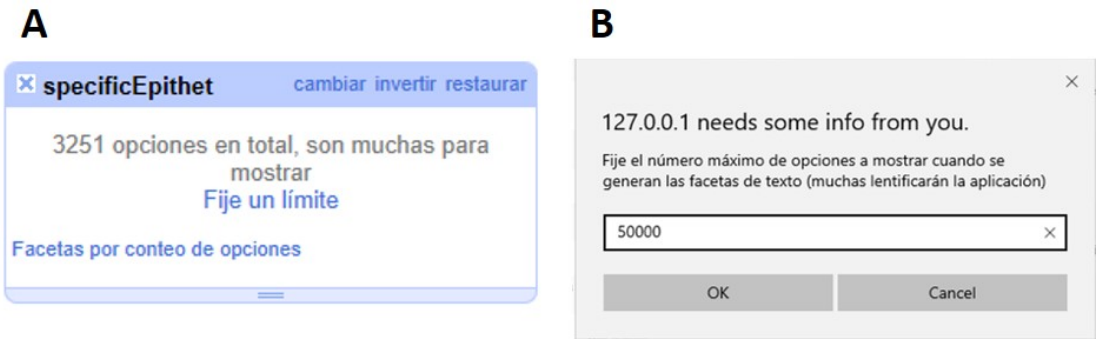

Figura 26

Haciendo click en “Fije un límite”, se abrirá otra ventana donde puede cambiar el límite al valor preferido (Figura 26b).

Una vez que haya cambiado el valor límite, y si este valor es lo suficientemente grande, podrá ver todos los valores en la faceta del campo de interés (en el ejemplo anterior, el campo `"specificEpithet"`).

Alternativamente, para modificar en cualquier momento el límite en el número de valores que se pueden desplegar por faceta, puede ir a la siguiente dirección en su navegador web:

http://127.0.0.1:3333/preferences

El navegador mostrará una ventana con ciertas opciones (Figura 27a). Allí, establezca el límite preferido para las facetas editando la clave "ui.browsing.listFacet.limit". Para ello haga click en "core-index/edit", y en la ventana que se abre, coloque el nuevo valor límite y oprima "OK" (Figura 27b).

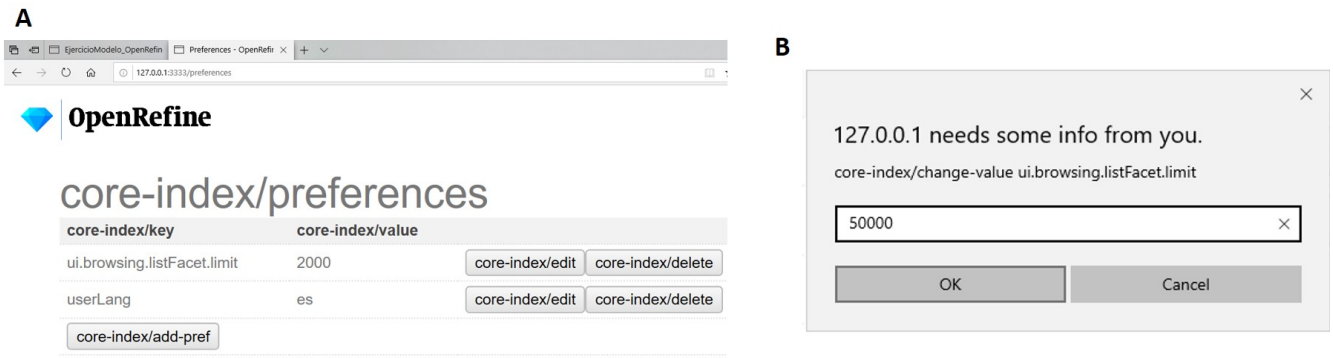

Figura 27

2.3. Uso de Filtros

2.3.1. Filtros simples

OpenRefine permite el uso de filtros sobre campos particulares, función que puede ser muy útil para la limpieza de datos. Veremos un ejemplo a continuación.

Ubique el campo `"specificEpithet"` y cree una faceta de texto (haga click en **la ▼ azul › Facetas › Faceta de texto**). Luego vaya nuevamente a **la ▼ azul** y cree un filtro de texto ("Filtro de texto"). Sobre el menú de la izquierda se abrirá una ventana como la que se muestra en la Figura 28.

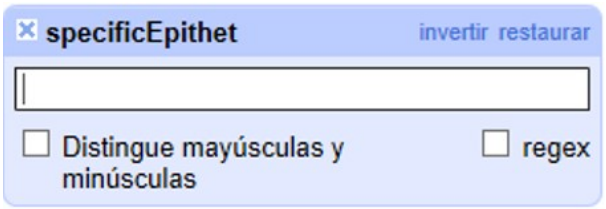

Figura 28

En el cuadro de texto puede escribir el valor sobre el cual desea filtrar.

Por ejemplo, pruebe escribiendo "sp.".

En el menú de la izquierda, dentro de la faceta se mostrará el valor que usted buscó, y en la pantalla principal se mostrarán los registros asociados que tienen dicho valor en el campo `"specificEpithet"` (Figura 29).

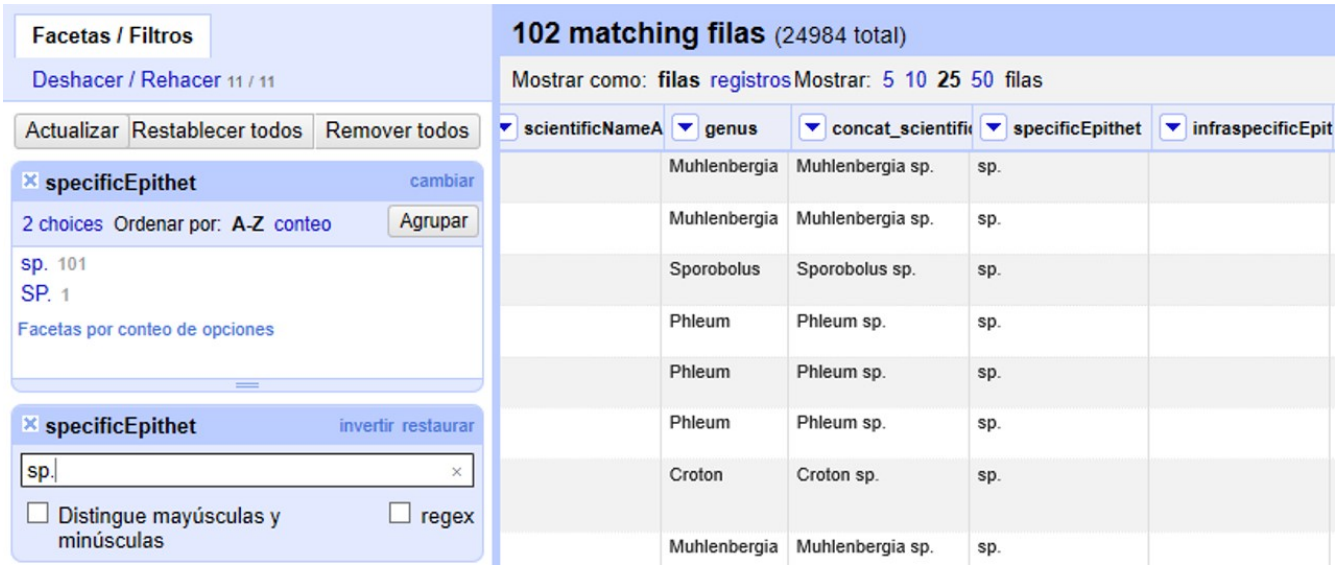

Figura 29

Note que verá dos valores, uno en letras minúsculas y otro en letras mayúsculas. Si sólo desea ver los valores escritos con minúscula, en el filtro debe seleccionar “Distingue mayúsculas y minúsculas”, o puede seleccionar “sp.” directamente sobre la faceta de `"specificEpithet"`.

Corrija los valores “sp.” y “SP.” utilizando la función “editar” sobre los valores en la faceta (el valor correcto debería ser nulo).

Cierre el filtro y la faceta de `"specificEpithet"`.

Abra una faceta de texto y un filtro para el campo `"scientificName"`. En el filtro, busque el valor “sp.”. Verá entonces varios valores para ese campo que incluyen “sp.”, como se muestra en la Figura 30.

Figura 30

Debe corregir esos nombres, sacando “sp.” y dejando solamente el nombre del género. Para no tener que hacerlo uno por uno, puede seguir los siguientes pasos.

Haga click sobre **la ▼ azul en** `"scientificName"` **› Editar celdas › Transformar...** (Figura 31).

Figura 31

Se abrirá entonces una ventana como la mostrada en la (Figura 32). En el cuadro de texto, pegue la siguiente expresión:

```
value.replace(" sp.", "")
```

Dicha expresión tiene la función de reemplazar lo que está entre las primeras comillas por aquello que está entre las segundas comillas, es decir, la porción `" sp."` (`[espacio]sp.`) por `""` (nada).

En la Figura 32 puede observar cómo se vería el resultado del cambio en la pestaña “Vista previa”.

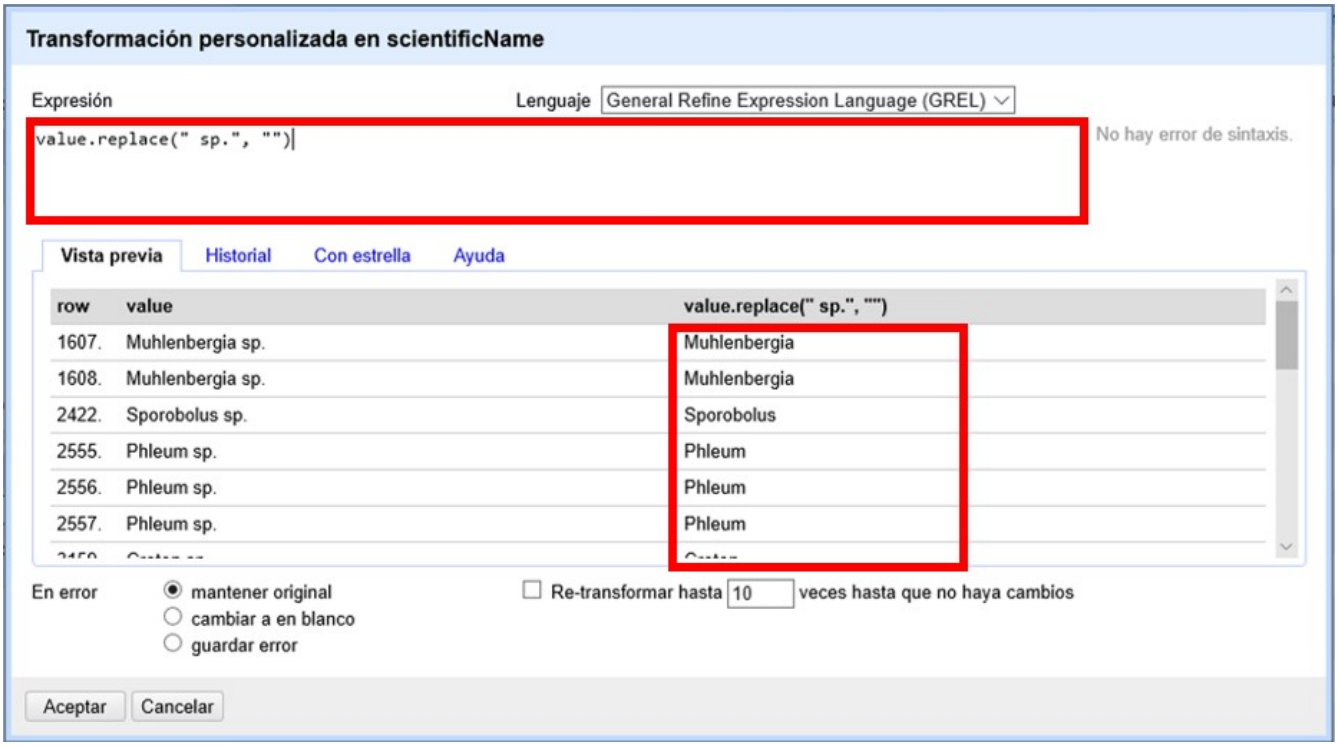

Figura 32

Oprima “Aceptar” para ejecutar la transformación, y verá que en la faceta que ha sido filtrada ya no hay registros que contengan “sp.” como parte del valor en el campo `"scientificName"`.

Cierre la faceta y el filtro del campo `"scientificName"`.

2.3.2. Filtros con expresiones regulares

Los filtros se pueden utilizar también incluyendo expresiones regulares, que permiten buscar ciertos patrones en los valores de los campos. Por ejemplo, se pueden buscar palabras que comiencen con ciertas letras, o que comiencen con mayúscula o minúscula, etc.

A modo de ejemplo, buscaremos valores en el campo `"genus"` que comiencen con minúscula. Para ello, abra una faceta y un filtro de texto para el campo `"genus"`. En el filtro coloque la siguiente expresión en el cuadro de texto: `^[a-z]`, y seleccione las opciones “Distingue mayúsculas y minúsculas” y “regex” (Figura 33a). Con dicha expresión se pueden buscar los valores en los que la primera letra es minúscula.

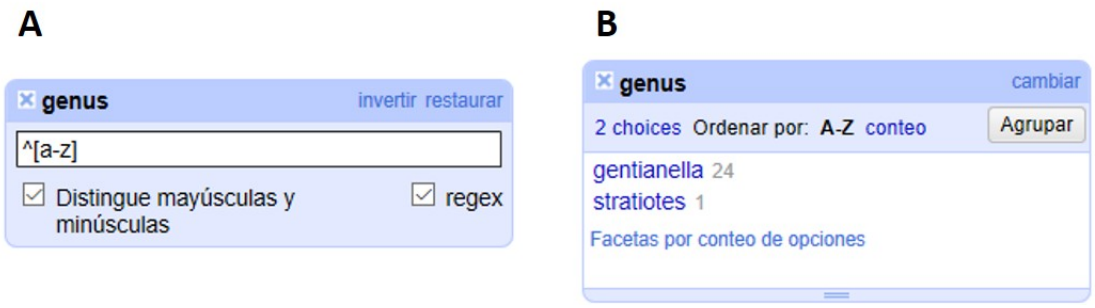

Figura 33

Siguiendo estos pasos, debería poder ver dos valores (Figura 33b). Corrija estos valores filtrados, dado que el género debe comenzar con mayúscula.

OpenRefine acepta un lenguaje de expresiones regulares Java, que puede consultar aquí: http://docs.oracle.com/javase/tutorial/essential/regex/. Algunas expresiones que pueden ser útiles como filtros para diversos campos son:

- `^[A-C]`

  Busca las cadenas de texto que comienzan (`^`) con mayúscula de la A a la C (`[A-C]`)

- `^[^a-d]`

  Busca las cadenas de texto que comienzan (`^`) con cualquier carácter en minúscula salvo de la a a la d (`[^a-d]`) – el `^` dentro del `[]` indica negación.

- `^\w`

  Busca las cadenas de texto alfanuméricas que comienzan (`^`) con un número o una letra (`\w`) –de 0 a 9 o de la a a la z, mayúscula o minúscula, o el carácter '_'.

- `^\s`

  Busca las cadenas de texto que comienzan (`^`) con un espacio en blanco (`\s`).

- `^\d`

  Busca las cadenas de texto que comienzan (`^`) con un dígito (`\d`).

- `^\D`

  Busca las cadenas de texto que comienzan (`^`) con un carácter no dígito (`\D`). Equivalente a la expresión con negación `^[^0-9]`.

- `\d{4}`

  Busca cadenas de texto que contengan dígitos (`\d`), en particular 4 dígitos (`{4}`).

- `^\w.*\d$`

  Busca las cadenas de texto que comiencen (`^`) un carácter alfanumérico o el carácter '_' (`\w`), sigan (`.`) con cualquier carácter (`*`) y terminen (`$`) con un dígito (`\d`).

- `^[A-Z].*\s[A-Z]`

  Busca las cadenas de texto que comienzan (`^`) con mayúscula (`[A-Z]`) –cualquier mayúscula de la A a la Z– seguidas de (`.`) cualquier carácter (`*`), luego un espacio (`\s`), luego otra letra mayúscula (`[A-Z]`).

- `[À-ÖØ-öø-ÿ]`

  Busca las cadenas de texto que contengan al menos un carácter especial ([À-ÖØ-öø-ÿ]) –dentro de los rangos de caracteres Unicode À-Ö o Ø-ö o ø-ÿ, por ejemplo caracteres con diacríticos (acentos graves, agudos, circunflejos, diéresis, virguillas, cedilla, etc.)

Pruebe el uso de algunas de esas expresiones en distintos campos.

Para más ejemplos y usos, puede consultar el repositorio de OpenRefine en GitHub.

2.4. Uso de Agrupamientos

2.4.1. Agrupamientos simples

Los agrupamientos permiten, como su nombre lo indica, agrupar valores de acuerdo a diferentes criterios. Por ejemplo, pueden agruparse valores de acuerdo al grado de similitud en cuanto a las letras que los componen o en cuanto a la fonética asociada. Esta función es muy útil para detectar y corregir errores de ortografía y variaciones en los datos.

Ubique el campo `"stateProvince"` y arme una faceta de texto para este campo.

En la ventana de la faceta, haga click en el botón “Agrupar”. Se abrirá entonces una ventana como la mostrada en la Figura 34.

Allí verá que algunos valores que son similares han sido agrupados por un algoritmo. El método y la función utilizados se muestran y se pueden modificar arriba de la lista de valores.

La ventana también muestra el tamaño del clúster (“Número de valores” agrupados), cuántos registros hay por cluster (“Número de filas”) y por valor (entre paréntesis junto a los valores en “Valores en la agrupación”).

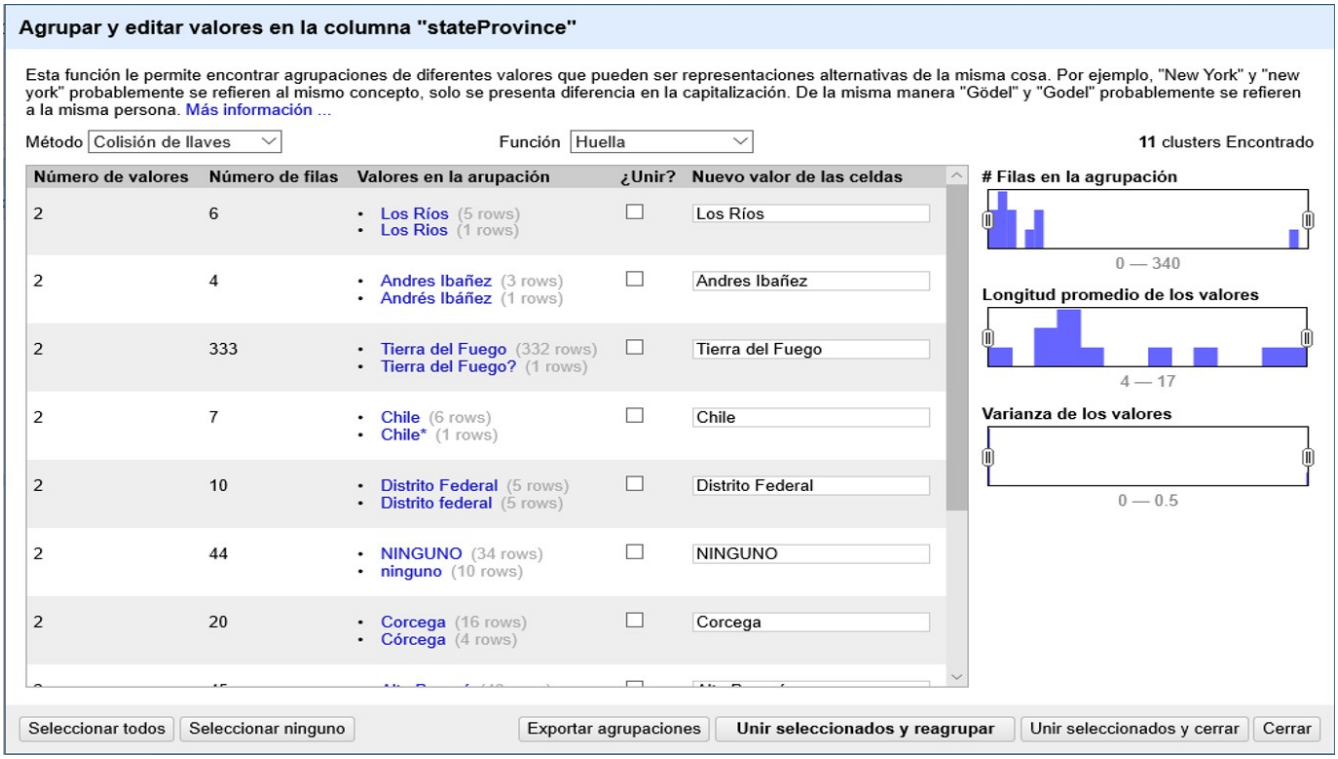

Figura 34

Además, para cada cluster verá una opción para fusionar los valores (“¿Unir?”) y un recuadro donde se captura el nuevo valor que se asignará a todos los registros del cluster (“Nuevo valor de las celdas”).

OpenRefine asigna de forma predeterminada como nuevo valor aquel que presenta mayor número de registros asociados. Esto no es necesariamente correcto. Por ejemplo, en el caso “Corcega” y “Córcega” el valor correcto lleva tilde. Puede modificar el nuevo valor al que se unificará el agrupamiento haciendo click en el valor deseado si está listado o, en caso de ser diferente, editando directamente el recuadro “Nuevo valor de las celdas”. Recuerde que todos los valores dentro de un agrupamiento dado se unificarán al valor escogido. Por ejemplo, en el caso de que los valores agrupados sean “NINGUNO” y “ninguno”, podría agrupar a un nuevo valor vacío (pues “ninguno” no es un valor válido para una provincia).

Explore los valores agrupados por el algoritmo y corrija los que considere apropiados, seleccionando el valor correcto y marcando la casilla “¿Unir?”.

Haga click sobre “Unir seleccionados y reagrupar”.

Cuando se agrupan valores se debe tener mucho cuidado a la hora de corregir registros. Esto es particularmente cierto para los nombres científicos, dado que variaciones en los nombres que podrían verse como aparentes errores (por ejemplo, si se evalúa el campo epíteto específico, pueden tenerse dos palabras iguales con diferente terminación –um, –us), no necesariamente lo sean (por ejemplo, si se evalúa también el campo género podría encontrarse que esos epítetos se aplican a géneros distintos, y que ambos son válidos). Por ello, si tiene dudas, consulte los registros completos. Y si aún tiene dudas, consulte en la colección. Otro ejemplo en que debe tenerse extremo cuidado es cuando se agrupan valores que difieren en el orden de las palabras. Un ejemplo típico se da en el campo de colectores. Aún cuando los agrupamientos pueden sugerir que “Colector A y Colector B” es lo mismo que “Colector B y Colector A”, ello puede no ser cierto, y el orden de los colectores puede tener en sí un valor particular. Nuevamente, antes de unificar, es fundamental consultar con la colección.

Una vez resueltos los agrupamientos, si ha decidido no agrupar algunas de las opciones, las verá nuevamente en el re-agrupamiento; en caso contrario, el programa le indicará que no se han encontrado agrupaciones con el método seleccionado. Puede cambiar el método y la función que se utiliza para agrupar escogiendo entre las opciones del menú, como se muestra en la Figura 35.

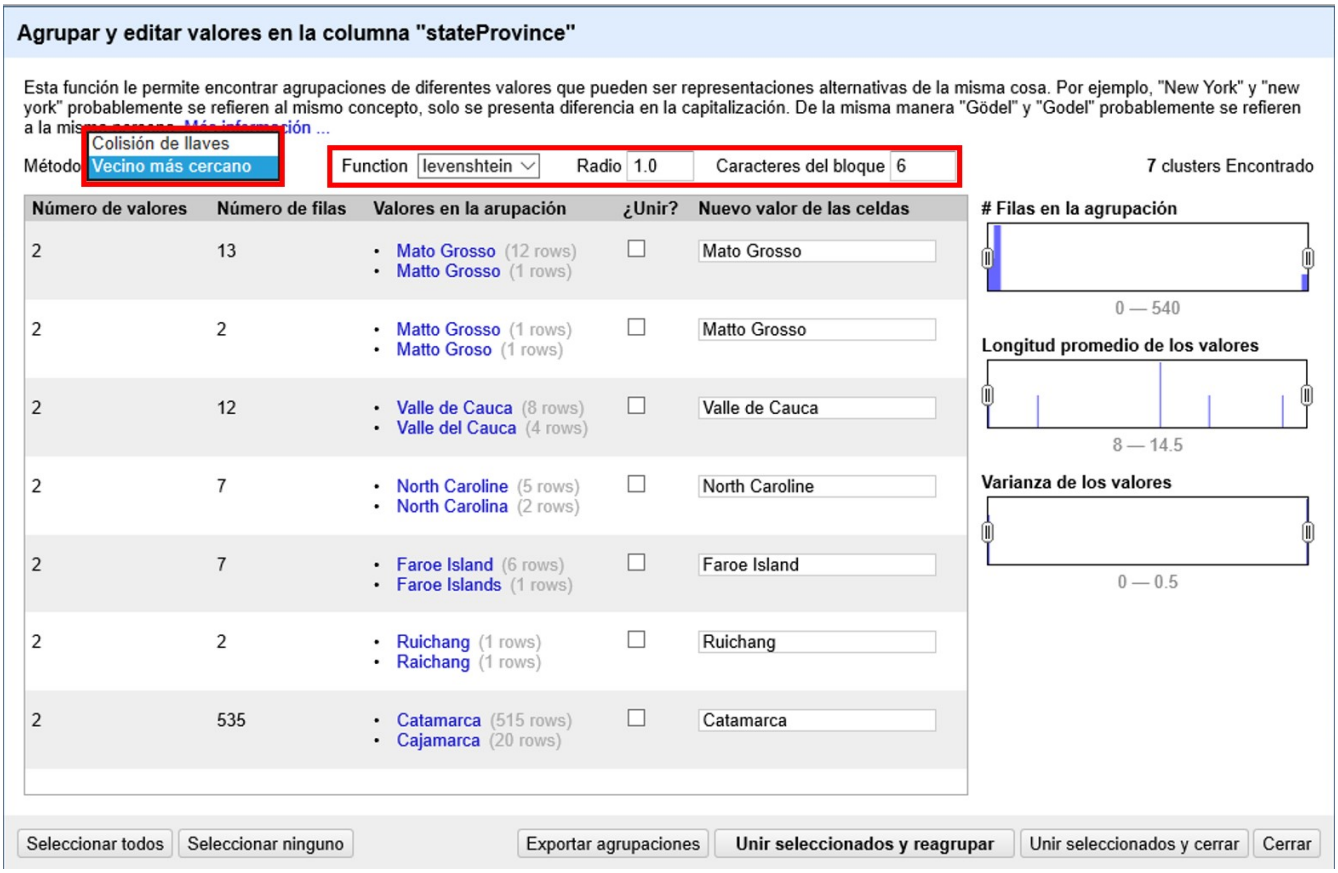

Figura 35

Pruebe agrupamientos con distintos métodos para limpiar los datos.

Para conocer los detalles de cada método de agrupamiento, puede consultar el repositorio de OpenRefine en GitHub.

2.5. Deshacer y rehacer cambios

Ahora que ya ha acumulado una serie de modificaciones al conjunto de datos, veamos cómo se pueden deshacer y rehacer cambios.

En el menú de arriba a la izquierda, abra la pestaña “Deshacer/Rehacer”, que está asociada a un número que indica el número de cambios acumulados hasta ahora. Verá entonces una lista de pasos realizados, como se muestra en la Figura 36a.

Note que el paso resaltado en azul en la figura es el que determina el estado de los datos. Todos los pasos hasta el resaltado, inclusive, han sido aplicados a los datos. Todos aquellos pasos ubicados después del paso resaltado no han sido aplicados.

2.5.1. Deshacer pasos

Si quiere deshacer todo lo posterior a algún paso, simplemente haga click sobre el paso inmediatamente anterior. Por ejemplo, si quiere deshacer los últimos pasos a partir del paso 5, haga click en el paso 5, y los todos los posteriores se revertirán automáticamente (Figura 36b).

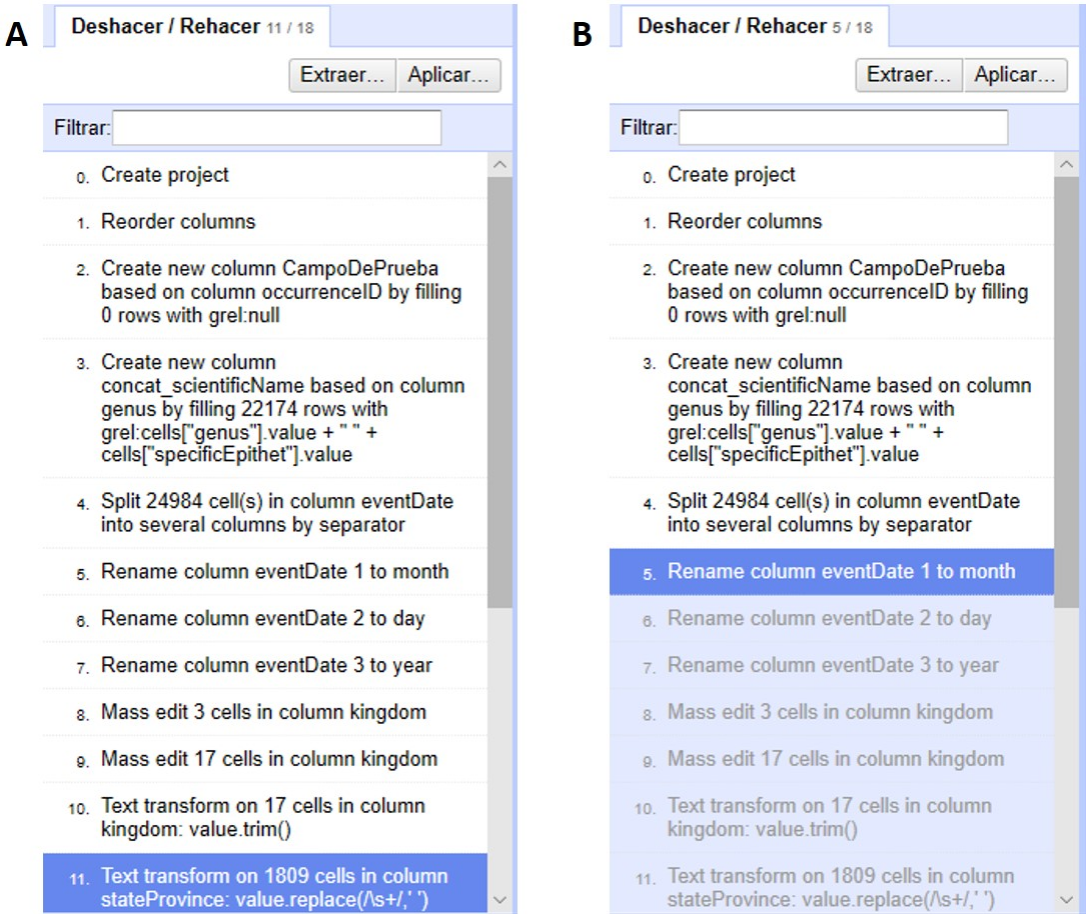

Figura 36

Para rehacer un paso luego de haberlo deshecho, simplemente haga click en ese paso, teniendo en cuenta que entonces se llevarán a cabo todos los pasos intermedios también.

El hacer y deshacer en OpenRefine trabaja sobre “estados”. Eso quiere decir que se puede ir y volver a estados determinados, por ejemplo, el estado de los datos una vez que se han hecho ciertas modificaciones. Ello implica que si se vuelve a un estado anterior y luego se realiza una nueva modificación a partir de ese estado, entonces perderá los pasos originales y no podrá recuperarlos. En el ejemplo de la Figura 36, si se vuelve al paso 5 y luego realiza sobre los datos alguna otra operación, no podrá volver a los pasos 6 a 11 previos.

2.5.2. Guardar pasos para rehacer luego

Es importante entonces que guarde sus pasos, especialmente para aquellos procesos más complejos. Para ello, en la pestaña “Deshacer/Rehacer”, haga click en el botón “Extraer...”. Se abrirá una nueva ventana, como se muestra en la Figura 37, donde puede seleccionar los pasos que desea guardar. Los pasos están dados en formato JSON en el panel de la derecha.

JSON (Java Script Object Notation) es un formato que utiliza texto legible para los humanos para transmitir datos en la forma de pares de atributo:valor y de matrices de datos.

Puede marcar y desmarcar pasos en el panel de la izquierda para seleccionar los pasos de interés. Copie las expresiones de los pasos de interés que se muestran a la derecha a un procesador de texto

(e.g., Notepad, MS Word, etc.) y guárdelas para uso posterior (en caso de que no esté familiarizado con el formato JSON, recuerde tomar nota de qué cambios representan esas expresiones).

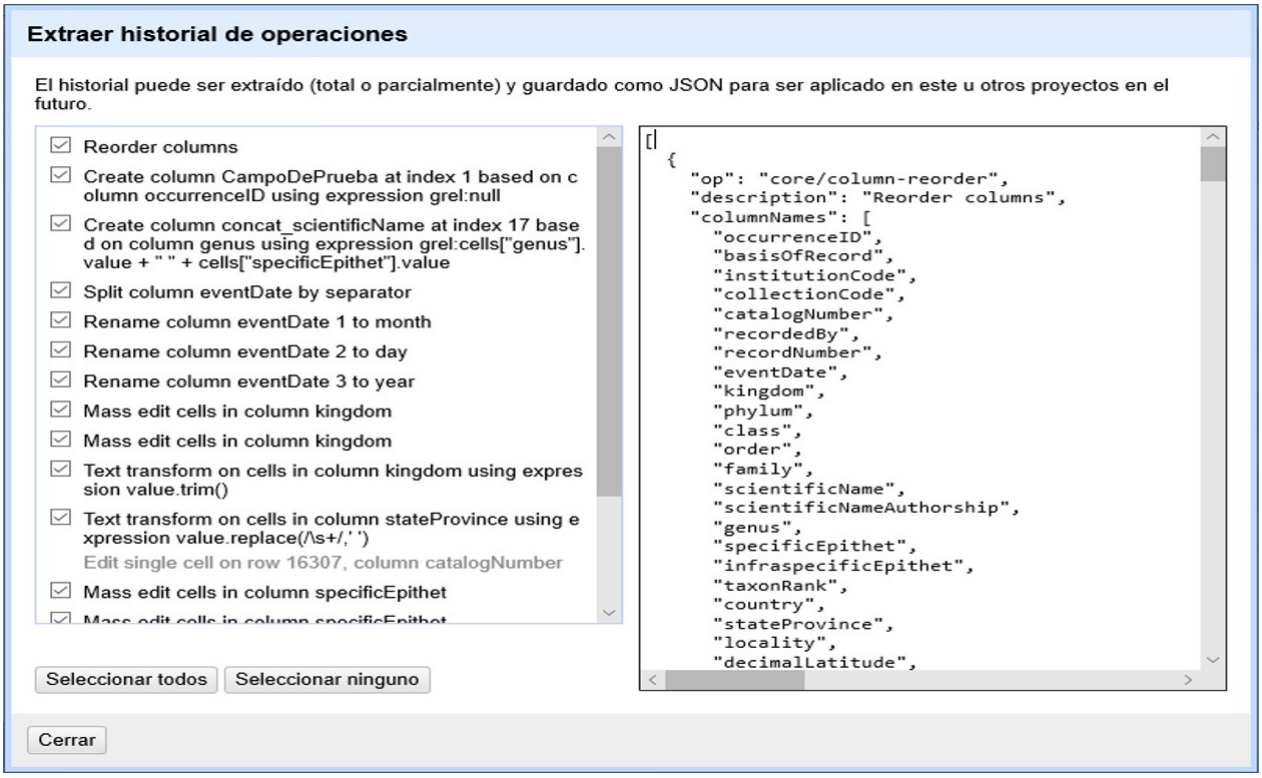

Figura 37

Los cambios hechos a celdas particulares no tienen la opción de guardar expresiones. En el ejemplo anterior, Figura 37, note que el cambio en una celda única del número de catálogo figura en gris y no puede ser seleccionado. Esto es una limitación actual de OpenRefine, por lo que si va a deshacer un cambio de esta naturaleza pero quiere rehacerlo luego, deberá tomar nota usted mismo de cuál fue el cambio y en qué celda de forma separada (e.g, “Cambié el número de catálogo del registro X, de “1234” a “1236””).

2.5.3. Rehacer pasos guardados

Si desea rehacer pasos que tenga guardados (en formato JSON), dentro de la pestaña “Deshacer/Rehacer” haga click en el botón “Aplicar...”. Se abrirá entonces una ventana como la que se muestra en la Figura 38, pero vacía.

Pegue en el cuadro de texto la expresión deseada (copie y pegue lo que guardó en su procesador de texto en el apartado anterior) y haga click en “Ejecutar Operaciones”.

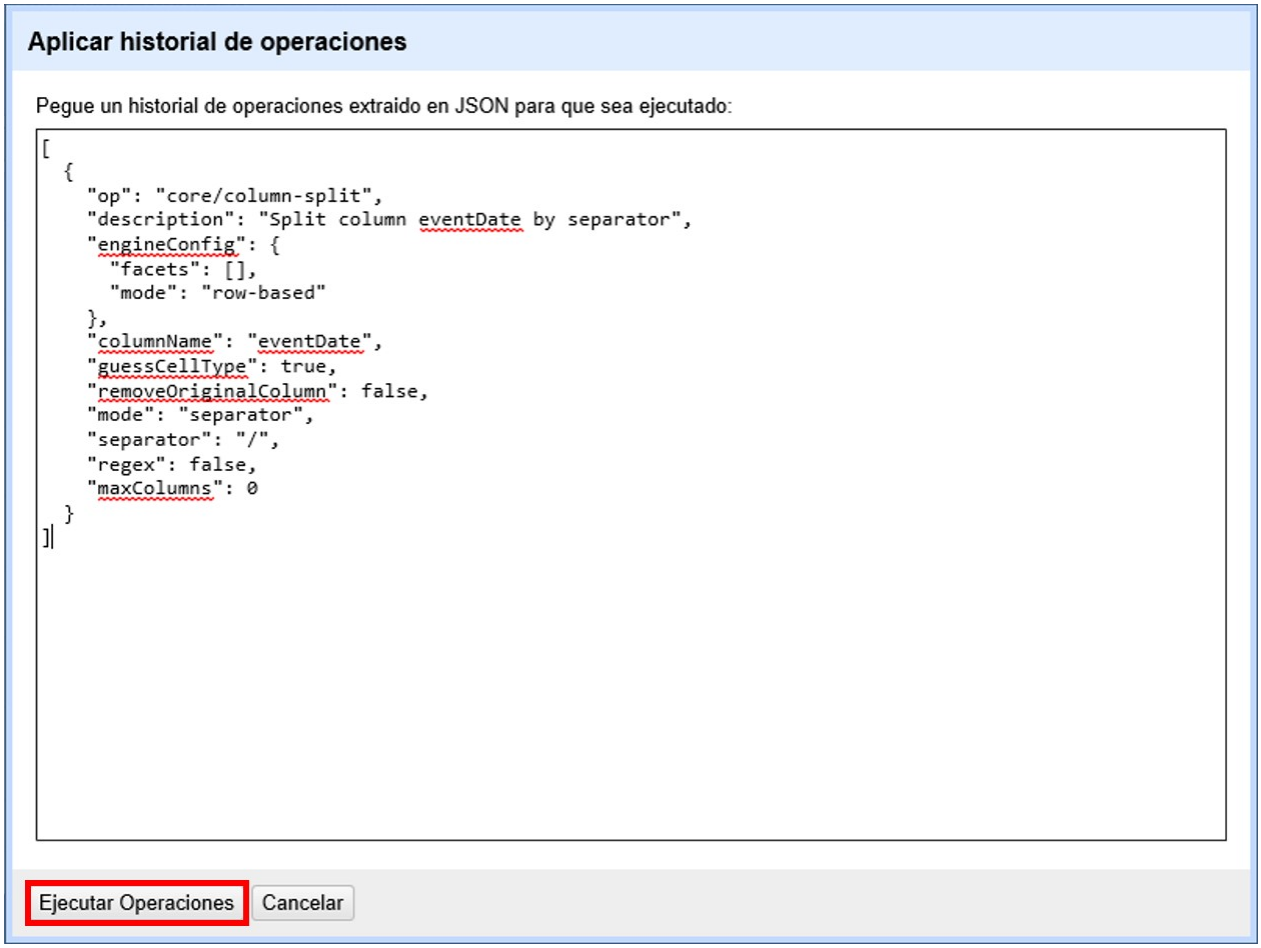

Figura 38

De este modo, puede rehacer pasos particulares o toda una rutina de trabajo, sobre el mismo conjunto de datos, o sobre otros conjuntos de datos (siempre y cuando las columnas sean las mismas).

2.5.4. Reutilizar expresiones regulares

En las secciones anteriores de esta guía (y en las siguientes), muchas funciones involucran utilizar expresiones regulares. Si bien el guardado de pasos es muy útil para repetir procesos, OpenRefine también brinda un simple historial de expresiones regulares que se han utilizado previamente. Se puede acceder a esta lista de expresiones en cualquier ventana que uno abra a partir de una columna, si en dicha ventana se espera el uso de una expresión.

Por ejemplo, al armar una nueva columna a partir del campo `"kingdom"` (click en **la ▼ azul en el campo › Editar columnas › Agregar columna basada en esta columna...** se abre una ventana como la mostrada en la (Figura 39a).

Allí, en la pestaña “Historial” pueden verse listadas las expresiones regulares que se han utilizado sobre este o cualquier otro campo. Para reutilizar las expresiones simplemente hacer click en “Reusar”.

Las expresiones serán incorporadas a los cuadros de texto tal cual fueron utilizadas antes, es decir, podrían contener referencias a otros campos o parámetros no pertinentes. Siempre debe revisar las expresiones para asegurarse de que la función actuará sobre los campos y con los parámetros deseados antes de ejecutar la acción.

Las distintas expresiones pueden ser destacadas haciendo click sobre la estrella que está a su izquierda (que se pintará de color amarillo, Figura 39a). Las expresiones a las que se han asignado estrellas se listarán también bajo el menú “Con estrella” (Figura 39b). Esta función es muy útil cuando la lista de expresiones utilizadas es muy larga y dentro de ella quiere resaltar expresiones de uso más frecuente.

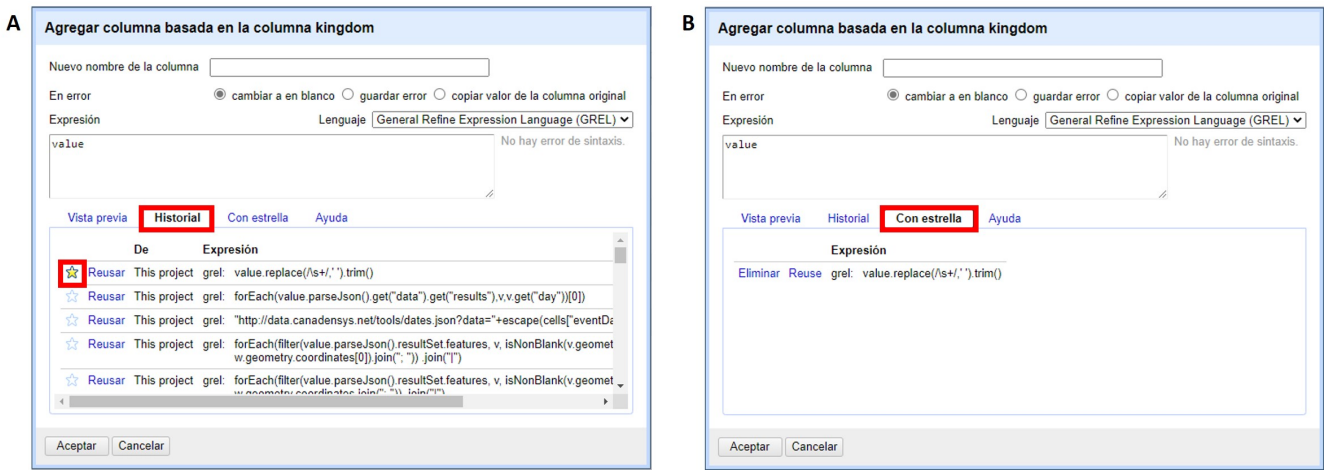

Figura 39

La lista de expresiones utilizadas se mantiene en el programa, de modo que puede volver a utilizarlas en el futuro a partir del historial o del menú “Con estrella”.

2.6. Marcado de registros: banderas y estrellas

OpenRefine ofrece la opción de marcar los distintos registros con banderas (flags) y/o estrellas (stars). Esta opción es a veces muy útil para reconocer registros o grupos de registros rápidamente.

Las banderas y estrellas NO forman parte de los datos. Son solamente una herramienta que facilita el trabajo dentro del programa. Por ello, aunque el marcado se registra como un cambio en el historial de cambios del proyecto, cuando exporte los datos NO verá las columnas que corresponden a estas funciones. Es decir, si usted marcó algún registro con una bandera, por ejemplo, no verá esa bandera ni ninguna otra marca indicadora de su existencia en los datos exportados.

2.6.1. Marcado con banderas y estrellas

Las banderas y estrellas se encuentran dentro del campo “Todo”. Para marcar un registro con una bandera o estrella, simplemente haga click sobre el ícono correspondiente en ese registro (que se pondrá de color amarillo).

Para desmarcar el registro, haga click nuevamente sobre el ícono (que volverá a su color blanco original).

También puede marcar o desmarcar conjuntos de varios registros.

Para ello escoja algún criterio que los agrupe. Por ejemplo, si quiere marcar todos los registros del género Acacia, arme una faceta sobre el campo `"genus"` (haga click sobre **la ▼ azul del campo › Facetas › Faceta de texto**).

En la faceta, seleccione el valor “Acacia” haciendo click en el valor (verá que en la ventana principal sólo se mostrarán esos registros).

Para marcar todos esos registros con una bandera, haga click en **la ▼ azul del campo “Todo”** y siga la siguiente ruta (Figura 40):

Editar filas › Marcar filas con bandera

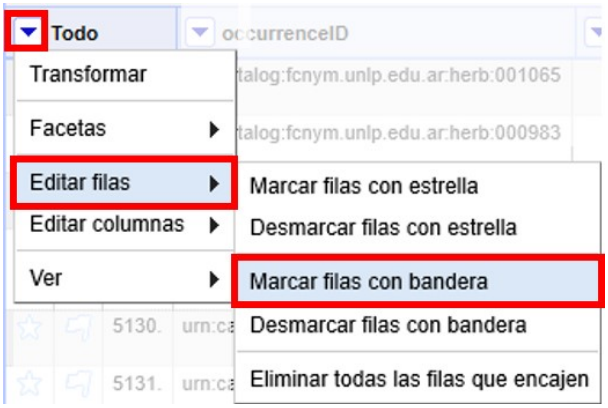

Figura 40

Una vez que lo haya hecho, verá que todos los registros seleccionados están marcados ahora con una bandera.

Para desmarcar todos esos registros, puede hacer click en **la ▼ azul del campo “Todo”** y seguir la ruta:

Editar filas › Desmarcar filas con bandera

Para marcar y desmarcar registros con estrellas, siga el mismo procedimiento con “estrellas” en lugar de “banderas”.

2.6.2. Conservación de banderas y estrellas en la exportación

Si desea marcar los registros de modo que al exportar se conserven las marcas, deberá crear un nuevo campo que capture esa información. Puede, por ejemplo, hacer lo siguiente:

Cree un nuevo campo: sobre cualquier campo haga click en **la ▼ azul › Editar columnas › Agregar columna basada en esta columna...**

Se abrirá una ventana como la mostrada en la Figura 41. Asigne un nombre al campo. Por ejemplo, si sus banderas significan que ha detectado errores en los registros, puede llamarlo “tieneError”.

En el cuadro de texto pegue la siguiente expresión:

```
if(row.flagged, "yes", "no")
```

Esta expresión hará que el campo nuevo tenga como valor “yes” si usted ha asignado una bandera al registro y “no” si no ha asignado una bandera.

Al oprimir “Aceptar” su campo se habrá creado. Verifique los valores que toma asignando a algunos registros una bandera.

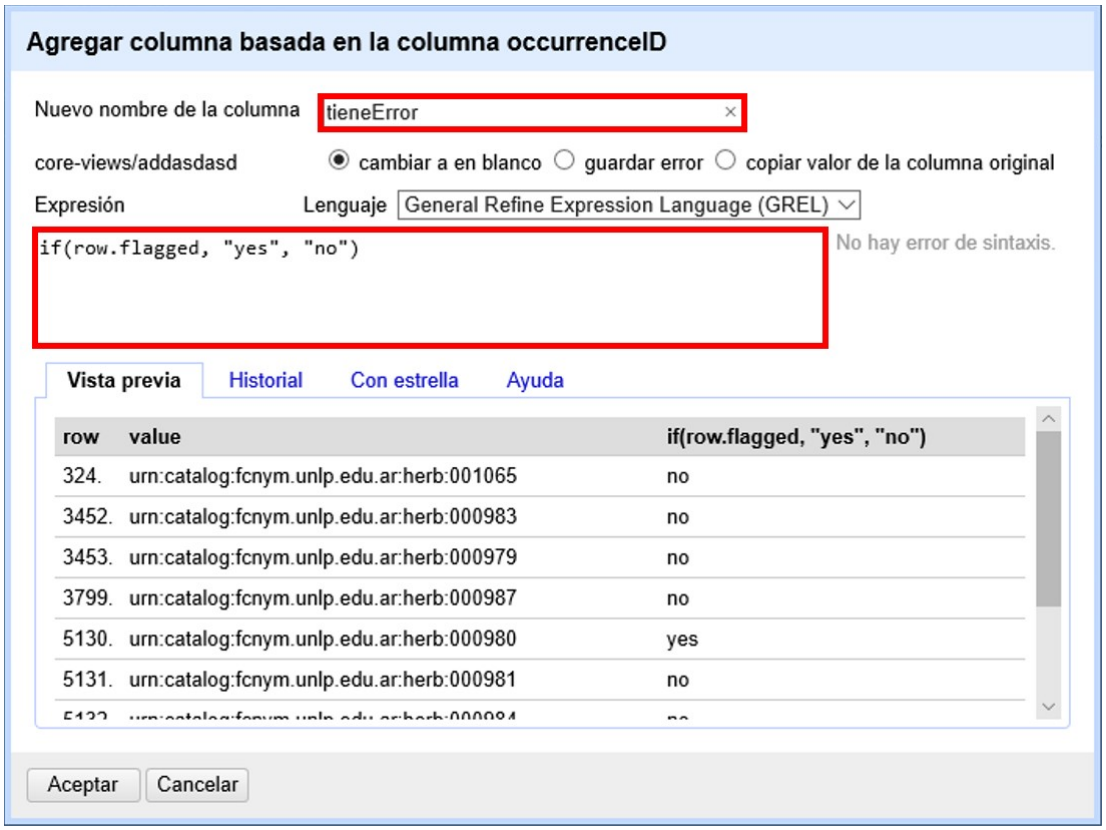

Figura 41

Puede repetir el proceso creando otro campo para las estrellas, usando la expresión:

```
if(row.starred, "yes", "no")
```

Para ver los pasos de exportación de datos, vea la sección de Exportación de datos y proyectos.

2.6.3. Uso de banderas y estrellas para eliminar registros

Las banderas y estrellas se pueden utilizar para eliminar grupos de registros. Para ello, siga los siguientes pasos:

1. Marque con una bandera (o estrella) los registros deseados. Puede hacerlo uno por uno o en grupos a través del marcado dentro de facetas (ver más arriba).
2. Cree una faceta para la bandera. Haga click en **la ▼ azul sobre el campo “Todo” › Facetas › Faceta por bandera** (Figura 42a).

3. En esta nueva faceta, a la izquierda, seleccione la opción “true” haciendo click sobre ella. Ello le mostrará los registros a los que se ha asignado una bandera.

4. Haga click nuevamente sobre **la ▼ azul del campo “Todo” › Editar filas › Eliminar todas las filas que encajen** (Figura 42b).

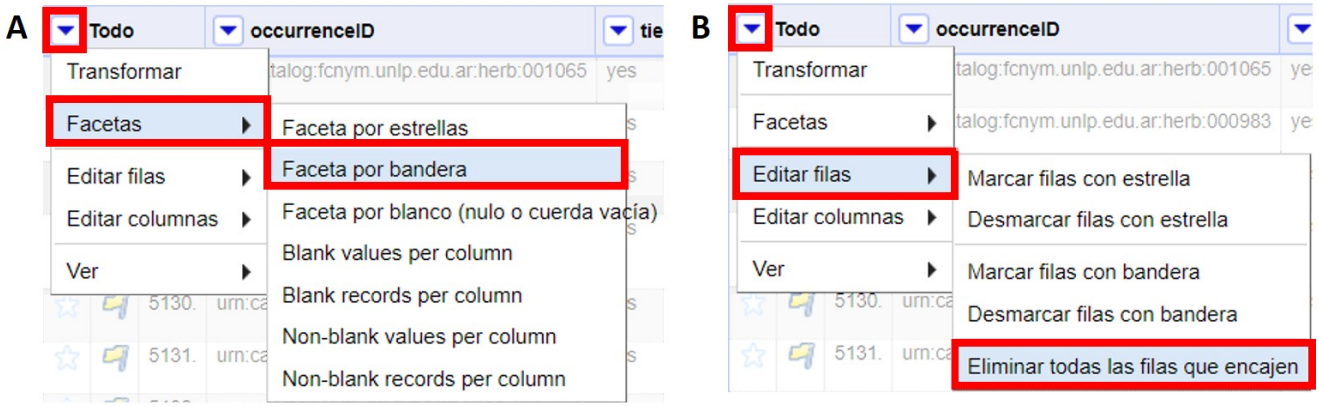

Figura 42

De esta forma habrá eliminado todos los registros que fueron marcados con una bandera.

3. Guardado y exportación de datos y proyectos

Debe tener en cuenta que lo que guarda al usar el programa es el proyecto, y que ello no implica en ningún caso que los cambios que realice vayan a verse reflejados automáticamente en su base de datos original. Para ello, deberá exportar los datos desde OpenRefine e importarlos nuevamente a su base de datos.

3.1. Guardado de datos y proyectos

Los proyectos con los que trabaja usando OpenRefine son guardados en su propia computadora de forma automática. En otras palabras, no existe un botón o un comando “Guardar”.

Los directorios en que se guardan los proyectos se listan a continuación:

Windows: dependiendo de la versión de Windows que utilice, los datos se encontrarán en uno de estos directorios:

- C:\Documents and Settings\(user id)\Local Settings\Application Data\OpenRefine
- C:\Users\(user id)\AppData\Roaming\OpenRefine
- C:\Users\(user id)\AppData\Local\OpenRefine
- C:\Users\(user id)\OpenRefine

MacOS:

- ~/Library/Application Support/OpenRefine/
- ~/Library/Application Support/Google/Refine/ (versions de Google Refine más antiguas)
- Ingreso a través de /var/log/daemon.log - grep para com.google.refine.Refine

Linux:

- ~/.local/share/openrefine/

3.2. Exportación de datos y proyectos

OpenRefine ofrece varias opciones para exportar los datos y proyectos. Se puede acceder a estas opciones en la esquina superior derecha de la ventana del programa, haciendo click en el botón “Exportar” (Figura 43).

Note que la primera opción, “Exportar proyecto”, permite exportar el proyecto completo, mientras que otras opciones (e.g., delimitado por... , Excel, etc.) permiten exportar los datos.

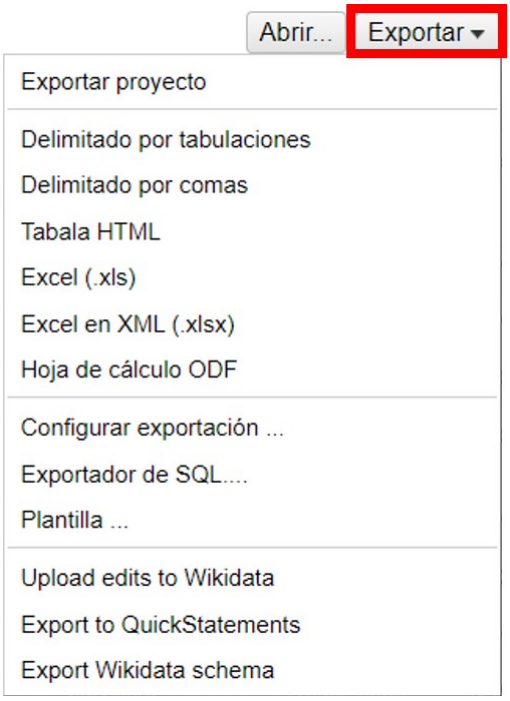

Figura 43

La exportación de proyectos es útil cuando uno quiere abrir el mismo proyecto en OpenRefine en otra computadora.

Haciendo click en “Exportar proyecto” se abrirá una ventana en la que puede escoger si exportar como archivo local o si exportar a Google Drive (Figura 44).

Figura 44

Escoja la opción deseada y haga click en “Exportar”. El archivo exportado tendrá una extensión .tar.gz, que sólo puede ser abierto por el programa (no se descarga un archivo de datos que pueda abrir en un procesador de textos ni en una planilla de cálculo).

Para exportar los datos y poder abrirlos en otro programa, puede seguir cualquiera de las otras opciones, que resultarán en un archivo con uno de los formatos disponibles.

La exportación de datos se realizará teniendo en cuenta las facetas y filtros aplicados. Esto implica que si usted tiene abierta por ejemplo una faceta, sólo los datos correspondientes a dicha faceta serán exportados. Por lo tanto, para asegurarse de exportar todos los datos, recuerde cerrar todos los filtros y facetas antes de hacer la exportación.

Para una exportación más personalizada, en el menú “Exportar” escoja “Configurar exportación...”. Se abrirá una ventana como la mostrada en la Figura 45, en la cual puede escoger una serie de

opciones.

Figura 45

En la pestaña “Contenido” puede elegir qué campos exportar y modificar ciertos parámetros para cada campo individualmente.

Observe que en esta pestaña también puede escoger ignorar todas las facetas y filtros al exportar, lo cual es muy útil en caso de que haya olvidado cerrar alguna.

Para descargar los datos, vaya a la pestaña “Descarga”, como se ve en la Figura 46.

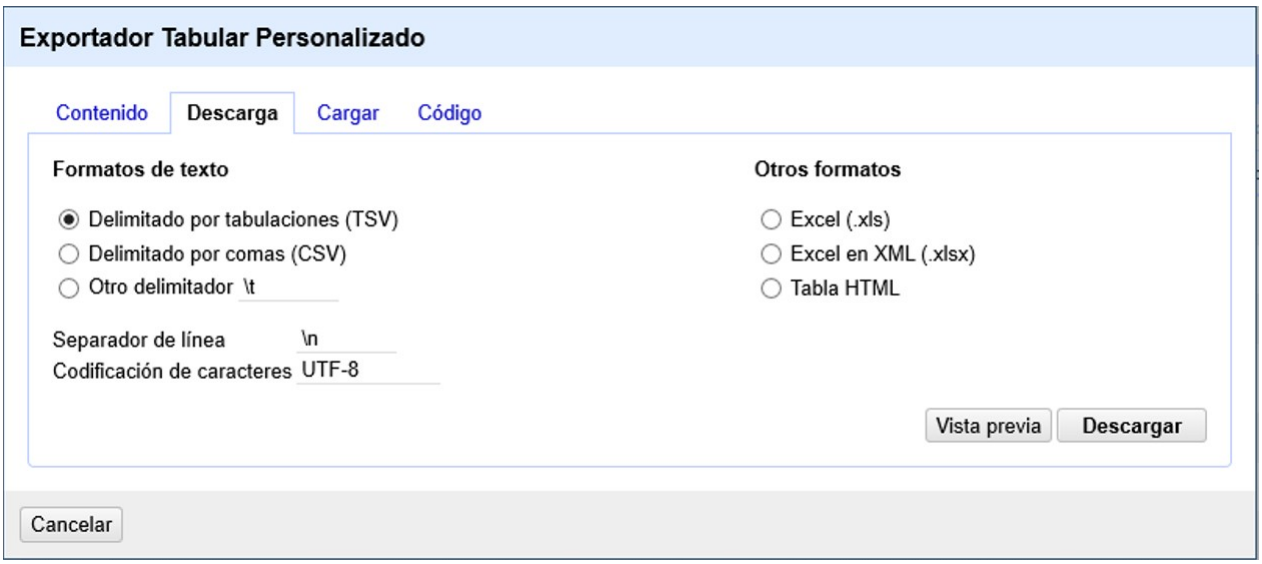

Figura 46

En esta pestaña puede seleccionar el formato de los datos para la descarga. Escoja el que prefiera y haga click en “Descargar”. Inmediatamente comenzará la descarga de los datos.

También, para ver una vista previa de los datos que descargará, puede hacer click en “Vista previa”, y

se abrirá otra ventana en su navegador web donde podrá ver una muestra de los datos a descargar.

4. Consultas a servicios externos a través de URLs

OpenRefine ofrece la posibilidad de consultar fuentes externas, una función que es muy útil cuando se intenta mejorar la calidad de los datos. Para el caso particular de datos sobre biodiversidad, permite, por ejemplo, validar nombres taxonómicos y geográficos contra fuentes de información que se consideren confiables, completar rangos taxonómicos y campos de geografía administrativa, georreferenciar, incorporar enlaces a imágenes almacenadas en sitios web, entre otros.

En OpenRefine las consultas externas pueden realizarse por dos vías: a través de URLs, o a través de servicios de reconciliación. En esta guía sólo se incluyen los métodos referidos a las consultas a través de URLs. Para ver explicaciones referidas al uso de algunos servicios de reconciliación consultar versiones anteriores de este documento; tener en cuenta que esos servicios no han sido actualizados en concordancia con las actualizaciones de OpenRefine, y muchos no funcionan a partir de la versión 2.8 de OpenRefine.

Debe recordarse que para poder realizar consultas a servicios que se encuentran en línea se requiere conexión a Internet.

La velocidad a la que se obtienen los resultados de las consultas depende de la velocidad de respuesta del servicio en particular. De esta forma, si se quiere obtener información para muchos registros, el tiempo de la operación será prolongado. Para acortar tiempos, se pueden hacer comparaciones de registros contra el servicio deseado dentro de facetas, es decir, en fracciones particulares de los registros.

Nos referimos a consultas a través de URLs cuando el proceso implica proveer a OpenRefine con la dirección web (URL) de un determinado servicio y ciertos parámetros mínimos para obtener de dicho servicio un resultado.

4.1. Resolución de nombres científicos usando Global Names Resolver

En el ejemplo siguiente, compararemos los nombres científicos (contenidos en el campo `"scientificName"`) contra el servicio Global Names Resolver (de aquí en más "GNR").

Para acortar el tiempo de consulta, cree una faceta para el campo `"genus"` (click en **la ▼ azul › Facetas › Faceta de texto**) y dentro de ella escoja el género Cinna. En el conjunto de datos utilizado Cinna tiene 3 especies asociadas: *C. lateralis* (1 registro), *C. arundinacea* (6 registros) y *C. latifolia* (3 registros); puede verlas listadas en el campo `"scientificName"`.

Para comparar los nombres contra el GNR, haremos un llamado al servicio y capturaremos los resultados en un nuevo campo:

A partir del campo `"scientificName"`, cree una nueva columna a partir de una dirección URL haciendo click en **la ▼ azul del campo** y siguiendo la siguiente ruta (Figura 47):

Editar columnas › Agregar columna accediendo a URLs...

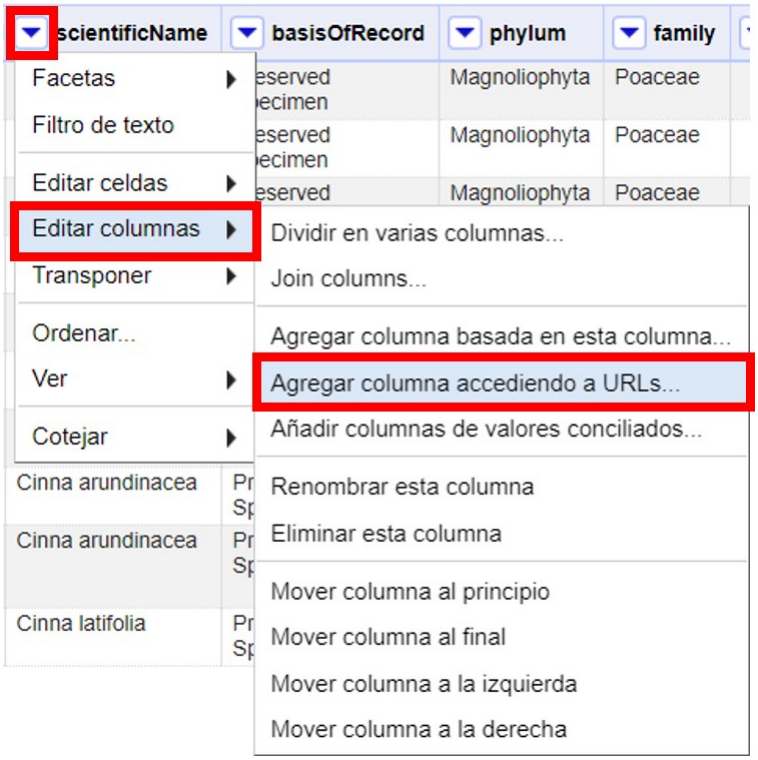

Figura 47

Se abrirá una ventana como la mostrada en la Figura 48. Allí, dé un nombre al nuevo campo (por ejemplo, GNR_Json_sciName), y en el cuadro de texto coloque la siguiente expresión:

```
"http://resolver.globalnames.org/name_resolvers.json?names=" +
  escape(cells["scientificName"].value,"url")
```

Dicha expresión indica que se hará una consulta en el GNR utilizando como valores de comparación aquellos que se encuentran en el campo `"scientificName"`. Es importante que el nombre del campo que utiliza en la expresión sea idéntico al nombre del campo del cual tomará los valores originales, o de otro modo el llamado será infructuoso.

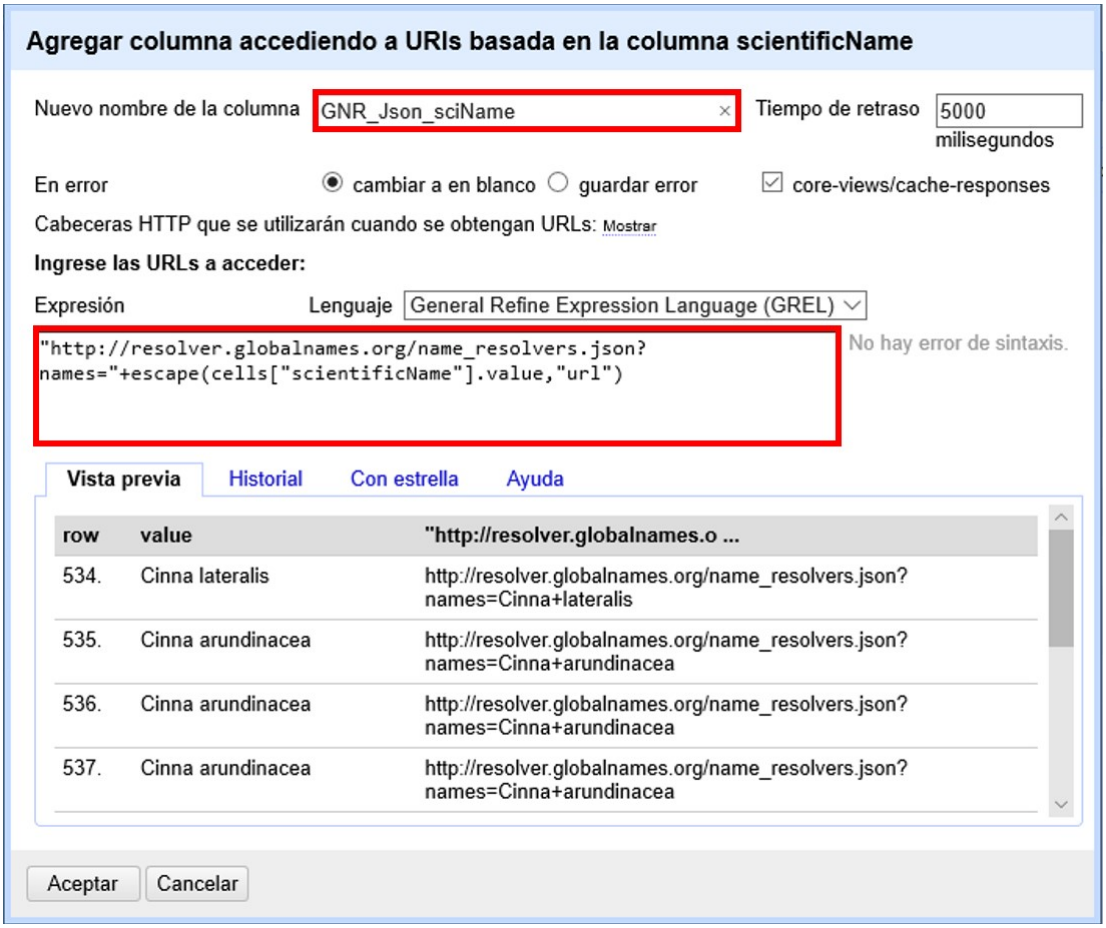

Figura 48

Note que en esta ventana, arriba a la derecha, tiene una opción para modificar el "Tiempo de retraso". Este valor indica el tiempo que transcurre entre llamados o consultas sucesivas que se hacen al servicio en cuestión. Por defecto, el valor es de 5000 milisegundos. Puede reducir este tiempo para acelerar el proceso de comparación. Tenga en cuenta, sin embargo, que muchos servicios bloquean los llamados si éstos ocurren muy cercanos en el tiempo, pues consideran que puede tratarse de un ataque. El máximo número de consultas que se pueden realizar por unidad de tiempo depende de cada servicio en particular.

La expresión utilizada es muy general, y devolverá los valores de todos los parámetros que GNR provee respecto de un nombre científico. Puede consultar dichos parámetros en http://resolver.globalnames.org/api. Si no quiere obtener en el resultado todos los valores, puede modificar la expresión especificando valores para todos o algunos de los parámetros. Por ejemplo: GNR resuelve los nombres consultando diferentes fuentes, a las que asigna identificadores (data_source_id); si sólo quiere obtener los resultados provenientes de la fuente Catalogue of Life (que en GNR tiene id=1), puede utilizar la siguiente expresión:

```
"http://resolver.globalnames.org/name_resolvers.json?names=" +
  escape(cells["scientificName"].value,"url") +
  "&data_source_ids=1"
```

Una vez que haya creado el nuevo campo con la expresión general, verá que contiene, en formato JSON, los resultados de la consulta en GNR para cada nombre, con todos los parámetros y valores

que GNR reporta.

Para poder trabajar con esto más cómodamente, debemos extraer de allí los valores de interés.

Dado que GNR consulta varias fuentes de nombres taxonómicos, nos interesa saber cuál es el nombre científico que figura en cada fuente. Algunas fuentes pueden tener listado el nombre pero considerarlo inválido y proveer el nombre correcto. Entonces, extraeremos del resultado en JSON, en un nuevo campo, los siguientes valores:

- Fuente consultada: "data_source_title"
- Nombre encontrado en la fuente: "name_string"
- Nombre aceptado por la fuente: "current_name_string"

Para ello, a partir del campo en JSON (en el ejemplo, GNR_Json_sciName), cree un nuevo campo (haga click en **la ▼ azul › Editar columnas › Agregar columna basada en esta columna**).

Dé un nombre al nuevo campo (por ejemplo, GNR_sciName_options) y en el cuadro de texto, coloque la siguiente expresión (Figura 49):

```
forEach(value.parseJson().get("data")[0].get("results"), v,
  v.get("data_source_title") + "; " +
  v.get("name_string") + "; " +
  if(isBlank(v.get("current_name_string")), "", v.get("current_name_string")))
.join(" | ")
```

Dicha expresión analiza la cadena en formato JSON, que tiene dentro de su estructura secciones “data” y dentro de esta “results” –un “result” proveniente de cada fuente consultada (por ejemplo, un “result” de Catalogue of Life). Dentro de cada sección “results” extrae los valores de interés (“data_source_title”, “name_string” y “current_name_string”) y los separa con un “;”. Como no todas las fuentes proveen un nombre aceptado (“current_name_string”), la expresión `if` especifica que si ese parámetro es nulo debe dejarse el espacio vacío (`""`), y si no, colocar el valor extraído. Por último, une los grupos de valores extraídos en una única cadena de texto, separados por un `|` .

Figura 49

Una vez que haya creado el campo, verá que contiene los valores de interés extraídos de GNR separados por ' | '. Por ejemplo:

```
uBio NameBank; Cinna lateralis; | Catalogue of Life; Cinna lateralis Walter;
Andropogon virginicus L. | ITIS; Cinna lateralis Walter; Andropogon virginicus L. |
GBIF Backbone Taxonomy; Cinna lateralis Walter; Andropogon virginicus L. | EOL; Cinna
lateralis Walter; | Tropicos - Missouri Botanical Garden; Cinna lateralis Walter; |
The International Plant Names Index; Cinna lateralis Walter; | uBio NameBank; Cinna
lateralis Walter; | uBio NameBank; Cinna lateralis Walter, 1788; | Arctos; Cinna
lateralis Walter;
```

Note que algunas fuentes encuentran el nombre pero no proveen un nombre aceptado, por ejemplo:

```
uBio NameBank; Cinna lateralis;
```

no tiene un valor en el tercer lugar, mientras que:

```
Catalogue of Life; Cinna lateralis Walter; Andropogon virginicus L.
```

provee el nombre encontrado y el nombre válido.

Note además que algunas fuentes tienen más de una variante asociada al nombre, por ejemplo:

```
uBio NameBank; Cinna lateralis;
uBio NameBank; Cinna lateralis Walter;
uBio NameBank; Cinna lateralis Walter, 1788;
```

No todos los nombres serán necesariamente encontrados en todas las fuentes consultadas, por lo que el número de fuentes variará de un nombre al otro. En consecuencia, la ubicación de las fuentes en la cadena de texto no será homogénea de un registro al otro. Una consecuencia de esto es que si usted quiere luego separar el contenido en campos distintos de acuerdo a la fuente consultada (e.g., un campo para ITIS, uno para Catalogue of Life, etc.), no podrá hacerlo de modo que cada nuevo campo tenga los datos de una misma y única fuente.

En este caso, le conviene en cambio hacer varios llamados a GNR separados, cada uno especificando una fuente determinada. Como se menciona más arriba, si quiere por ejemplo sólo consultar los valores dados por Catalogue of Life, use la expresión siguiente:

```
"http://resolver.globalnames.org/name_resolvers.json?names=" +
  escape(cells["scientificName"].value,"url") +
  "&data_source_ids=1"
```

y luego arme un nuevo campo extrayendo los resultados de interés, usando la expresión:

```
forEach(value.parseJson().get("data")[0].get("results"), v,
  v.get("data_source_title") + "; " +
  v.get("name_string") + "; " +
  if(isBlank(v.get("current_name_string")), "", v.get("current_name_string")))
.join(" | ")
```

A partir de los resultados obtenidos, puede extraer los nombres separando la nueva columna en columnas distintas utilizando separadores apropiados (ver Divisiones en la sección 2.1.3).

4.2. Georreferenciación usando GEOLocate

En este ejemplo, para facilitar la explicación y reducir el tiempo de consulta al servicio, construiremos previamente dos facetas. La primera sobre el campo `"country"`, dentro de la cual seleccionaremos el valor “Argentina”. La segunda faceta será sobre el campo `"genus"`, dentro de la cual seleccionaremos el valor “Acacia”. Una vez aplicadas ambas facetas y escogidos los valores, verá que en la ventana principal sólo se muestra un subconjunto de registros que cumplen estas condiciones simultáneamente.

Llevaremos a cabo la georreferenciación a partir del campo `"locality"`. Para ello, cree un nuevo campo a partir de éste siguiendo la ruta: click en **la ▼ azul › Editar columnas › Agregar columna accediendo a URLs...**.

Se abrirá una nueva ventana (Figura 50). Allí dé un nombre al nuevo campo, por ejemplo

“GeoLocate_Json_georref”, y pegue en el cuadro de texto la siguiente expresión:

```
"http://www.geo-
locate.org/webservices/geolocatesvcv2/glcwrap.aspx?Country=Argentina&fmt=json&Locality
=" +
  escape(value,'url')
```

En esta expresión, `fmt` indica el formato en el que el resultado será devuelto por el servicio. GEOLocate ofrece dos posibles formatos, JSON y GeoJSON.

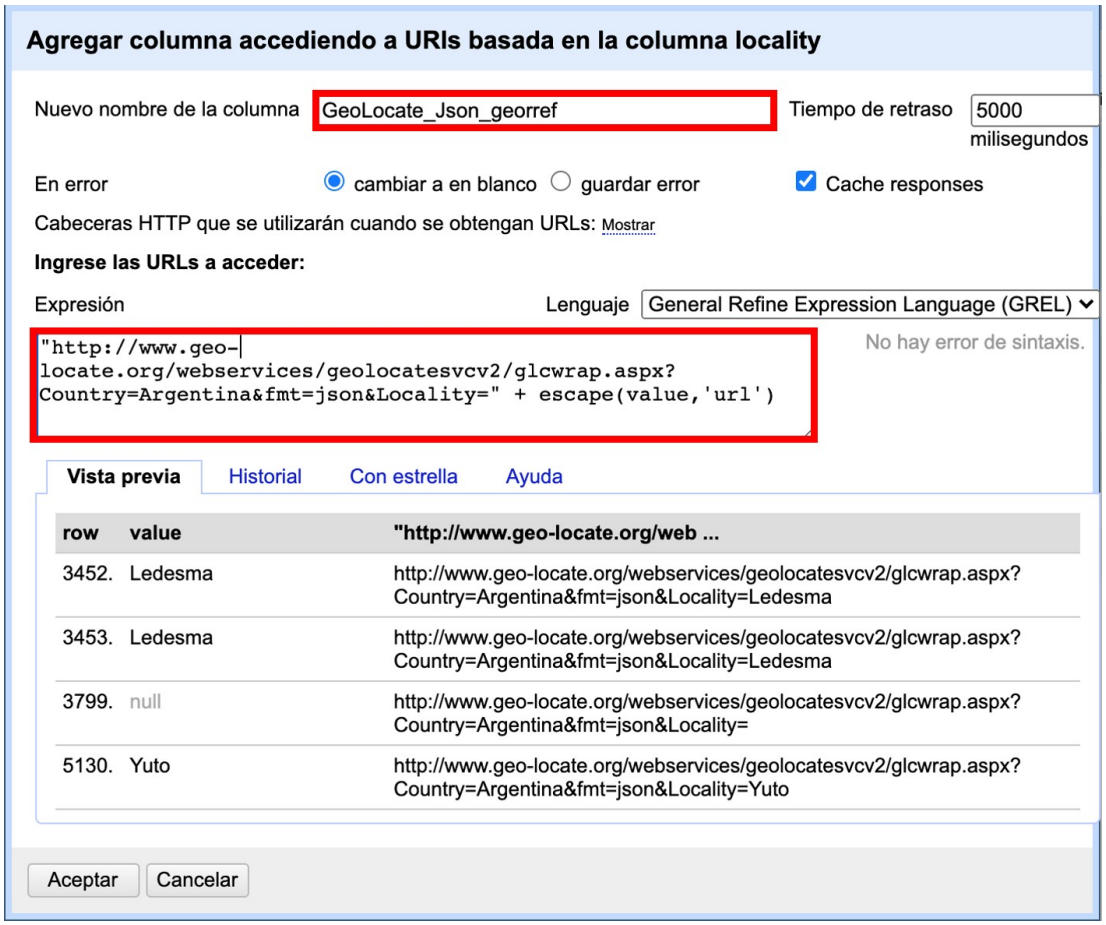

Figura 50

Una vez que haya creado el nuevo campo con la expresión general, verá que contiene, en formato JSON, los resultados de la consulta en GEOLocate para cada localidad, con todos los parámetros y valores que este servicio reporta.

En los resultados puede tener tres casos:

Caso 1) Ningún resultado encontrado. Ello quiere decir que GEOLocate no ha podido ubicar la localidad de interés. En la celda correspondiente verá lo siguiente:

```
{
  "engineVersion" : "GLC:7.0|U:1.01374|eng:1.0",
  "numResults" : 0, ①
  "executionTimems" : 578.1462
}
```

① Ningún resultado encontrado.

Caso 2) Un único resultado encontrado. En la celda correspondiente verá, por ejemplo, lo siguiente:

```
{
  "engineVersion": "GLC:7.0|U:1.01374|eng:1.0",
  "numResults": 1,
  "executionTimems": 484.3969,
  "resultSet": {
    "type": "FeatureCollection",
    "features": [
      {
        "type": "Feature",
        "geometry": {
          "type": "Point",
          "coordinates": [ -64.471941, -23.643418 ] ①
        },
        "properties": {
          "parsePattern": "YUTO", ②
          "precision": "High",
          "score": 79,
          "uncertaintyRadiusMeters": 3036, ③
          "uncertaintyPolygon": "Unavailable", ④
          "displacedDistanceMiles": 0, ⑤
          "displacedHeadingDegrees": 0,
          "debug":
":GazPartMatch=False|:inAdm=True|:Adm=JUJUY|:NPExtent=5040|:NP=YUTO|:KFID=|YUTO" ⑥
        }
      }
    ],
    "crs": { "type": "EPSG", "properties": { "code": 4326 } }
  }
}
```

① Las coordenadas: `"coordinates": [-64.471941, -23.643418]`

② Las localidad original que consultó: `"parsePattern" : "YUTO"`

③ El radio de incerteza en metros: `"uncertaintyRadiusMeters" : 3036`

④ El polígono de incerteza asociado: `"uncertaintyPolygon" : "Unavailable"`, en este caso no disponible.

⑤ Los desplazamientos: distancia en millas y grados en una dirección: `"displacedDistanceMiles" : 0, "displacedHeadingDegrees" : 0`, en este caso con valores `0` porque no se especifica

desplazamiento de ningún tipo en la localidad (e.g., 45km de Yuto, o 45km N Yuto).

⑥ La correspondencia en el gacetero consultado: `GazPartMatch`, y en éste la división administrativa bajo la cual se encontró la localidad: `|:Adm=JUJUY|`.

Caso 3) Varios resultados encontrados para un mismo valor de localidad. Esto sucede comúnmente cuando no se especifican en la consulta niveles administrativos por debajo de país (e.g., podría haber en un mismo país varios lugares con el mismo nombre). Un ejemplo sería:


```
{
  "engineVersion": "GLC:7.0|U:1.01374|eng:1.0",
  "numResults": 3, ①
  "executionTimems": 468.7555,
  "resultSet": {
    "type": "FeatureCollection",
    "features": [
      {
        "type": "Feature",
        "geometry": {
          "type": "Point",
          "coordinates": [ -64.158097, -26.21252 ] ②
        },
        "properties": {
          "parsePattern": "TARTAGAL", ③
          "precision": "High",
          "score": 83,
          "uncertaintyRadiusMeters": 301,
          "uncertaintyPolygon": "Unavailable",
          "displacedDistanceMiles": 0,
          "displacedHeadingDegrees": 0,
          "debug": ":GazPartMatch=False|:inAdm=True|:Adm=SANTIAGO DEL
ESTERO|:NPExtent=500|:NP=TARTAGAL|:KFID=|TARTAGAL" ④
        }
      },
      {
        "type": "Feature",
        "geometry": {
          "type": "Point",
          "coordinates": [ -59.846115, -28.671732 ] ②
        },
        "properties": {
          "parsePattern": "TARTAGAL", ③
          "precision": "High",
          "score": 83,
          "uncertaintyRadiusMeters": 3036,
          "uncertaintyPolygon": "Unavailable",
          "displacedDistanceMiles": 0,
          "displacedHeadingDegrees": 0,
          "debug": ":GazPartMatch=False|:inAdm=True|:Adm=SANTA
FE|:NPExtent=5040|:NP=TARTAGAL|:KFID=|TARTAGAL" ④
        }
```

```
      },
      {
        "type": "Feature",
        "geometry": {
          "type": "Point",
          "coordinates": [ -63.801314, -22.516365 ] ②
        },
        "properties": {
          "parsePattern": "TARTAGAL", ③
          "precision": "High",
          "score": 83,
          "uncertaintyRadiusMeters": 3036,
          "uncertaintyPolygon": "Unavailable",
          "displacedDistanceMiles": 0,
          "displacedHeadingDegrees": 0,
          "debug":
":GazPartMatch=False|:inAdm=True|:Adm=SALTA|:NPExtent=5040|:NP=TARTAGAL|:KFID=|TARTAGA
L" ④
        }
      }
    ],
    "crs": { "type": "EPSG", "properties": { "code": 4326 } }
  }
}
```

Note que los tres resultados del ejemplo corresponden a provincias distintas en las que se encuentra una localidad “Tartagal”, puede comparar las coordenadas para cada una.

Visualizando JSON

Para visualizar la estructura de los resultados en JSON de modo más amigable, puede probar copiando el resultado de alguna celda en un analizador de JSON en línea. Existen muchas opciones, una de ellas es http://json.parser.online.fr/. Allí, seleccionando distintas opciones arriba a la derecha podrá distinguir mejor la estructura, cuáles son los objetos, los arreglos y las cadenas de texto y cómo están relacionados unos con otros (Figura 51). Esto puede ser muy útil a la hora de armar expresiones para desglosar el contenido de los campos en nuevos campos sin perder información.

Figura 51

La expresión utilizada es muy simple y sólo le pide al servicio que resuelva la georreferenciación en base al campo localidad y teniendo como valor fijo “Argentina” para el campo país, pero sin especificar los valores de otros campos geográficos. Sin embargo, todos los campos se pueden incluir en la expresión para obtener resultados más específicos. Ello puede hacerse de dos maneras:

1. Establecer los valores de los campos como valores fijos, como hicimos con el país, agregando luego por ejemplo: `&state=VALOR` donde VALOR es el valor fijo que uno establece (e.g., “Córdoba”). Esto restringirá los resultados en función de esos parámetros.
2. Incluir los campos como valores a consultar, en cuyo caso para cada campo hay que incluir como valor: `escape(cells.NOMBREDELCAMPO.value,'url')`

La expresión con todos los campos se verá entonces como:

```
"http://www.geo-
locate.org/webservices/geolocatesvcv2/glcwrap.aspx?country=Argentina&state=" +
  escape(cells.stateProvince.value,'url')+"&locality="+escape(cells.locality.value,
'url')
```

Note que el nombre del campo será el que tiene en su base de datos. Note también que en la base de datos dada para este ejercicio no hay un campo correspondiente a `"county"`, pero GEOLocate permite incluirlo si lo hubiera.

Para poder trabajar con estos resultados más cómodamente, debemos extraer de allí los valores de interés. En este paso debe tener cuidado. Debido a que no especificamos todos los campos geográficos en la consulta a GEOLocate, recuerde que los registros pueden tener más de un resultado posible, y que cada resultado tiene sus propios parámetros de georreferenciación.

A modo de ejemplo, extraeremos en nuevos campos los valores de las coordenadas. (El conjunto de datos provisto para realizar los ejercicios de esta guía contiene campos originales de latitud y

longitud provistos por la fuente, puede utilizarlos para contrastar los resultados obtenidos utilizando GEOLocate).

Para extraer las coordenadas puede seguir dos métodos: 1) extraer latitud y longitud conjuntamente y luego separar; o 2) extraer latitud y longitud de modo independiente.

Método 1: extraer latitud y longitud conjuntamente

Haga click en **la ▼ azul del campo** `"GeoLocate_Json_georref"` **› Editar columnas › Agregar columna basada en esta columna**.

De un nombre al nuevo campo, por ejemplo, GeoLocate_parseCoord, y en el cuadro de texto pegue la siguiente expresión:

```
forEach(filter(value.parseJson().resultSet.features, v, isNonBlank(v.geometry)), w,
  w.geometry.coordinates.join("; "))
.join("|")
```

Esta expresión es un poco más compleja que las que hemos estado utilizando, debido a que se requiere extraer información de una estructura JSON particular Objeto → Arreglo → Objeto → Arreglo. (Puede visualizar la estructura en JSON como se menciona en la nota de la Figura 51).

El nuevo campo tendrá valores como los siguientes, por ejemplo, para un registro cuya consulta devolvió tres resultados:

```
-64.158097; -26.21252|-59.846115; -28.671732|-63.801314; -22.516365
```

Note que GEOLocate provee como primer valor de coordenadas la longitud y como segundo valor la latitud.

Dividiremos ahora este campo en tres partes, una para cada resultado:

Haga click en **la ▼ azul del campo › Editar columnas › Dividir en varias columnas**.

Escoja como separador `|`. Desmarque la opción “Eliminar esta columna” si quiere mantener el campo original (esto es recomendable, siempre puede eliminar los campos después).

Tendrá entonces ahora una serie de campos con valores del tipo: `-64.158097; -26.21252`. Sobre cada uno, puede realizar una nueva separación utilizando como separador `;`.

Método 2: extraer latitud y longitud independientemente

Haga click en **la ▼ azul del campo** `"GeoLocate_Json_georref"` **› Editar columnas › Agregar columna basada en esta columna**.

De un nombre al nuevo campo, por ejemplo, GeoLocate_parseLong, y en el cuadro de texto pegue la siguiente expresión:

```
forEach(filter(value.parseJson().resultSet.features, v, isNonBlank(v.geometry)), w,
  w.geometry.coordinates[0]).join("; "))
```

Esta expresión es diferente a la usada anteriormente en que se especifica qué valor del arreglo coordenadas se desea obtener: `[0]`. En OpenRefine, el primer valor se indica con 0, el segundo con 1, y así sucesivamente. Dado que en los resultados de la consulta se indica primero la longitud, ésta será el valor `[0]`, y la latitud será el valor `[1]` dentro del arreglo "coordinates".

El nuevo campo creado tendrá valores como los siguientes: `-64.158097; -59.846115; -63.801314` cada uno correspondiente a una longitud de uno de los resultados obtenidos de la consulta a GEOLocate para un determinado registro.

Puede repetir el proceso para obtener las latitudes, cambiando en la expresión anterior `[0]` por `[1]`, y luego separar los campos por resultado, utilizando como separador `;`.

Debe tener en cuenta que, como se mencionó antes, cuantos más datos se provean al servicio de GEOLocate en la consulta más sencillo será desglosar los resultados después. El proceso de desglose puede ser muy engorroso y requiere que sea muy meticuloso/a a la hora de nombrar campos y separar contenido. Si no está familiarizado/a con el uso de JSON, es preferible que realice el desglose "pasito a pasito" para evitar perder o mezclar información. Por ejemplo, puede crear un documento con el flujo de trabajo donde enumere los pasos a seguir con todos los detalles necesarios (incluya allí el tipo de resultados que espera ver y cómo se verían en los campos).

A la hora de agregar datos de georreferenciación, contraste siempre los resultados contra los campos geográficos que tiene. En el caso de tener varios resultados posibles, no siempre el primer resultado es el correcto. Recuerde reportar cuál fue el proceso de georreferenciación utilizado y todos los parámetros posibles asociados. Para consultar en qué campos de Darwin Core se reporta cada parámetro, puede referirse a: https://dwc.tdwg.org/terms/#location, y consultar: https://github.com/tdwg/dwc-qa/wiki/Georeferences.

Para conocer más acerca de georreferenciación y las mejores prácticas asociadas, consulte Georeferencing Best Practices (Chapman & Wieczorek 2020).

4.3. Limpieza de fechas utilizando Canadensys Date Parsing

4.3.1. Breve introducción

Uno de los campos sobre el que se puede corroborar la calidad de los datos es el campo de fecha: `"eventDate"`.

Recordemos primero la **definición de "eventDate" en el estándar Darwin Core**:

> The date-time or interval during which an Event occurred. For occurrences, this is the date-time when the event was recorded. Not suitable for a time in a geological context. Recommended best practice is to use a date that conforms to ISO 8601-1:2019.

Si piensa en un ejemplar de museo, `"eventDate"` refiere a cuándo fue colectado el ejemplar. Si piensa en una observación, `"eventDate"` refiere a cuándo fue realizada esa observación.

Darwin Core sugiere que se utilice para capturar la información de fecha el estándar ISO 8601-1:2019. Para fechas únicas, este estándar tiene el siguiente formato:

```
AAAA-MM-DDTHH:mmX
```

Donde:

- `AAAA`: año, con cuatro dígitos.
- `MM`: mes, con dos dígitos. E.g.: mayo sería 05.
- `DD`: día, con dos dígitos. E.g.: segundo día de un mes sería 02.
- `T`: indica que lo que viene a continuación es la hora.
- `HH`: horas, con dos dígitos, en formato de 24 hs.
- `mm`: minutos, con dos dígitos.
- `X`: indica la zona horaria. La zona horaria se determina tomando como base UTC (Coordinated Universal Time). Si uno está justo sobre la zona horaria UTC, X se reemplaza por "Z". Si uno está en otra zona horaria, debe reemplazarse X por la diferencia horaria correspondiente.

Por ejemplo, Argentina es UTC-3, o sea, 03horas00minutos al oeste (-) de UTC, por lo cual X debe reemplazarse por "-0300".

De este formato, uno puede utilizar tanto el formato completo (incluyendo la hora) como sólo la primera parte, AAAA-MM-DD.

Este formato también puede utilizarse para expresar rangos de fecha de manera estandarizada. Para ello, se usa el mismo formato y se separan las fechas con barras "/", ver ejemplos abajo (Tabla 3).

Tabla 3. Ejemplos de formatos de fechas y su versión estandarizada.

Fecha original	**Fecha estandarizada**
12 Feb 1809	1809-02-12
12/02/1809	1809-02-12
Jun 1906	1906-06
1971	1971

Fecha original	Fecha estandarizada
20 Feb 2009 8:40am UTC	2009-02-20T08:40Z
8 Mar 1963 2:07pm, en la zona horaria 6 horas más temprano que UTC	1963-03-08T14:07-0600
13-15 Nov 2007	2007-11-13/15
1 Mar 2007 1pm UTC – 11 May 2008 3:30pm UTC	2007-03-01T13:00:00Z/2008-05-11T15:30:00Z

4.3.2. Limpieza de fechas

Muchas veces, a pesar de lo que indica el estándar Darwin Core, encontramos en el campo `"eventDate"` fechas que no siguen el formato sugerido. Para estandarizarlas, puede hacer uso de la herramienta que ofrece Canadensys: Date Parsing.

Esta herramienta permite interpretar fechas, devolviéndolas en formato estándar. Ejemplos de los tipos de valores que puede interpretar son:

- Jun 13, 2008
- 15 Jan 2011
- 2009 IV 02
- 2 VII 1986

Algunas fechas, sin embargo no las interpreta, veamos el siguiente ejemplo (Figura 52):

Date parsing results

original	year	month	day	ISO 8601
2-4-1980 2/4/1980				
2/13/1980	1980	2	13	1980-02-13
13/2/1980	1980	2	13	1980-02-13

Figura 52

En las dos líneas inferiores, “13” sólo puede referir a días, pues no hay un mes “13”.

En las dos líneas superiores, en cambio, “2” y “4” pueden ambos referir a mes y día. Como en distintas partes del mundo se utilizan sistemas distintos (primero se pone día y luego mes, o viceversa), la herramienta no puede determinar inequívocamente cuál es cuál, y por ende no hace la interpretación.

Debe tener esto en cuenta cuando utilice la herramienta para limpiar los datos.

Ahora sí, invocaremos Date Parsing desde OpenRefine. Para ello, primero seleccione algunas fechas mediante una faceta, para reducir el tiempo de consulta. Luego, sobre la columna `"eventDate"` haga click en **la ▼ azul › Editar columna › Agregar columna accediendo a URLs...** (Figura 53). En la ventana que aparece, nombre la nueva columna (por ejemplo “Canadensys_eventDate”) y pegue en el cuadro de texto la siguiente expresión:

```
"http://data.canadensys.net/tools/dates.json?data="+escape(cells["eventDate"].value,"u
rl")
```

Esta expresión le indica a la herramienta que evalúe los valores del campo `"eventDate"` y que devuelva los resultados en formato JSON.

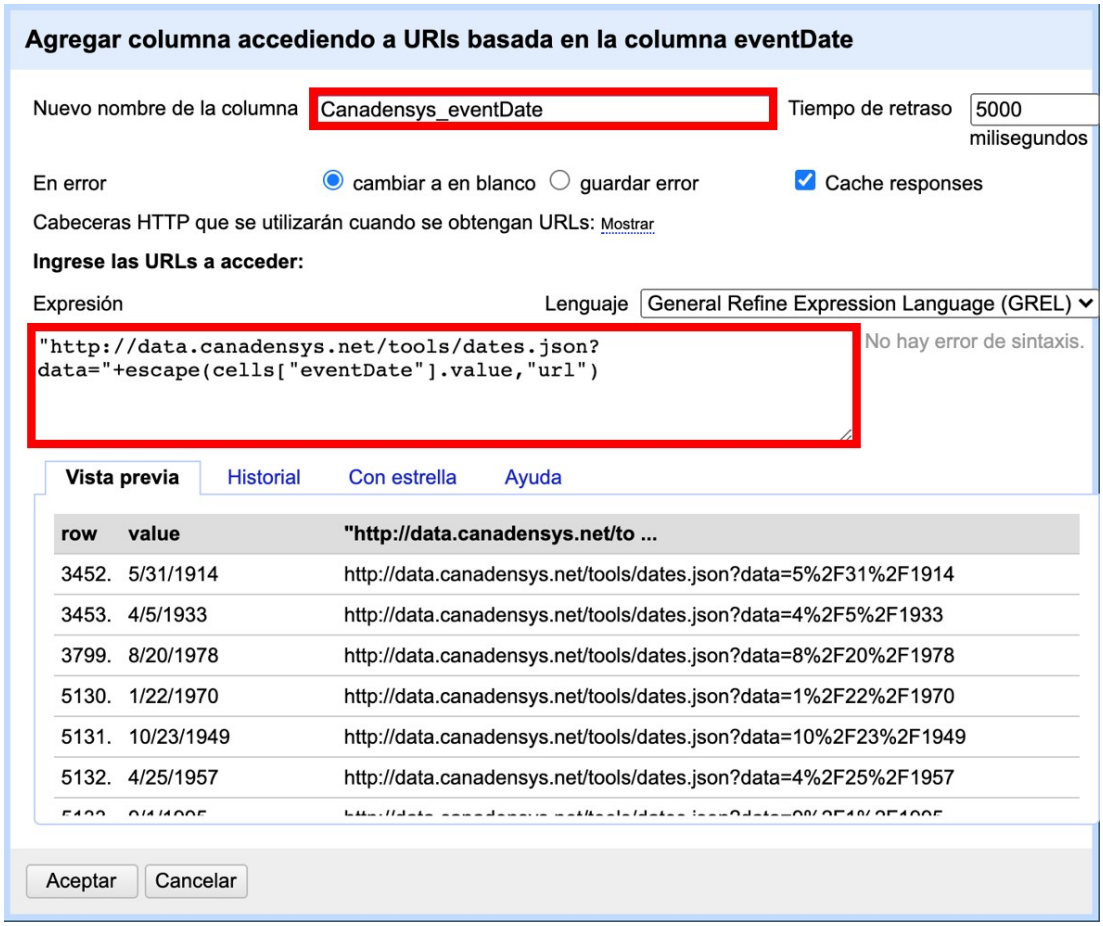

Figura 53

La limpieza puede tomar bastante tiempo, incluso horas, sea paciente… váyase a almorzar, o incluso a dormir y lo revisa al día siguiente… Cuando vuelva, encontrará el nuevo campo con los valores estandarizados! En formato JSON… (Figura 54).

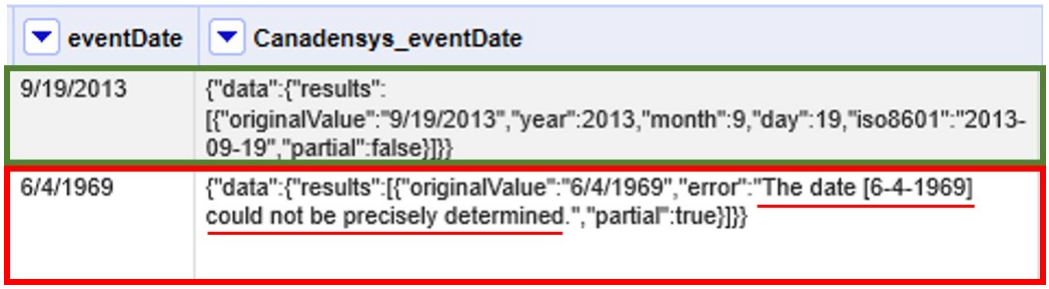

Figura 54

Fíjese que en el primer caso de la figura, Canadensys ha podido resolver la fecha, mientras que en el segundo caso no ha podido, dado que no puede interpretar inequívocamente “6” y “4” como día y mes o viceversa (como se explica más arriba). Ahora que tiene el resultado en formato JSON, extraeremos de allí los valores de interés. Podría extraer sólo la fecha en formato ISO, o también

año, mes y día en campos separados. Para ello, a partir de la columna que tiene el resultado en JSON, cree nuevas columnas: **Editar columnas › Agregar columna basada en esta columna** (Figura 55).

Para extraer sólo la fecha en formato ISO, en la ventana nombre la nueva columna (por ejemplo, "ISO_eventDate") y en el cuadro de texto pegue la siguiente expresión:

```
forEach(value.parseJson().get("data").get("results"),v,v.get("iso8601"))[0])
```

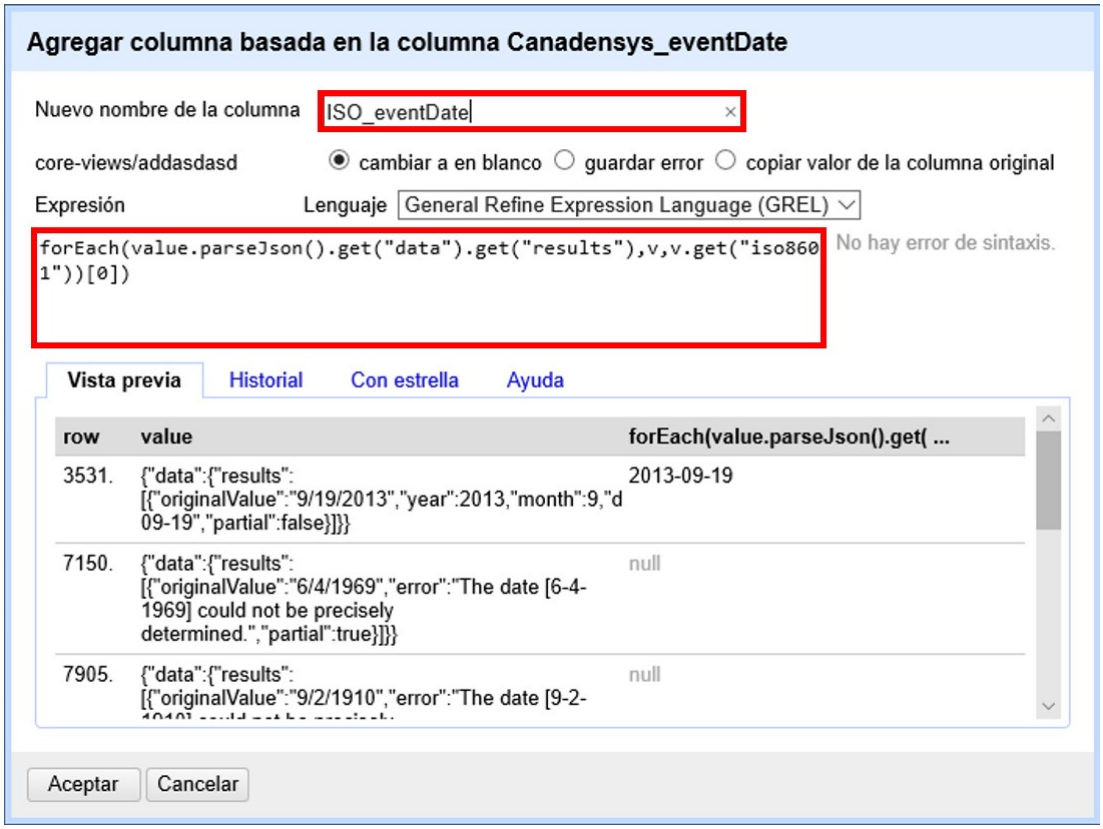

Figura 55

Para extraer el año, mes o día, pegue en cambio una de las siguientes expresiones:

- Año: `forEach(value.parseJson().get("data").get("results"),v,v.get("year"))[0])`
- Mes: `forEach(value.parseJson().get("data").get("results"),v,v.get("month"))[0])`
- Día: `forEach(value.parseJson().get("data").get("results"),v,v.get("day"))[0])`

Verá que algunos de los resultados serán nulos, éstos corresponden a los casos que Canadensys no ha podido resolver (como se explica más arriba) (Figura 56).

eventDate	Canadensys_eventDate	ISO_eventDate
9/19/2013	{"data":{"results":[{"originalValue":"9/19/2013","year":2013,"month":9,"day":19,"iso8601":"2013-09-19","partial":false}]}}	2013-09-19
6/4/1969	{"data":{"results":[{"originalValue":"6/4/1969","error":"The date [6-4-1969] could not be precisely determined.","partial":true}]}}	

Figura 56

Para terminar de limpiar las fechas, entonces, tendrá que revisar los valores que no hayan sido estandarizados por la herramienta. Para ello, sobre el campo ISO_eventDate puede armar una faceta y seleccionar el valor “blank”. Luego, arme una faceta sobre el campo `"eventDate"` (el que tenía los valores originales) y si estos son pocos, puede hacer un chequeo manual y completar el campo ISO_eventDate.

Transformación de fechas usando expresiones regulares

La consulta del servicio de Canadensys Date Parsing es particularmente útil cuando se tienen en un mismo campo fechas con distintos formatos, pues permite resolverlas todas utilizando un único proceso. En cambio, si todas las fechas están dadas en un único formato, resulta más rápido realizar transformaciones utilizando expresiones regulares. Para ello, a partir de la columna con fechas a estandarizar, por ejemplo `"eventDate"`, Hacer click en **la ▼ azul en el campo › Editar columnas › Agregar columna basada en esta columna...**. En la ventana que se abre, dar nombre al nuevo campo (por ejemplo “standard_eventDate”), y en el cuadro de texto utilizar una expresión como la siguiente, que convierte el formato original a una cadena de texto con formato “año-mes-día”:

```
value.toDate("formato_original").toString("yyyy-MM-dd")
```

donde en lugar de “formato_original” debe colocar el formato utilizado en el campo a estandarizar, considerando el orden en que aparecen el año, mes y día, el número de dígitos de cada uno y los separadores utilizados (e.g., guión, barra, etc.). Por ejemplo, si las fechas originales están todas escritas de la forma “mes/día/año”, la expresión a utilizar sería:

```
value.toDate("MM/dd/yyyy").toString("yyyy-MM-dd")
```

Con el formato especificado arriba, la expresión convertirá por ejemplo una fecha “04/25/1989” a “1989-04-25”.

Recordar que esta opción solo será útil aplicada a una columna si **todas** las fechas en esa columna tienen **el mismo formato**.

5. Rutinas de validación de la calidad de los datos

A partir del uso de servicios o archivos externos (ver sección 4) y la posibilidad que ofrece OpenRefine de guardar y rehacer pasos (ver sección 2.5.2) es posible crear rutinas para ejecutar de manera automática varias acciones de validación de calidad.

Aprovechando las múltiples herramientas de calidad de datos ya existentes en la red de GBIF es posible abordar de manera semi-automatizada a través de OpenRefine los retos y problemas más comunes de calidad que se presentan a nivel taxonómico y geográfico en un conjunto de datos. Acá se presentan diferentes rutinas que validan la calidad de los datos contrastando un conjunto de datos contra dichos servicios externos agilizando la obtención de resultados y asegurando una metodología de validación replicable.

5.1. ¿Cómo funcionan las rutinas?

Las rutinas comparan la información documentada en el conjunto de datos contra diferentes fuentes de referencia, y a partir de dicha comparación crean columnas de validación donde se puede identificar la correspondencia entre el archivo original y la fuente de referencia a través de operadores lógicos, unos (1) y ceros (0), que funcionan como indicadores de validación.

Los **indicadores de validación** se interpretan así:

- **0**: El valor documentado en el conjunto de datos NO coincide con la fuente de referencia, el valor debe ser revisado y ajustado **en caso de ser necesario**.
- **1**: El valor documentado en el conjunto de datos coincide con la fuente de referencia, no es necesario tomar acciones adicionales.

Observe el ejemplo de la Figura 57, en la primera fila el valor original de la columna `"family"` no coincide con la columna `"familySuggested"` ya que tiene un error de tipeo, por lo tanto el indicador de validación (columna `"familyValidation"`) es cero (0). Note que en las filas donde sí hay coincidencia el indicador de validación (`"familyValidation"`) es uno (1).

family	familyValidation	familySuggested
Poace	0	Poaceae
Poaceae	1	Poaceae
Poaceae	1	Poaceae

Figura 57

Las rutinas utilizan como fuentes de validación API's (Interfaces de Programación de Aplicaciones) de repositorios globales taxonómicos, geográficos o archivos de texto plano obtenidos como resultado

de herramientas de validación externas.

Se encuentran disponibles cinco (5) rutinas (Tabla 4) las cuales incluyen adicionalmente posibles escenarios al momento de realizar la validación, tal como la caída temporal de servicios web, o requisitos adicionales según la naturaleza de los datos (e.g. el grupo biológico de interés). A continuación se explica cada una:

Tabla 4. Lista de rutinas para la validación de datos primarios sobre biodiversidad

Nombre	Uso	Requerimientos
Validación taxonómica con el API de GBIF	Validación taxonómica que usa como referencia el árbol taxonómico de GBIF. Permite validar registros de varios grupos biológicos a la vez, así como obtener la taxonomía superior de cada taxa.	Requiere como mínimo los elementos DwC `"scientificName"` y `"kingdom"` documentados y acceso a internet para hacer la petición al API de GBIF.
Validación taxonómica con Species Matching de GBIF	Validación taxonómica que usa como referencia el árbol taxonómico de GBIF, a diferencia de la rutina anterior realiza la validación contra el archivo de resultados *normalized* obtenido de *Species Matching* permitiendo así aprovechar las funcionalidades de validación y limpieza de esta herramienta. La rutina facilita el cruce de los resultados obtenidos con *Species Matching* con el conjunto de datos original.	Requiere como mínimo el elemento DwC `"scientificName"` documentado y que el archivo *normalized* sea previamente cargado en OpenRefine para la ejecución de la rutina.
Validación taxonómica con el API de WoRMS	Validación taxonómica específica para organismos marinos, que usa como referencia el árbol taxonómico de LifeWatch (LW-SIBb) por medio de la API de WoRMS (*World Register of Marine Species*). Permite obtener la taxonomía superior de cada taxa, así como elementos taxonómicos obligatorios para la publicación de datos a través de OBIS.	Requiere como mínimo el elemento DwC `"scientificName"` documentado y acceso a internet para hacer la petición al API de WoRMS.

Nombre	Uso	Requerimientos
Validación de elevaciones con el API de GeoNames	Validación y/o obtención de la elevación a partir de las coordenadas usando el servicio geográfico de GeoNames.	Requiere los elemento DwC `"decimalLatitude"` y `"decimalLongitude"` documentados adecuadamente y acceso a internet para hacer la petición al API de GeoNames.
Transformación de fechas con el API de Canadensys	Transformación de fechas en múltiples formatos al estándar ISO 8601.	Requiere el elemento DwC `"eventDate"` documentado y acceso a internet para hacer la petición al API de Canadensys.

Las rutinas cuya fuente de referencia es un API, hacen una consulta a un servicio externo y obtienen una respuesta en formato JSON, la rutina interpreta esta respuesta y la hace legible en forma de columnas dentro del conjunto de datos. Posteriormente el resultado de la consulta al API es comparado con el valor documentado en el conjunto de datos y se generan nuevas columnas con los indicadores de la validación (unos y ceros). Las rutinas que usan como fuente archivos de texto plano, hacen una consulta sobre un archivo cargado previamente en OpenRefine que posteriormente es comparado con el valor documentado en el conjunto de datos. Como resultado de la comparación se generan nuevas columnas con los indicadores de la validación.

Todas las rutinas se ejecutan de manera similar, los detalles específicos para cada una se explican más adelante. En esta sección se presentan instrucciones generales para su ejecución en OpenRefine:

Paso 1

Carga de los archivos en OpenRefine

Cree un proyecto en OpenRefine con el conjunto de datos que desea validar, si tiene dudas sobre cómo hacerlo revise la sección 1. Asegúrese que el conjunto de datos o los elementos que desea validar estén estructurados en el estándar Darwin Core, si no lo están ajuste el nombre de cada columna según el estándar siguiendo las instrucciones de la sección 2.1.1.

Según la validación que desee realizar asegúrese de cumplir los requerimientos de la rutina. Si la rutina lo requiere cargue también en OpenRefine los archivos adicionales de validación (ver Tabla 2.), de lo contrario vaya directamente al paso 2.

Paso 2

Ejecutar la rutina

Ubique en esta guía la rutina de interés según la validación que desee realizar, haga clic en el enlace a la rutina y será redirigido a GitHub donde encontrará un archivo de texto plano con la rutina, copie el texto de la rutina de validación (Figura 58). Asegúrese de seleccionar solo la rutina -sin las instrucciones- y copiar todos los corchetes iniciales { y finales }.

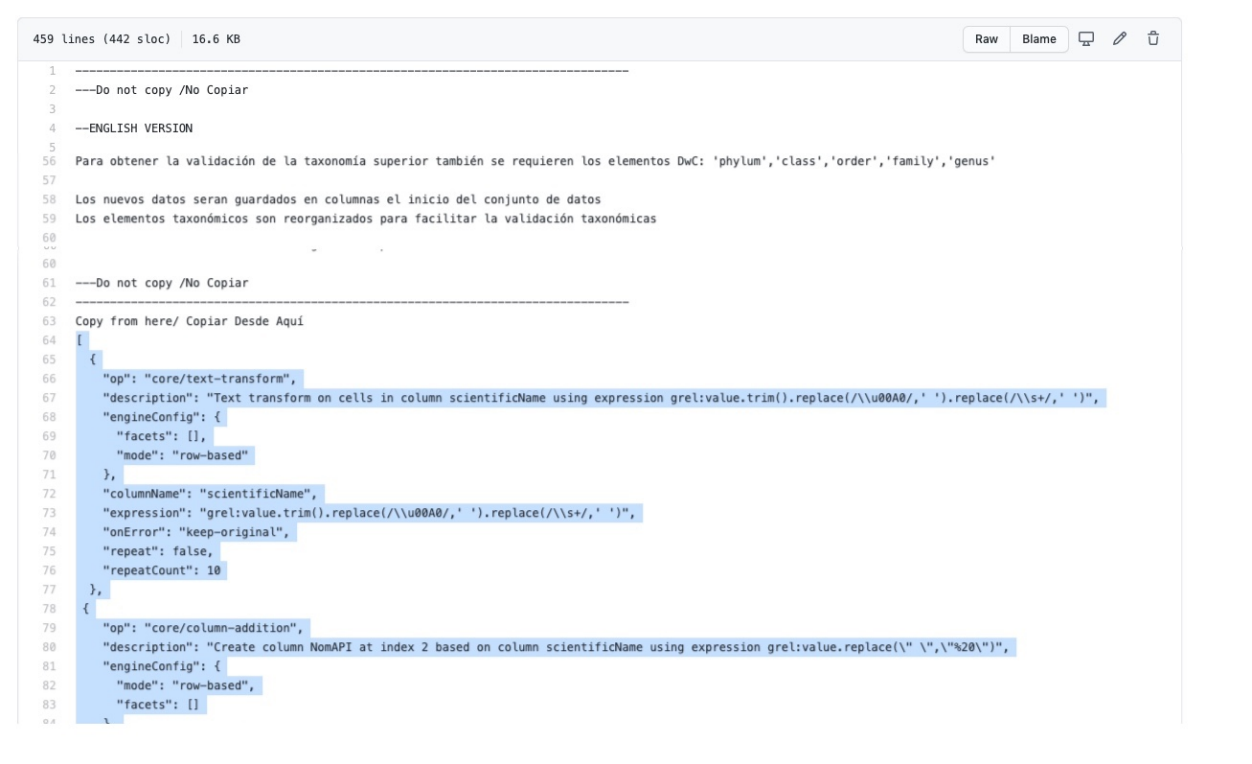

Figura 58

Ubíquese en el conjunto de datos a validar en OpenRefine, diríjase al menú arriba a la izquierda, seleccione la pestaña “Deshacer/Rehacer” y haga clic en el botón “Aplicar...”. A continuación se abrirá una ventana de texto vacía, pegue en el cuadro de texto la rutina a ejecutar y haga clic en “Ejecutar Operaciones” (Figura 59). Si tiene dudas sobre este proceso revise la sección 2.5.

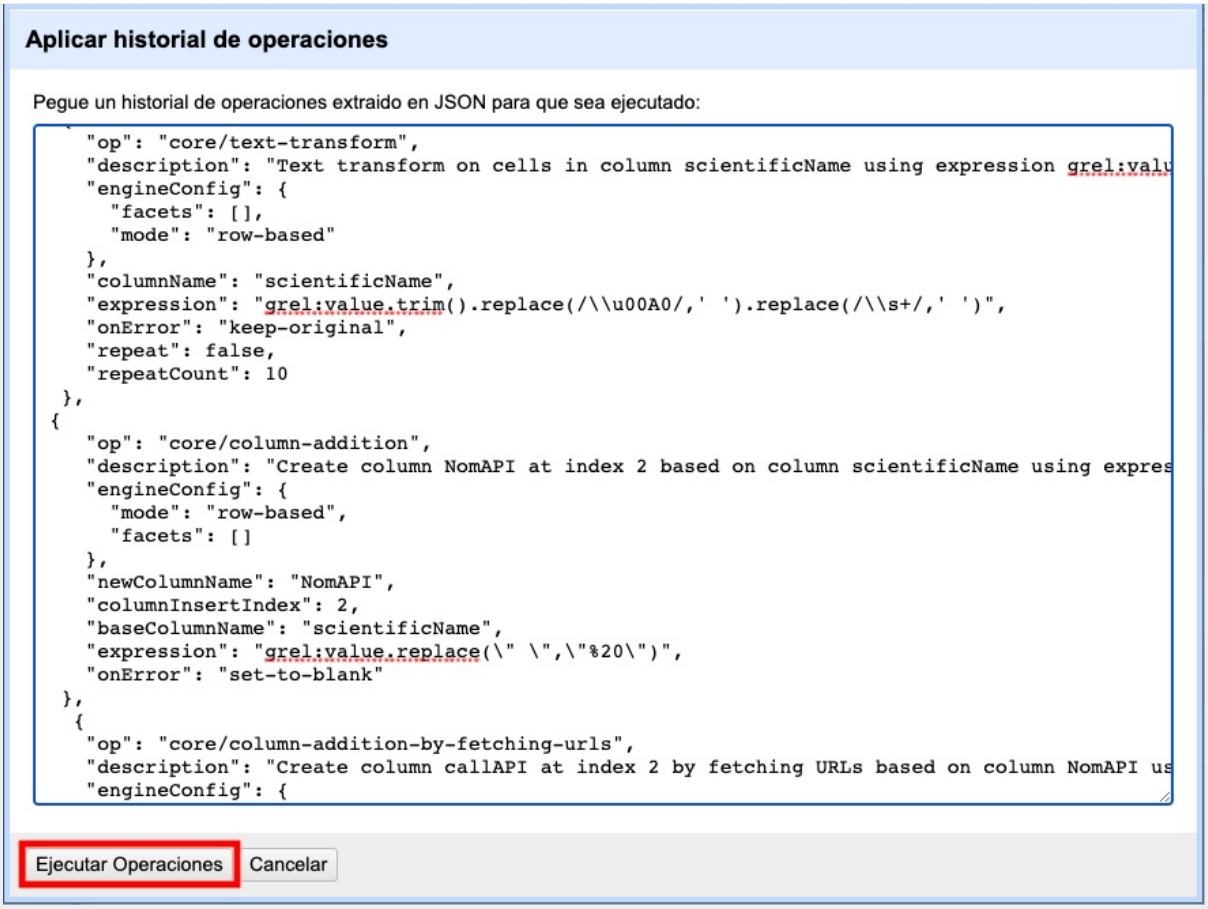

Figura 59

El avance de la ejecución de la rutina se observa en la parte superior de la pantalla (Figura 60).

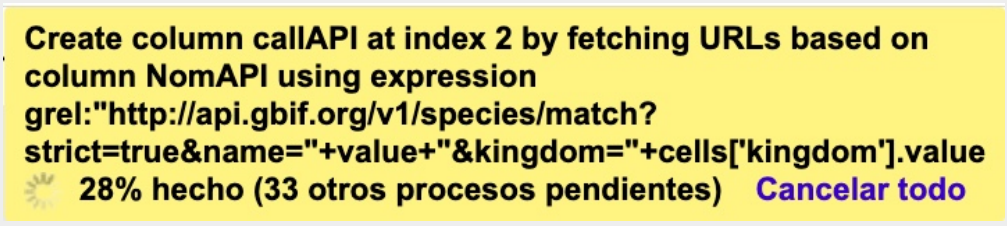

Figura 60

Espere a que finalice la ejecución de la rutina. Las rutinas que requieren hacer llamados a servicios externos, dependen de la conexión a internet, estas consultas toman un tiempo en ejecutarse que varía según el número de filas del conjunto de datos, de la velocidad de la conexión y de la memoria RAM de su equipo.

Paso 3

Resultados de la validación

Al terminar la ejecución de la rutina, obtendrá nuevas columnas en el conjunto de datos, puede identificarlas por su terminación:

- ***Suggested***: Valores sugeridos resultantes de la validación con las fuentes de referencia, dependiendo de la rutina seleccionada pueden ser sugerencias taxonómicas, geográficas, o temporales.
- ***Validation***: Corresponde a los indicadores de validación (unos y ceros) que permiten rastrear diferencias entre el valor original y el valor sugerido, y realizar posteriormente una limpieza de los datos.

En la Figura 61 se muestra un ejemplo de cómo se ven los identificadores de la validación y las nuevas columnas con las sugerencias después de ejecutar la rutina; en el ejemplo se observa una validación taxonómica, las columnas de resultado varían según el objetivo de cada rutina.

order	orderValidation	orderSuggested	family	familyValidation	familySuggested
Poales	1	Poales	Poace	0	Poaceae
Poales	1	Poales	Poaceae	1	Poaceae
Poales	1	Poales	Poaceae	1	Poaceae
	0	Malvales	Poaceae	0	Malvaceae
Poales	1	Poales	Poaceae	1	Poaceae

Figura 61

Paso 4

Limpieza de los datos

A partir de las nuevas columnas de validación (finalizadas en las palabra *Suggested*) seleccione los registros donde el valor original y el valor sugerido son diferentes (identificador de validación = 0) y realice los ajustes que considere necesarios sobre los elementos del estándar Darwin Core. Se recomienda realizar este proceso de limpieza utilizando las funcionalidades de OpenRefine descritas en la Sección 2 de limpieza de datos. El proceso de validación con las rutinas busca facilitar la identificación de filas y elementos que necesitan ser verificados y limpiados, sin embargo, **un identificador de validación con valor cero (0) no necesariamente implica que haya un error en los datos. Cada publicador según su conocimiento de los datos y del grupo biológico debe determinar si los datos se deben ajustar y cómo.**

Por ejemplo de la Figura 62 se muestra una Faceta de texto que permite seleccionar las filas cuyo indicador de validación es cero (0) para el elemento de familia y por lo tanto necesita ser verificado. En la primera fila se muestra una inconsistencia entre la familia documentada en el conjunto de datos original y la sugerida por la rutina, mientras en la segunda fila se evidencia un problema de tipeo. **En cada caso debe revisar de manera integral cada fila y decidir qué ajuste se debe o no realizar**.

Figura 62

Tenga en cuenta que los identificadores de validación no cambiarán (de 0 a 1) automáticamente así usted haya ajustado los datos originales según las sugerencias de la rutina. Cámbielos manualmente cuando realice la limpieza de cada fila indistintamente del ajuste realizado.

Una vez terminada la validación y limpieza de sus datos, puede eliminar las columnas resultantes de la validación (finalizan en las palabras *Validation* y *Suggested*) y dejar solo las columnas corregidas de su archivo original.

5.2. Validación taxonómica con el API de GBIF

Enlace a la rutina:

https://github.com/SIB-Colombia/data-quality-open-refine/blob/master/ValTaxonomicAPIGBIF_ValTaxonomicaAPIGBIF.txt

Requerimientos:

- El conjunto de datos a validar debe tener como mínimo los elementos DwC `"scientificName"` y `"kingdom'"` documentados.
- Si también desea validar la taxonomía superior de su conjunto de datos se requieren los elementos DwC: `"scientificName"`, `"kingdom"`,`"phylum"`,`"class"`,`"order"`,`"family"`, y `"genus"`.

Funcionamiento:

Esta rutina valida la información taxonómica de un conjunto de datos usando como referencia el árbol taxonómico de GBIF, esto se hace a través de un llamado al API de GBIF basado en los elementos del estándar Darwin Core `"scientificName"` y `"kingdom"` documentados en el conjunto de datos. Como resultado, el llamado retorna la taxonomía superior, nombres aceptados, estatus taxonómico y autoría del nombre científico de acuerdo al árbol taxonómico de GBIF. La rutina toma los valores obtenidos del árbol y los compara con los elementos documentados en el archivo base, generando los indicadores de validación.

Resultados:

En las primeras columnas del proyecto encontrará las columnas con los datos taxonómicos reorganizadas junto con nuevas columnas resultantes de la rutina. Primero encontrará las columnas asociadas al cruce con el árbol taxonómico y luego de manera intercalada columnas con el valor taxonómico original, un valor sugerido de acuerdo al árbol taxonómico de GBIF y el indicador de validación indicando si los valores son iguales (1) o difieren (0) como se muestra en la Figura 63.

class	classValidation	classSuggested	order	orderValidation	orderSuggested
Liliopsida	1	Liliopsida	Poales	1	Poales
Liliopsida	1	Liliopsida	Poales	1	Poales
Liliopsida	1	Liliopsida		0	Malvales
Liliopsida	1	Liliopsida	Poales	1	Poales

Figura 63

A continuación se listan las columnas que encontrará después de ejecutar la rutina:

- `taxonMatchType`: Indica el resultado del cruce de los datos originales con el árbol taxonómico de GBIF a partir de los elementos `"scientificName"` y `"kingdom"`. Los valores que encontrará en esta columna son:
  - EXACT: La correspondencia entre el `"scientificName"` del conjunto de datos y el árbol taxonómico es completa.

    - FUZZY: La correspondencia entre el `"scientificName"` del conjunto de datos y el árbol taxonómico es parcial, el nombre difiere en su escritura. Comúnmente indica errores de tipeo o diferencias por correcciones nomenclaturales (ejem: la terminación `i` vs. `ii` cuando la especie se dedica a una persona).
    - HIGHERRANK: La correspondencia entre el nombre científico del conjunto de datos y el árbol taxonómico fue parcial. No se identificó el taxon a nivel taxonómico de `"scientificName"` si no a un nivel superior. Por ejemplo si el `"scientificName"` corresponde a una especie, la correspondencia con el árbol taxonómico de GBIF fue a nivel de género. Esto sucede porque el taxon aún no está en el árbol taxonómico de GBIF o por errores de tipeo mayores.
    - NONE y BLANK: La correspondencia entre el `"scientificName"` del conjunto de datos y el árbol taxonómico fue **nula** o **hubo varias coincidencias** con muy poca información para determinar un resultado, esto sucede comunmente cuando hay homónimos o si el taxon aún no se encuentra en el árbol taxonómico de GBIF, como es el caso de especies recientemente descritas o algunas endémicas.
- `"scientificName"`: Columna original del conjunto de datos.
- `"acceptedScientificName"`: Nombre científico aceptado según el árbol taxonómico de GBIF.
- `"canonicalNameSuggested"`: Nombre canónico sugerido según el árbol taxonómico de GBIF.
- `"taxonRankSuggested"`: Categoría del taxon sugerido según el árbol taxonómico de GBIF (e.g.: SPECIES, GENUS, FAMILY).
- `"taxonomicStatusSuggested"`: Estado del taxon sugerido según el árbol taxonómico de GBIF (e.g.: ACCEPTED, SYNONYM).
- Tripleta de elementos validados donde se encuentra la columna original del conjunto de datos, la columna de validación y la columna con la sugerencia según el árbol taxonómico, por ejemplo: `"class"`,`"classValidation"`,`"classSuggested"`. Los siguientes elementos de estar documentados en el conjunto de datos original tendrán dicha tripleta: `"scientificNameAuthorship"`, `"kingdom"`, `"phylum"`, `"class"`, `"order"`, `"family"`, `"genus"`, `"specificEpithet"`
- `callAPI`: Respuesta del API a la rutina, contiene todos los resultados en formato JSON.

El llamado al API permite hacer una consulta sobre un número ilimitado de registros, sin embargo si su conjunto de datos tiene muchas filas se recomienda ejecutar la rutina sobre nombres científicos únicos, lo cual disminuirá el tiempo de respuesta y agilizará la ejecución de la rutina.

5.3. Validación taxonómica con Species Matching de GBIF

Enlace a la rutina:

https://github.com/SIB-Colombia/data-quality-open-refine/blob/master/ValTaxonomicSpeciesMatchGBIF_ValTaxonomicaSpeciesMatchGBIF.txt

Requerimientos:

- El conjunto de datos a validar debe tener como mínimo el elemento DwC `"scientificName"`

documentado.

- Si también desea validar la taxonomía superior de su conjunto de datos se requieren los elementos DwC: `"scientificName"`, `"kingdom"`,`"phylum"`,`"class"`,`"order"`,`"family"`, y `"genus"`.
- Archivo titulado *normalized*, obtenido de la herramienta *Species Matching* tras validar los datos originales, y cargado en OpenRefine, el título del proyecto debe ser exactamente ***normalized***.

El archivo *normalized* debe ser el único proyecto en OpenRefine titulado de esta manera. Cambie el nombre de cualquier otro archivo *normalized* cargado previamente, de lo contrario la rutina no podrá identificar adecuadamente el archivo de referencia.

Funcionamiento:

La rutina obtiene y valida la información taxonómica de un conjunto de datos con el árbol taxonómico de GBIF a partir del archivo de texto plano *normalized* obtenido de la herramienta en línea *Species Matching* y cargado en OpenRefine. La rutina retorna la taxonomía superior, nombres aceptados, estatus taxonómico y autoría del nombre científico de acuerdo al árbol taxonómico de GBIF y los compara con los elementos documentados en el archivo base, generando los indicadores de validación.

Al usar *Species Matching* como fuente de referencia, el usuario puede realizar una validación y limpieza previa a OpenRefine directamente en *Species Matching*, la cual es especialmente útil para verificar y resolver sinonimias complejas, como es el caso de los homónimos.

A diferencia del API de GBIF, *Species Matching* tiene un límite de consulta de 6.000 registros o nombres científicos. Para evitar exceder el límite de consulta, se recomienda hacer la consulta en *Species Matching* por nombres científicos únicos.

Resultados:

Como en la rutina anterior, en las primeras columnas del proyecto encontrará de manera intercalada una columna con el valor taxonómico original, un valor sugerido de acuerdo al árbol taxonómico de GBIF y el indicador de validación indicando si los valores son iguales (1) o difieren (0) como se muestra en la Figura 63. Obtendrá las mismas columnas que en la rutina anterior menos la columna `"callAPI"`.

5.4. Validación taxonómica con el API de WoRMS

Enlace a la rutina:

https://github.com/SIB-Colombia/data-quality-open-refine/blob/master/ValTaxonomicAPIWoRMS_ValTaxonomicaAPIWoRMS.txt

Requerimientos:

- El conjunto de datos a validar debe tener como mínimo el elemento DwC `"scientificName"` documentado.

- Si también desea validar la taxonomía superior de su conjunto de datos se requieren los elementos DwC: `"scientificName"`, `"kingdom"`,`"phylum"`,`"class"`,`"order"`,`"family"`, y `"genus"`.

Funcionamiento:

Esta rutina está diseñada para ser implementada en conjuntos de datos de grupos biológicos marinos, emplea como fuente de referencia los taxones marinos del árbol taxonómico de LifeWatch (LW-SIBb) a través de un llamado al API de WoRMS (*World Register of Marine Species*). La rutina retorna la taxonomía superior, nombres aceptados, estatus taxonómico y autoría del nombre científico de acuerdo al árbol taxonómico de LifeWatch y los compara con los elementos documentados en el archivo base, generando los indicadores de validación.

Adicionalmente a los elementos taxonómicos, esta rutina retorna otros elementos útiles que dan información sobre el tipo de hábitat del taxon y el LSID de WoRMS o AphiaID, elemento requerido para la publicación de datos a través de OBIS (Ocean Biodiversity Information System).

Resultados:

En las primeras columnas del proyecto encontrará de manera intercalada una columna con el valor taxonómico original, un valor sugerido de acuerdo al árbol taxonómico y el indicador de validación indicando si los valores son iguales (1) o difieren como se muestra en la las rutinas previas (Figura 63).

A continuación se listan las columnas que encontrará despues de ejecutar la rutina, adicionales a las ya mencionadas en las rutinas previas de validación taxonómica (Figura 64):

- `"matchType"`: Indica el resultado del cruce de los datos originales con el árbol taxonómico de WoRMS a partir del elemento `"scientificName"`. Los valores que encontrará en esta columna son:
  - `"exact"`: La correspondencia entre el `"scientificName"` del conjunto de datos y el árbol taxonómico es completa.
  - `"phonetic"`: La correspondencia entre el `"scientificName"` del conjunto de datos y el árbol taxonómico es completa a nivel fonético a pesar de algunas diferencias menores en la escritura.
  - `"near_1"`: Hay una diferencia de un carácter entre el `"scientificName"` del conjunto de datos y el árbol taxonómico. Es una correspondencia bastante confiable.
  - `"near_2"`: Hay una diferencia de dos caracteres entre el `"scientificName"` del conjunto de datos y el árbol taxonómico. Se sugiere una revisión del nombre.
  - `"near_3"`: Hay una diferencia de tres caracteres entre el `"scientificName"` del conjunto de datos y el árbol taxonómico. Se requiere una revisión del nombre.
  - Para otras posibilidades poco frecuentes como`"match_quarantine"` y`"match_deleted"`, WoRMS recomienda contactarlos directamente.
- `"scientificNameID"`: Identificador del taxón construido a partir del AphiaID proveniente del árbol taxonómico de WoRMS.
- `"nameAccordingTo`: La referencia bibliográfica del nombre científico según WoRMS
- `"nameAccordingToID`: Identificador de la referencia bibliográfica del nombre científico según WoRMS.

- `"isMarine"`: Valor booleano (TRUE o FALSE) que indica si el registro corresponde a un taxon marino.
- `"isBrackish"`: Valor booleano (TRUE o FALSE) que indica si el registro corresponde a un taxon de aguas salobres.
- `"isFreshwater"`: Valor booleano (TRUE o FALSE) que indica si el registro corresponde a un taxon de aguas continentales, i.e. taxones asociados a ríos o lagos.
- `"isTerrestial"`: Valor booleano (TRUE o FALSE) que indica si el registro corresponde a un taxon terrestre.
- `"callAPIworms"`: Respuesta del API a la rutina, contiene todos los resultados en formato JSON.

nameAccordingTo	nameAccordingToID	isMarine	isBrackish	isFreshwater	isTerrestrial
Hoeksema, B. W.; Cairns, S. (2021). World List of Scleractinia. Favia De Blainville, 1820. Accessed through: World Register of Marine Species at: http://www.marinespecies.org/aphia.php?p=taxdetails&id=718691 on 2021-02-18	http://www.marinespecies.org/aphia.php?p=taxdetails&id=718691	1	0	0	0
Van Soest, R.W.M.; Boury-Esnault, N.; Hooper, J.N.A.; Rützler, K.; de Voogd, N.J.; Alvarez, B.; Hajdu, E.; Pisera, A.B.; Manconi, R.; Schönberg, C.; Klautau, M.; Kelly, M.; Vacelet, J.; Dohrmann, M.; Díaz, M.-C.; Cárdenas, P.; Carballo, J.L.; Ríos, P.; Downey, R.; Morrow, C.C. (2021). World Porifera Database. Amphimedon viridis Duchassaing & Michelotti, 1864. Accessed through: World Register of Marine Species at: http://www.marinespecies.org/aphia.php?p=taxdetails&id=166701 on 2021-02-18	http://www.marinespecies.org/aphia.php?p=taxdetails&id=166701	1	0	0	0

Figura 64

5.5. Validación de elevaciones con el API de GeoNames.

Enlace a la rutina:

https://github.com/SIB-Colombia/data-quality-open-refine/blob/master/ValElevationAPIGeoNames_ValElevacionAPIGeoNames.txt

Requerimientos:

- El conjunto de datos a validar debe tener como mínimo los elemento DwC `"decimalLatitude"` y `"decimalLongitude"` documentados adecuadamente.
- Tener una cuenta activa en GeoNames, si no tiene una regístrese aquí antes de correr la rutina.

Funcionamiento:

Antes de ejecutar la rutina remplace la palabra *demo* en la expresión `username=demo` por su nombre de usuario en GeoNames, por ejemplo `username=rartizgt`. Si ejecuta la rutina sin hacer este cambio utilizará la opción de prueba (`demo`) incorporada por defecto en la rutina, la cual tiene un límite de 20.000 consultas **diarias mundiales**, por lo que puede que el servicio esté agotado y no obtenga resultados.

La rutina captura la elevación a partir de las coordenadas decimales documentadas en los elementos

DwC `"decimalLatitude"` y `"decimalLongitude"` del archivo base, a través de una consulta a los servicios de GeoNames. La rutina se ejecuta sobre valores únicos de pares de coordenadas para evitar superar el límite de consultas diarias por usuario.

La rutina utiliza por defecto el modelo de elevación SRTM-1 (`"srtm1"`), que cuenta con una resolución aproximada de 30 metros. Sin embargo, el usuario puede usar otro de los modelos de elevación disponibles:

- SRTM3 (`"srtm3"`): Datos de elevación de la *Shuttle Radar Topography Mission (SRTM)*, con resolución aproximada de 90 x 90 metros.
- Astergdemv2 (`"astergdem"`): Datos de elevación del *Aster Global Digital Elevation Model V2* (2011) con resolución aproximada de 30 x 30 metros.
- GTOPO30 (`"gtopo30"`): Modelo de elevación global con resolución aproximada de 30 arcos por segundo, equivalente a una grilla de 1 km x 1 km.

Para cambiar el modelo de elevación reemplace en la rutina el valor `srtm1` en la expresión `grel:\"http://api.geonames.org/srtm1` por el valor que corresponda al servicio que desea utilizar `srtm3`, `astergdem` o `gtopo30`.

Resultados:

En las primeras columnas del proyecto encontrará las columnas con los datos de elevación reorganizadas junto con nuevas columnas resultantes de la rutina. Encontrará de manera intercalada las columnas originales, un valor sugerido de acuerdo al servicio de elevación y dos indicadores de validación (Figura 65). El primer indicador contrasta la elevación obtenida con el servicio y el elemento `"minimumElevationInMeters"` y debe ser interpretado así:

- **1**: La diferencia entre la elevación en `"minimumElevationInMeters"` y `"elevationSuggested"` es menor a 100 m.
- **0**: La diferencia entre la elevación en `"minimumElevationInMeters"` y `"elevationSuggested"` es mayor a 100 m.
- blank: No hay elevación mínima documentada.

El segundo indicador contrasta la elevación obtenida con el servicio contra el rango de elevación indicado por los elementos `"minimumElevationInMeters"` y `"maximumElevationInMeters"` y debe ser interpretado así:

- **1**: El rango de elevaciones contiene la elevación sugerida.
- **0**: El rango de elevaciones NO contiene la elevación sugerida.

minimumElevationInMeters	maximumElevationInMeters	elevationSuggested	elevationValidation	elevationRangeValidation
edit 1086	1087	1286	0	0
1200	1300	1286	1	1
1200	1250	1286	1	0
1086	1336	1286	0	1

Figura 65

Si las coordenas se encuentran sobre plataforma marina, puede que reciba como resultado valores negativos (ej. -1, -3), o valores como: "/home/data/srtm1/N02/N02W080.zip" o "No data".

5.6. Transformación de fechas con el API de Canadensys

Esta rutina recopila los pasos de la sección 4.3 y automatiza su ejecución para el mismo procedimiento.

Enlace a la rutina:

https://github.com/SIB-Colombia/data-quality-open-refine/blob/master/DateTransform_TransformFechas.txt

Requerimientos:

- El conjunto de datos a validar debe tener como mínimo el elemento DwC `"eventDate"` documentado.

Funcionamiento:

A partir de la fecha documentada en el archivo base en el elemento `"eventDate"` se realiza una consulta al API de Canandensys que retorna las fechas transformadas al estándar ISO 8601. A diferencia de las rutinas anteriores el objetivo de esta rutina es transformar las fechas, por ello no retornará identificadores de validación.

Resultados

En las primeras columnas del proyecto encontrará las columnas con los datos temporales reorganizadas junto con nuevas columnas resultantes de la rutina.

A continuación se listan las columnas que encontrará después de ejecutar la rutina:

- `"eventDateSuggested"`: Fecha transformada al estándar ISO 8601.
- `"yearSuggested"`: Año extraído a partir de la transformación de la fecha.
- `"monthSuggested"`: Mes extraído a partir de la transformación de la fecha.
- `"daySuggested"`: Día extraído a partir de la transformación de la fecha.
- `"verbatimEventDateSuggested"`: Fecha en el formato original.

Para no generar conflicto con elementos ya existentes en el conjunto de datos, todas las columnas generadas por la rutina se marcan como sugeridas o *Suggested* (Figura 66). Si algún registro no tiene datos de fecha, los elementos resultantes aparecerán vacíos.

verbatimEventDateSuggested	eventDateSuggested	yearSuggested	monthSuggeste	daySuggested
2 VII 1986	1986-07-02	1986	07	02
15 Jan 2011	2011-01-15	2011	01	15
1999/02/24	1999-02-24	1999	02	24
Feb/17/1921	1921-02-17	1921	02	17
May/17/2017	2017-05-17	2017	05	17
Marzo/17/2017	2017-03-17	2017	03	17
Septiembre/17/2017	2017-09-17	2017	09	17

Figura 66

Los formatos de fechas que son ambiguos, es decir donde no se diferencia con claridad el mes, el día o el año, no son transformados. Revise las celdas donde el resultado haya sido nulo o vacío y realice los ajustes necesarios de forma manual.

Epílogo

Agradecimientos

Los autores agradecen especialmente a David Bloom (VertNet) y Anabela Plos (GBIF Argentina, Museo Argentino de Ciencias Naturales “Bernardino Rivadavia”) por sus comentarios sobre versiones anteriores de esta guía. Esta guía ha sido actualizada en el marco de la iniciativa de actualización de documentación del Secretariado del Global Biodiversity Information Facility (GBIF). Este documento se ha actualizado en parte a partir de su uso en distintos cursos de entrenamiento organizados por diversas instituciones, entre las que se cuentan GBIF Argentina - Museo Ciencias Naturales “Bernardino Rivadavia”, GBIF España - Real Jardín Botánico de Madrid, y el SiB Colombia incorporando los comentarios de los respectivos alumnos. Las consultas, comentarios y sugerencias de Susana Devincenzi (IANIGLA-CONICET), usuaria incansable de OpenRefine, han sido de particular utilidad para mejorar esta guía. Los valiosos comentarios y sugerencias durante la etapa de revisión comunitaria de la guía por Carole Sinou (Canadensys) y Florencia Grattarola (Biodiversidata) permitieron mejorar el contenido de este documento. Se extienden los agradecimientos a otras personas que también contribuyeron de una u otra manera en esta versión del documento: Katia Cezón, Kyle Copas, Matt Blissett, Mélianie Raymond, Laura Russell, Néstor Beltrán y miembros del Panel Editorial de Documentación de GBIF.

Apéndice 1: instalación de OpenRefine

Estas instrucciones han sido adaptadas y traducidas al castellano a partir de un instructivo preparado por el GBIF-BID Programme.

Requerimientos

1. Java JRE instalado.
2. Navegador de internet instalado (preferentemente Google Chrome).

Para instalar **OpenRefine 3.3** en su computadora, siga los siguientes pasos:

Instalación en MS Windows

1. Descargue aquí el kit de Windows.
2. Descomprima, y copie la carpeta en su computadora. Abra la carpeta y haga doble click en openrefine.exe. Si encuentra algún problema en este punto, haga doble click sobre refine.bat.
3. Aparecerá una ventana de comando (que no debe cerrar) e inmediatamente después su navegador web mostrará una nueva ventana con la aplicación.

Instalación en Mac

1. Descargue aquí el kit de Mac.
2. Abra y arrastre el icono en la carpeta Applications.
3. Haga doble clic en él y su navegador web mostrará una nueva ventana con la aplicación.

Existe también una versión para Linux.

Para saber más

- Puede encontrar instrucciones adicionales para la instalación y consultar funciones básicas en el Manual del Usuario de OpenRefine.
- Si desea instalar otras versiones del programa, puede encontrarlas en la página de Descargas de Open Refine.
- OpenRefine permite además trabajar con **Extensiones** creadas por miembros de la comunidad, que proveen diversas funciones. Puede consultar aquí la Lista de Extensiones disponibles con sus descripciones y las Instrucciones para Instalar Extensiones.