

OpenRefine



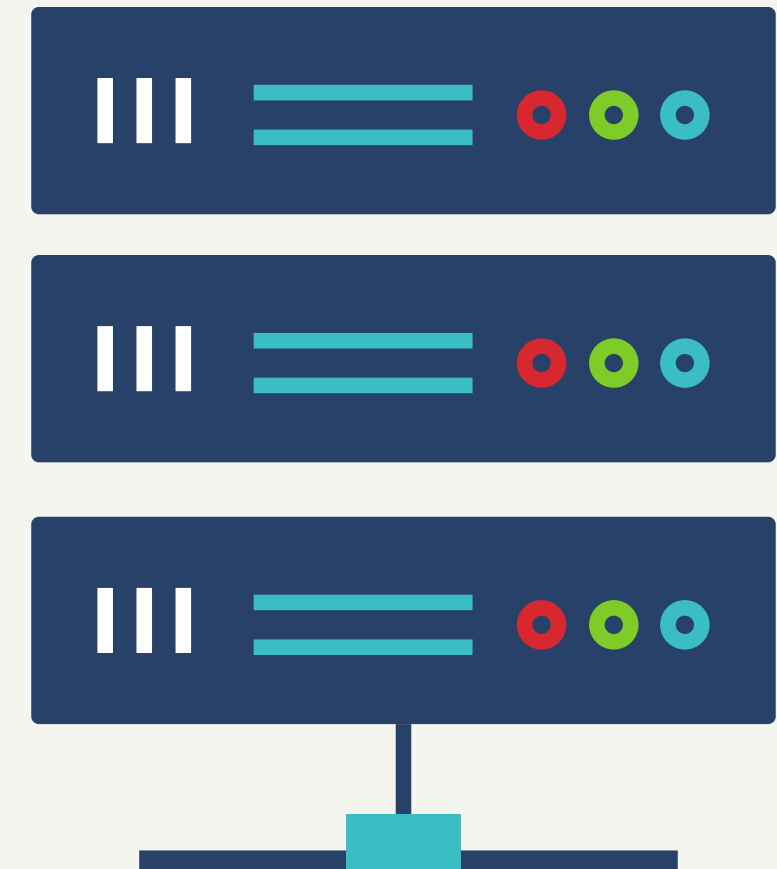
La aplicación de escritorio que nos ayuda a  
limpiar nuestros datos



# Expresiones Regulares (Regex)

Secuencias de caracteres compuestas por un patrón empleadas para **buscar** o **buscar y reemplazar** palabras en un texto

OpenRefine funciona  
mediante un servidor en  
local (localhost) escrito  
en Java y ejecutando por  
la terminal de nuestro SO





.JSON

OpenRefine permite importar  
y exportar ficheros en  
diferentes formatos



.CSV



# Data Types

string

"I'm a string"

number

40

boolean

true / false

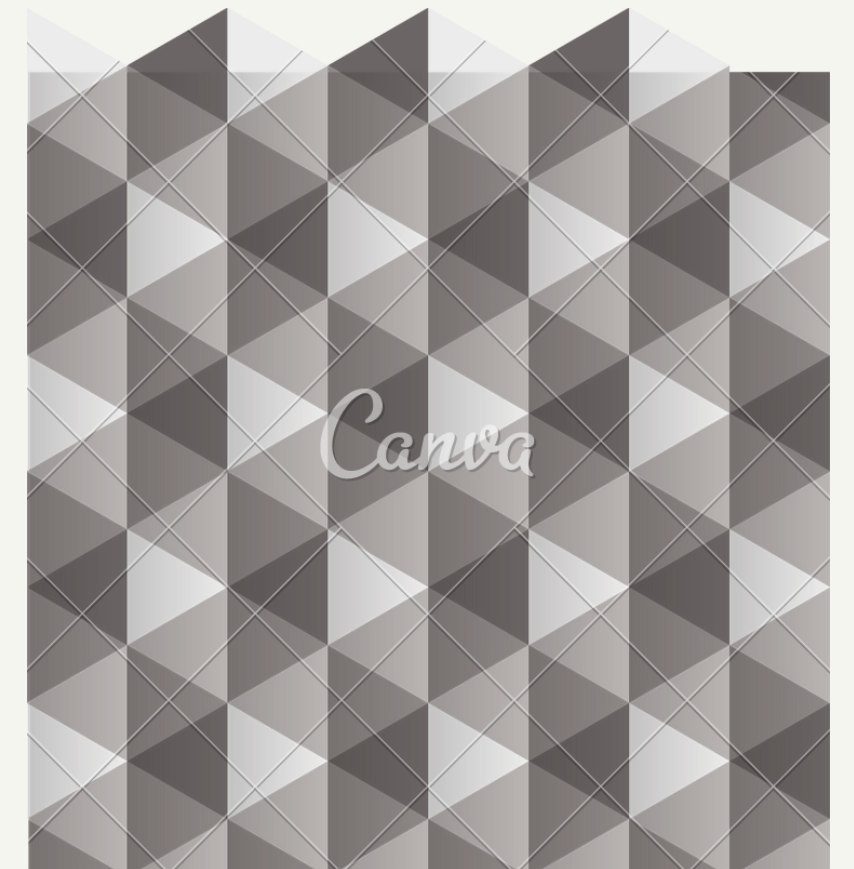
date

YYYY-MM-DDTHH:MM:SSZ



# Facets

Permiten buscar patrones y  
tendencias en diferentes tipos de  
datos

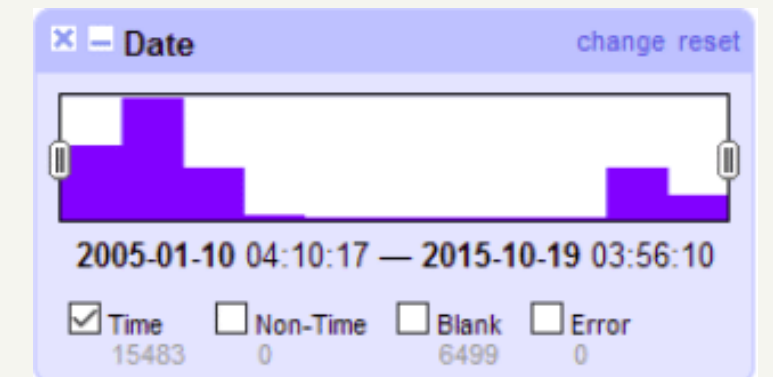
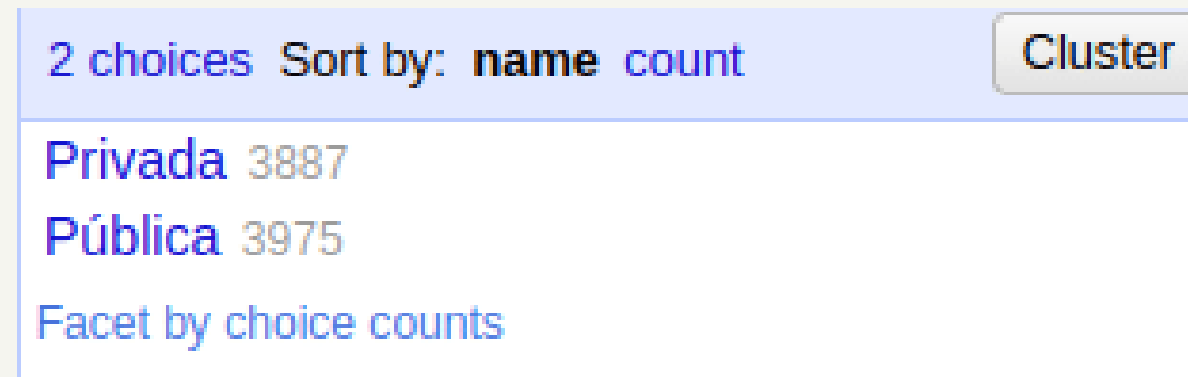
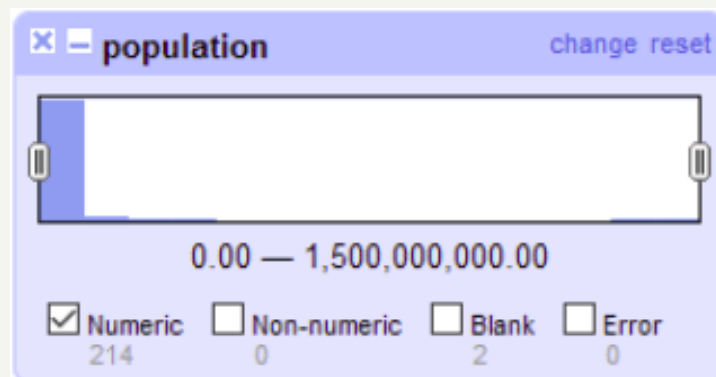


# Facets

Numeric

Text

Date





# Sort

Se puede ordenar en base  
a cualquier tipo de dato  
ASC/DESC

**Sort by tipo\_establecimiento**

Sort cell values as

☒ text ☐ case-sensitive

☐ numbers

☐ dates

☐ booleans

Position blanks and errors

Valid values

Errors

Blanks

Drag and drop to re-order

☒ a - z ☐ z - a

OK Cancel

# Transform data

split

**Split column Location into several columns**

**How to Split Column**

☒ by separator  
Separator  ☐ regular expression  
Split into  columns at most (leave blank for no limit)

☐ by field lengths

List of integers separated by commas, e.g., 5, 7, 15

**After Splitting**

☒ Guess cell type  
☒ Remove this column

OK Cancel

join

**Join columns**

Select and order columns to join

- ☒ Location
- ☐ Location\_Code
- ☐ Property\_Size\_Ha
- ☐ Parish\_Code
- ☐ Parish\_Extension\_Code
- ☒ District
- ☒ Parish
- ☐ Livestock\_Name
- ☐ Livestock\_Type\_Code

Select All De-select All

Select options

Separator between the content of each column:   
Enter one or more characters, or keep blank to join the columns without separator.

☒ Replace nulls with...   
Enter one or more characters, or keep blank to replace nulls with blank strings.  
☐ Skip nulls.

☐ In separator and nulls substitutes, use \n for new lines, \t for tabulation, \\n for \n, \\t for \t.

☒ Write result in selected column.  
☐ Write result in new column named...

☐ Delete joined columns.

OK Cancel

# Variables

Variable	Meaning
<code>value</code>	The value of the cell in the current column of the current row (can be null)
<code>row</code>	The current row
<code>row.record</code>	One or more rows grouped together to form a record
<code>cells</code>	The cells of the current row, with fields that correspond to the column names (or <code>row.cells</code> )
<code>cell</code>	The cell in the current column of the current row, containing value and other attributes
<code>cell.recon</code>	The cell's reconciliation information returned from a reconciliation service or provider
<code>rowIndex</code>	The index value of the current row (the first row is 0)
<code>columnName</code>	The name of the current cell's column, as a string

# GREL

El lenguaje de expresiones que  
nos permite trabajar con todos  
los valores de OpenRefine

# Anatomía de una expresión

```
value.expresion(<arguments>, [options])
```

```
if(<statement>, true, false)
```

# forEach

```
forEach(<iterable>, <element>, <condition>)
```

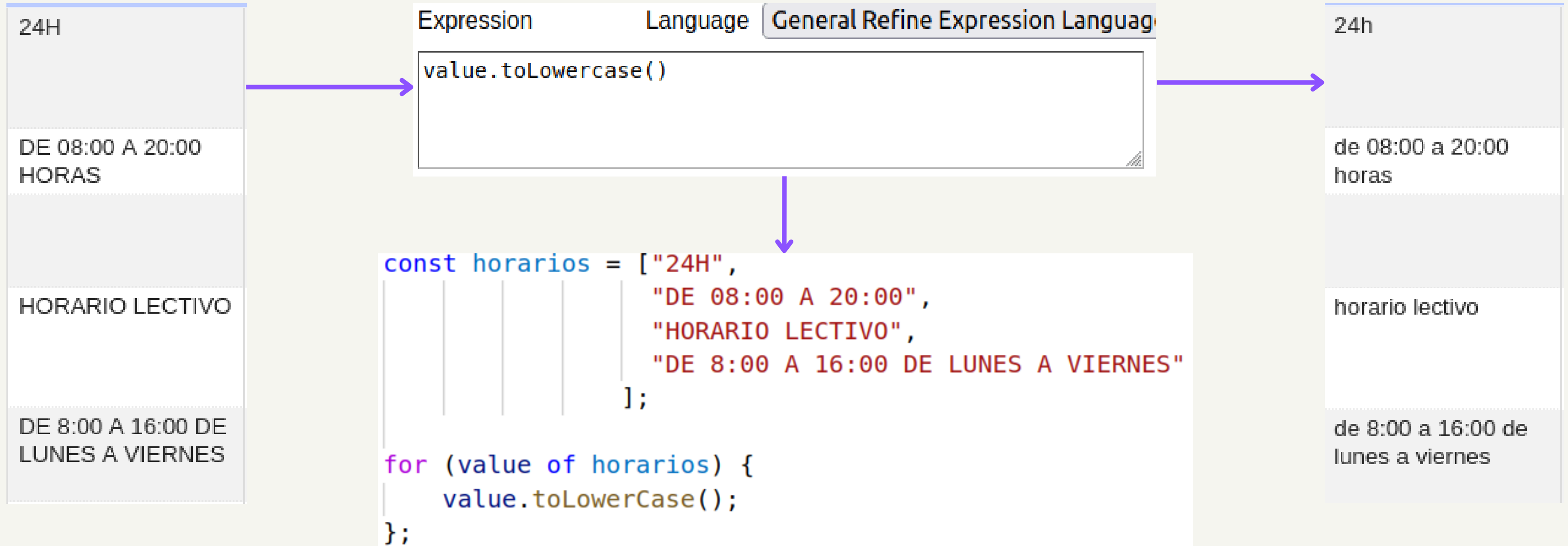
```
forEach([1,2,3,4], v, mod(v,2) == 0)  
// [false, true, false, true]
```

# forEach

```
filter(<iterable>, <element>, <condition>)
```

```
filter([1,2,3,4], v, mod(v,2) == 0)  
// [2, 4]
```

# Lowercase transformation





# Operators

Arithmetic

+, -, \*, /, pow

Comparison

<, >, >=, ==, !=

Logical

and, or, not

# Concat

El uso del operador de suma (+) entre `strings` es denominado concatenación

`value + "!"`



`frase!`

# RegEx

El idioma de las serpientes



# Character classes

<code>\d</code>	Matches a single digit character
<code>\w</code>	Matches a single word character (letters, numbers, and underscore)
<code>\s</code>	Matches a single white space character, including tabs and line breaks
<code>\D</code>	Matches a single non-digit character
<code>\W</code>	Matches a single non-word character (letters, numbers, and underscore)
<code>\S</code>	Matches a single non-white space character, including tabs and line breaks
<code>.</code>	Matches any single character

# Quantifier

<code>*</code>	Indicates zero or more
<code>+</code>	Indicates one ore more
<code>?</code>	Indicates zero or one
<code>x{n}</code>	Used to specify the number of times ('n') the previous character ('x') should appear
<code>x{n, }</code>	Used to specify the minimum number of times ('n') the previous character ('x') should appear
<code>x{n, m}</code>	Used to specify the minimum ('n') and maximum ('m') number of times the previous character ('x') should appear