XMLtoCSV.jl Documentación

XMLtoCSV es un script que permite procesar múltiples archivos XML para unificarlos en un archivo CSV.

Registro de paquetes

```
using Mongoc, CSV, CSVFiles, LightXML, DataFrames, DataStructures;
```

Función addRow

```
function addRow(df, row)
```

Parámetros

- df dataFrame donde se almacenará el contenido xml
- row contenido xml que se agregará

Qué hace la fucnión

compruebe si el nuevo contenido xml tiene los mismos nombres de columnas que el dataFrame

```
finalRow = Dict()
   for name in names(df)
      if haskev(row, name)
          finalRow[name] = row[name]
          delete!(row, name)
          finalRow[name] = "NaN"
```

Qué retorna la función

Returns a new dataFrame with the new xml content added

```
for missingKey in keys(row)
   insertcols!(df, missingKey => "NaN")
   finalRow[missingKey] = row[missingKey]
push!(df, finalRow)
return df
```

Función getAnswers

```
function getAnswers(survey, document)
```

Parámetros

- survey xml document from which the content will be extracted document name of the file

Qué hace la fucnión

Agregue columnas fijas a un diccionario ordenado.

```
Survey= OrderedDict(
  "filePath" => document["file-path"],
  "latitude" => string(document["latitude"]),
  "longitude" => string(document["longitude"]),
  "workshopKey" => document["workshop-key"],
  "surveyKey" => document["survey-key"],
  "capturedTimestamp" => string(document["captured-timestamp"]),
  "storedTimestamp" => string(document["stored-timestamp"])
```

Extraiga el contenido del documento xml de acuerdo con las etiquetas de respuesta y crea columnas en el diccionario de acuerdo con la identificación de la pregunta

```
for topic in get_elements_by_tagname(survey, "topic")
     for answer in get\_elements\_by\_tagname(topic, "question")
         push!(Survey, "Q$(parse(Int, attribute(answer, "id")))" => content(find_element(answer, "answer")) )
     end
 end
 return Survey
```

Oué retorna la función

Devuelve un diccionario ordenado con el contenido xml.

Implementación de código

```
for document in occurrences
    doc = string(evidencesPath, document["file-path"])
    survey = root(parse_file("$doc"))
    if size(data)==(0, 0) || !(document in data[:,1])
        addRow(data, getAnswers(survey, document))
    end
end
CSV.write(output, data; delim=';')
println("--Julia--> preprocessed CSV was updated.")
```

Diagrama de flujo

evidences_preprocessor.jl

