Ciclo de vida de los datos: PEC 1

Francisco Alonso Sarria y Jose Luis García Bravo

1 Contexto

Vamos a trabajar con el Portal Estadístico de la Región de Murcia, sitio web del Centro de Estadística de la Región de Murcia (CREM) dependiente de la Comunidad Autónoma de la Región de Murcia (CARM). Se trata de un sitio que alberga 4012 tablas con datos socioeconómicos de interés regional organizadas en 115 grandes temas.

La página https://econet.carm.es/web/crem/informacion-de-la-a-z (figura 1) actúa como índice a partir del cual se puede acceder a cualquiera de estos grandes temas.

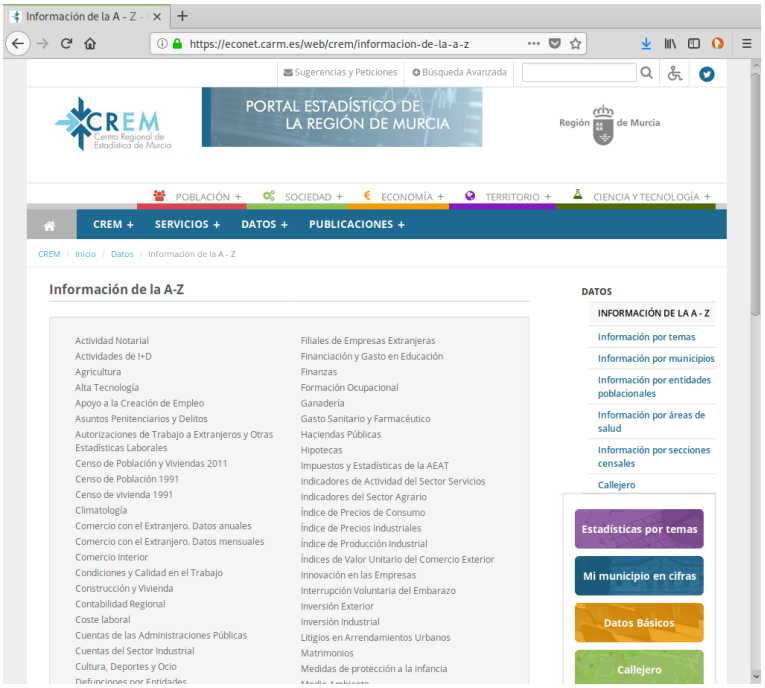


Figura 1: Página principal con los enlaces a los grandes temas

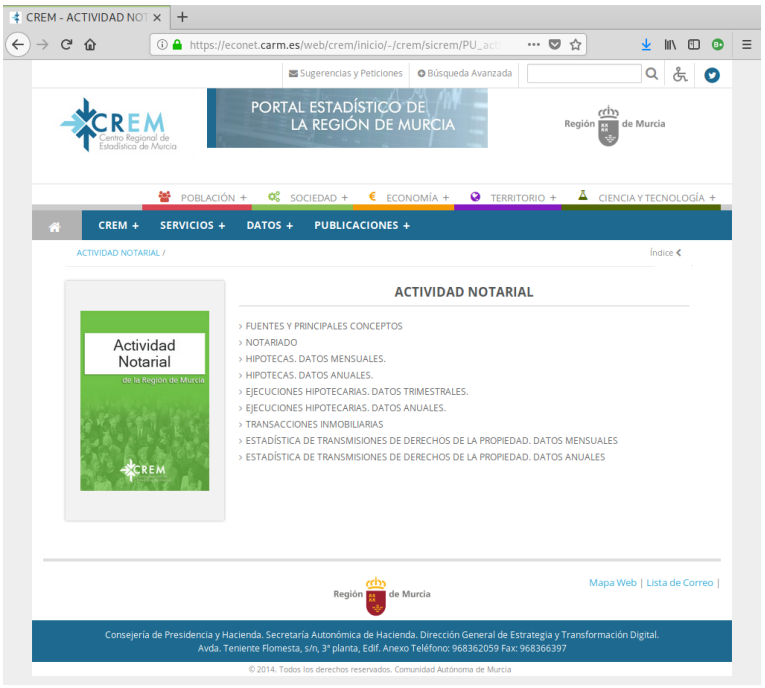


Figura 2: Página del primer gran tema con los enlaces a los subtemas

Al pinchar en alguno de los grandes temas vamos a otra página donde se desglosan los subtemas corrrespondientes (Figura 2). Al pinchar en alguno de estos enlaces, éste se desglosa en nuevos enlaces (figura 3) que nos llevan a las tablas (figura 4). Se trata por tanto, en principio, de una estructura en 3 niveles. Sin embargo, solo en algunos casos, aparece un cuarto nivel.

2 Título

Recopilación de datos estadísticos de la Región de Murcia.

3 Descripción del dataset

Como resultado de la práctica hemos recopilado las 4281 tablas en un fichero en formato JSON (`CREM.json`). La figura que aparece en el apartado de representación gráfica muestra la estructura de este archivo.

Lo acompaña un fichero `estructura.csv` que contiene 4281 filas (una por tabla) y 3 columnas que corresponden al tema principal, subtema y nombre de la tabla.

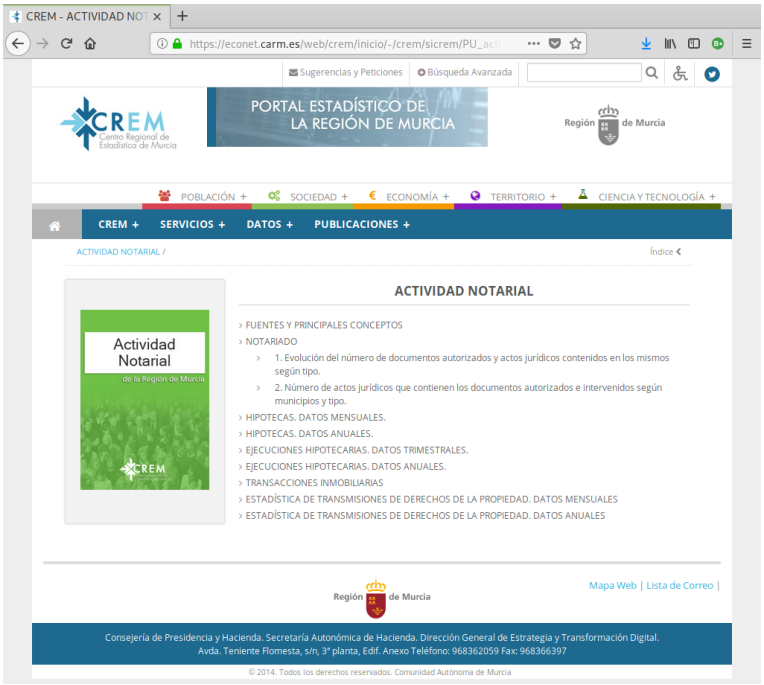
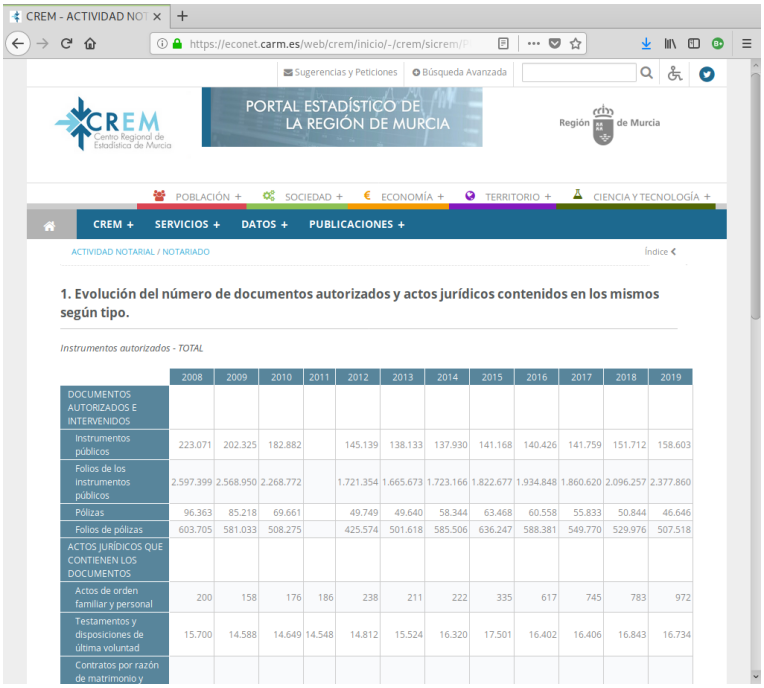


Figura 3: Página del primer subtema con con los enlaces a las tablas



	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
DOCUMENTOS AUTORIZADOS E INTERVENIDOS												
Instrumentos públicos	223.071	202.325	182.882		145.139	138.133	137.930	141.168	140.426	141.759	151.712	158.603
Folios de los instrumentos públicos	2.597.399	2.568.950	2.268.772		1.721.354	1.665.673	1.723.166	1.822.677	1.934.848	1.860.620	2.096.257	2.377.860
Pólizas	96.363	85.218	69.661		49.749	49.640	58.344	63.468	60.558	55.833	50.844	46.646
Folios de pólizas	603.705	581.033	508.275		425.574	501.618	585.506	636.247	588.381	549.770	529.976	507.518
ACTOS JURÍDICOS QUE CONTIENEN LOS DOCUMENTOS												
Actos de orden familiar y personal	200	158	176	186	238	211	222	335	617	745	783	972
Testamentos y disposiciones de última voluntad	15.700	14.588	14.649	14.548	14.812	15.524	16.320	17.501	16.402	16.406	16.843	16.734
Contratos por razón de matrimonio y												

Figura 4: Página con la primera tabla del primer subtema del primer gran tema

Adicionalmente, hemos realizado un subset que contiene únicamente las tablas municipales (`tablasMunicipales.`
en un único fichero con 45 filas que corresponden con los 45 municipios de la Región de Murcia y 3725 columnas. La primera columna contiene los nombres de los municipios y los nombres de las restantes columnas son simplemente `columna_X`, donde `X` es un número del 2 al 2725.

El fichero `clavesColumnas.csv` contiene una tabla con 3261 filas (una por cada columna en `tablasMunicipales.csv` y 2 filas. El primer campo contiene el nombre de columna que aparece en `tablasMunicipales.csv` y el segundo es una cadena de texto que incluye el tema, subtema, nombre de latabla y nombre de la columna correspondientes, separados por ";".

4 Representación gráfica

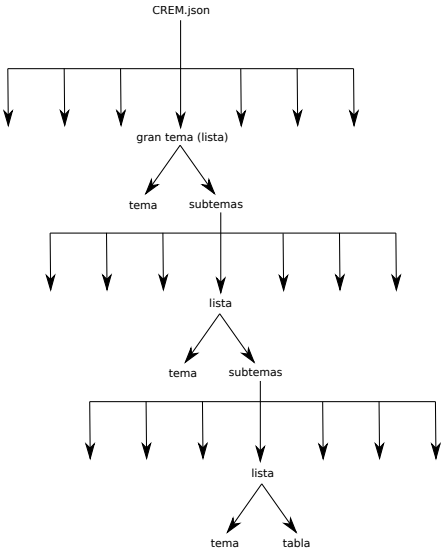


Figura 5: Representación de la lista que alberga todas las tablas extraidas

5 Contenido

En realidad son 4012 tablas por lo que no tiene sentido explicar todos los campos. En todo caso puede explorarse la estructura del fichero, sus tablas y columnas en la página web:

https://alonsarp.shinyapps.io/aplicacion

6 Agradecimientos

Los datos proceden de la página web del CREM, dependiente de la Dirección General de Estrategia y Transformación Digital de la Consejería de Presidencia y Haciendad de la Comunidad Autónoma de la Región de Murcia (CARM). Se trata del órgano responsable de la actividad estadística de interés para la CARM, a cuyo frente está José Blaya Verdú. Entre sus objetivos podemos destacar:

- Proponer, dirigir y coordinar la actividad estadística pública de interés para la CARM.
- Elaborar el anteproyecto del Plan Estadístico de la CARM y los proyectos de Programas Anuales de Estadística.
- Proponer normas, establecer requisitos y promover la coordinación metodológica con las unidades de estadística de otros organismos.
- Realizar las actividades estadísticas encomendadas por la legislación vigente.
- Promover la investigación estadística y la formación y el perfeccionamiento profesional del personal estadístico.
- Promover la difusión de las estadísticas relativas a la Región de Murcia.
- Informar todo proyecto que promueva o en el que participe la administración de la CARM para la realización de actividad estadística.
- Colaborar en el diseño y reforma de los procedimientos administrativos de la CARM que por su naturaleza puedan tener consecuencias en la producción de informaciones de posible utilización como fuente estadística.
- Promover, gestionar y centralizar la creación y mantenimiento de bancos de datos de carácter estadístico.
- Realizar investigaciones para contrastar la objetividad y corrección técnica de la metodología en las actividades estadísticas.
- Actividades de creación, elaboración, mantenimiento, actualización y tratamiento de los datos del padrón municipal de habitantes, registros de comercio exterior y contenido económico, directorios de instituciones, empresas y establecimientos que ejerzan su actividad en la Región, censos económicos y todas las estadísticas necesarias para la implantación de indicadores económicos.
- La carga de datos, actualización y mantenimiento de la base de datos estadísticos regional y municipal.

En 2010, el Centro Regional de Estadística, sustituyó el envío de publicaciones en papel por informes en su página web (https://econet.carm.es/). Estos informes se acompañan de un gran número de tablas que aparecen en diferentes páginas web y que han sido objeto de esta práctica.

7 Inspiración

Este dataset contiene todas las tablas que el CREM tiene en abierto en su dataset. Aunque las tablas son descargables, son muchas y se hace muy pesado descargarlas todas, además varían de año a año, por lo que tener un protocolo de descarga automatizado puede resultar útil.

Cualquier estudio económico, social, geográfico, etc. sobre la Región de Murcia puede beneficiarse de la explotación de estos datos. Aunque obviamente haría falta una fase de comprobación y limpieza de datos en las tablas involucradas.

Una idea interesante sería completar la aplicación shiny que hemos creado para permitir un descubrimiento y análisis exploratorio de los datos más amigable y orientado a aplicaciones concretas. La integración con información espacial para presentar como mapas de coropletas aquellas tablas que lo permitan sería otra opción a explorar.

8 Licencia

Para estos datoe elegimos la licencia: *Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)*

En la página web del CREM no aparece ninguna nota al respecto de la propiedad intelectual de estos datos. Tampoco hay ninguna restricción en el fichero robots.txt. Por ello consideramos estos datos como datos públicos y no consideramos adecuado añadir ninguna restricción, por tanto optamos por la licencia menos restrictiva siempre que no se haga un uso comercial de nuestro desarrollo.

9 Código

```
# Funcion que crea un data.frame con los nombres y los links presentes en obj vía el x
extraeLinks <- function(obj, xpath, urlBase=NULL) {
    nodos    = rvest::html_nodes(obj, xpath=xpath)
    nombres  = rvest::html_text(nodos)
    enlaces  = rvest::html_attr(nodos, "href")
    if (!is.null(urlBase)) enlaces = paste0(urlBase,"/",enlaces)
    return(data.frame(nombres=nombres, enlaces=enlaces))
}

# Funcion que reemplaza NA por val en un data.frame.
replace_na <- function(tabla,val) {
     for (c in 1:ncol(tabla)) {
        tabla[which(is.na(tabla[,c])),c] = val
     }
     return(tabla)
}
```

```
# Comprobamos en primer lugar que puedo proceder:

robotstxt::get_robotstxt(domain= "https://econet.carm.es")

#Leemos la página base que es la que contiene los enlaces a los grandes temas.

url0 <- "https://econet.carm.es/web/crem/informacion-de-la-a-z"
html0 <- xml2::read_html(url0)

#Los enlaces a grandes temas aparecen en dos columnas y a cada columna se accede con u

xpath1 = "//div/div/div[1]/div/div/div/div/div/div/div/div[1]/ul/li/a"
xpath2 = "//div[2]/ul/li/a"
nivel1 = rbind(extraeLinks(html0, xpath1),
               extraeLinks(html0, xpath2))

# Para cada uno de los grandes temas leemos el segundo nivel. Guardamos los enlaces ju

for (n1 in 1:nrow(nivel1)) {
    cat("Progreso:",n1, "/", nrow(nivel1), "\n")
    url1 = nivel1[n1,2]
    html1 <- xml2::read_html(url1)

    urlBase=url1
    if(grepl("html",url1)) {
      urlBase=paste0(head(strsplit(url1, "/", fixed=TRUE)[[1]],-1), collapse="/")
    }

    xpath = "//td/table/tbody/tr/td/a"
    niv2 = extraeLinks(html1, xpath=xpath, urlBase=urlBase)
    niv2 = niv2[grep("Indice",niv2[,2]),]  # Nos quedamos con enlaces a datos
    if (n1==1) {
        nivel2 = niv2
        temas1 = rep(n1, nrow(niv2))
    } else {
        nivel2=rbind(nivel2, niv2)
        temas1 = c(temas1, rep(n1, nrow(niv2)))
    }
    Sys.sleep(runif(1, min = 0.5, max = 3))
}

# Importamos los datos que hemos recopilado a mano con las particularidades de los dis
source("getnc.R")

# Extraemos el nivel 3 y el nivel 4 (que trataremos como un nivel 3)
for (n2 in 1:nrow(nivel2)) {

    cat("Progreso:",n2, "/", nrow(nivel2), "\n")
```

```
    if (n2==88) next # excepción
    url2 = nivel2[n2,2]

    urlBase=url2
    if(grepl("html",url2)) {
      urlBase=paste0(head(strsplit(url2, "/", fixed=TRUE)[[1]],-1), collapse="/")
    }

    html2 <- xml2::read_html(url2)
    xpath = "//td[2]/table/tbody/tr[2]/td/table/tbody/tr/td/table/tbody/tr/td/a"
    niv3 = extraeLinks(html2, xpath=xpath, urlBase=urlBase)

    # Pero que pasa cuando hay un enlace al cuarto nivel????
    # Cuales tienen 4º nivel pero no graficos?
    w = which(grepl("Indice",niv3[,2]) & !(grepl("GRÁFICOS",niv3[,1])))
    # para todos ellos, ábrelo y extrae las tablas.

    for (ww in w) {
       html3 <- xml2::read_html(niv3[ww,2])

       nt=getnt(n2,ww)

       xpath=paste0("//tr[",nt,"]/td[2]/table/tbody/tr/td/table/tbody/tr/td/a")
       if (n2==35 & ww==1) {
          xpath="//td/table/tbody/tr[2]/td[2]/table/tbody/tr/td/table/tbody/tr/td/a"
       }

       niv4 = extraeLinks(html3, xpath=xpath, urlBase=urlBase)
       niv3 = rbind(niv3, niv4)
    }
    niv3 = niv3[grep("sec|pagina",niv3[,2]),]  # Nos quedamos con enlaces a tablas

    if (n2==1) {
        nivel3 = niv3
        temas2 = rep(n2, nrow(niv3))
    } else {
        nivel3=rbind(nivel3, niv3)
        temas2 = c(temas2, rep(n2, nrow(niv3)))
    }
    Sys.sleep(runif(1, min = 0.5, max = 3))
}

tablas = list()
nt = 0

# Solo nos falta leer las tablas de datos, que se situan siempre en la tabla 4 o 6, y
for (n3 in 1:nrow(nivel3)) {
    cat(n3, "/", nrow(nivel3), "\n")
```

```
    url3 = nivel3[n3,2]
    html3 <- xml2::read_html(url3)
    nt = nt+1
    tbs = rvest::html_nodes(html3, "table")
    if(length(tbs)==0) next
    if (length(tbs)>=6) {
        tablas[[nt]] <- rvest::html_table(tbs[6])[[1]]
    } else {
        tablas[[nt]] <- rvest::html_table(tbs[4],fill = TRUE)[[1]]
    }
    Sys.sleep(runif(1, min = 0.5, max = 3))
}

# Un poco de data cleaning en las tablas

for (tb in 1:length(tablas)) {
    if (is.null(tablas[[tb]])) {
       cat(tb, "/", length(tablas), "\n")
    } else {
       # Convierto en numérico las celdillas que cumplen esta regExp
       rexpReal = "^(\\+|-)?((\\d*).?)*\\,?(\\d*)$"
       for (r in 1:nrow(tablas[[tb]])) { for (c in 1:ncol(tablas[[tb]])) {
           if (grepl(rexpReal, tablas[[tb]][r,c])) {
               tablas[[tb]][r,c] = gsub(",", ".",
                                        gsub(".","",tablas[[tb]][r,c], fixed=TRUE),
                                        fixed=TRUE)
           }
       }}
    }
}


#  Lo introducimos en un JSON

lista_temas1 = list()
nt = 0
for (n1 in 1:nrow(nivel1)) {
    lista_temas1[[n1]] = list(tema = nivel1[n1,1], subtemas=list())

    w = which(temas1==n1)
    for (n2 in 1:length(w)) {
        ll = list(tema=nivel2[w[n2],1], subtemas=list())
        lista_temas1[[n1]]$subtemas[[n2]] = ll

        w2 = which(temas2==w[n2])
        if (length(w2)>0) {
           for (n3 in 1:length(w2)) {
              nt = nt+1
```

```
            if (nt %in% c(1321,2021,2293)) {
               ll = list(tema=nivel3[w2[n3],1], tabla=NULL)
            } else {
               ll = list(tema=nivel3[w2[n3],1], tabla=tablas[[nt]])
               ll$tabla = replace_na(ll$tabla,-9999)
            }
            lista_temas1[[n1]]$subtemas[[n2]]$subtemas[[n3]] = ll

         }
      }
   }
}

# Las escribimos en un fichero

jsonlite::write_json(lista_temas1, "CREM.json")

# Creamos una supertabla con todos los datos para crear los ficheros de estructura y m

estructura = data.frame(nivel1 = nivel1[temas1[temas2],1],
                        nivel2 = nivel2[temas2,1],
                        nivel3 = nivel3[,1])

# Para arreglar inconsistencias en los nombres de algunos municipios.
tablas[[listaTab[34]]][17,1]="Caravaca de la Cruz"
tablas[[listaTab[35]]][17,1]="Caravaca de la Cruz"
tablas[[listaTab[39]]][17,1]="Caravaca de la Cruz"
tablas[[listaTab[39]]][40,1]="Torre-Pacheco"
for (k in c(45:47, 52:54,122)) {
   tablas[[listaTab[k]]][23,1]="Fuente Ãlamo"
   tablas[[listaTab[k]]][35,1]="Puerto Lumbreras"
}

# Creo la tabla de municipios
k=0
listaTab = c()
rm("supertabla")
for (nt in 1:length(tablas)) {
   cond1 = any(grep("olina",tablas[[nt]][,1]))
   cond2 = !is.null(nrow(tablas[[nt]]))
   if (cond1 & cond2) {
       if (nrow(tablas[[nt]])==46 & !(nt %in% c(1885:1886,2578,3587))) {
          k=k+1
          listaTab[k] = nt
          if (exists("supertabla")) {
              supertabla = merge(supertabla,tablas[[nt]],
                                 by.x=names(supertabla)[1],
                                 by.y=names(tablas[[nt]])[1])
```

```
            } else {
                supertabla = tablas[[nt]]
            }
            cat(k,nt,dim(tablas[[nt]]),dim(supertabla),"\n")
            if (nrow(supertabla)<45) break
        }
    }
}

# Pongo nombres a las columnas y preparo el dataframe con los nombres detallados
# de las columnas
ncols = 0
daf = data.frame(numcol="municipio", namecol="municipio")
for (nt in listaTab) {
    numcols = (ncols + 1):(ncols + ncol(tablas[[nt]]) -1)
    columnas = paste0( paste0(estructura[nt,],collapse=";"),
                       colnames(tablas[[nt]])[-1], sep=";")
    daf = rbind(daf,
                data.frame(numcol=paste0("columna_", numcols),
                           namecol=columnas))
    ncols = ncols + ncol(tablas[[nt]]) -1
}
names(supertabla) = c("municipio", paste("columna_",2:3260))

# Convierto valores numéricos
regexp = "^(\\d+)\\.?(\\d+)?$"
for (cl in 2:ncol(supertabla)) {
    if(all(grepl(regexp,supertabla[,cl]))) {
        supertabla[,cl] = as.numeric(supertabla[,cl])
    }
}

# Escribo los ficheros.
write.table(supertabla, "tablasMunicipales.csv", quote=FALSE, row.names=FALSE, sep=";"
write.table(daf, "clavesColumnas.csv", quote=FALSE, row.names=FALSE, sep=";")
write.table(estructura, "estructura.csv", quote=FALSE, row.names=FALSE, sep=";")
```

10 Dataset

Se puede consultar el estado actual del proyecto siguiendo el siguiente enlace a GitHub.

En el momento de la redacción del este documento, se generó un dataset almacenado en Zenodo debido a su gran tamaño que no permite incluirlo en el proyecto de GitHub.

En ese momento se realizo tambien un release del proyecto GitHub tambien almacenado en Zenodo.

El proyecto esta pendiente de una revisión exhaustiva de los datos para aclarar su completitud.

En las pruebas finales nos hemos dado cuenta de que si la estructura se modifica sustancialmente, los datos utilizados en el fichero getnc.R dejan de ser correctos y debe actualizarse manualmente. En las siguientes versiones intentaremos que este proceso se realice automáticamente.

En la página https://alonsarp.shinyapps.io/aplicacion se tiene acceso a todo el documento JSON mediante una aplicación shiny.

11 Contribuciones

Investigación previa	Francisco Alonso Sarria
Redaccion de las respuestas	Francisco Alonso Sarria y Jose Luis Garcia Bravo
Desarrollo del codigo	Francisco Alonso Sarria y Jose Luis Garcia Bravo