```r
### SCRIPT 3 - R ###
# Obtención de criterios de adjudicación desde ATOM
#
#########################################################
library(XML)
library(dplyr)
library(foreach)
library(doParallel)

#setup parallel backend to use many processors
cores=detectCores()
cl <- makeCluster(cores[1]-1, type = "PSOCK",outfile="log.txt") #not to overload your computer
registerDoParallel(cl)


# Cargar el fichero atom
setwd("C:/Users/guillermo.alonso/Desktop/Tesis Guillermo/_BD actualizada/Datos atom/PLACSP")

# configuración del script
PATH <- "licitacionesPerfilesContratanteCompleto3_2025"
RData_FILE <- "4.33_AwardingCriteria_2025.Rdata"
##DB <- "C:/Users/Joaquin/Desktop/Guille/Atom.accdb"
DBtablename <-  "AwardingCriteria"

#'Busca el padre con el tag inicado
#' @param xml_a punto de partida
#' @param tag etiqueta del padre que busca
#'
DamePadre <- function(xml_a,tag){
  xml_p <- xml_a
  repeat{
    xml_p <- xmlParent(xml_p)
    if(xmlName(xml_p) == tag) break
  }
  return(xml_p)
}


# Creamos un data.frame vacio para añadir los atributos.
nombres <- c("ContractID","entryID","WeightNumeric","Description")

Proy <- data.frame(matrix(NA,0,length(nombres)))
colnames(Proy) <- nombres

arch <-  list.files(path = PATH, pattern = "\\.atom$", full.names = F)

Proy <- foreach(ci = 1:length(arch), .combine=rbind, .packages = c("XML","dplyr"), .verbose = F) %dopar% {

    archivoTemp <- xmlParse(paste0(PATH,"/",arch[ci]))

    #  list of the children or sub-elements of an XML node whose tag name matches the one specified
    cat(paste(ci," Buscando 'ContractModification' en ",arch[ci],"\n"))
    buscar <-  xmlElementsByTagName(xmlRoot(archivoTemp),"AwardingCriteria", recursive = T)

    Proy <- data.frame(matrix(NA,0,length(nombres)))
    colnames(Proy) <- nombres

    for (xmla in buscar){
        foo <- Proy[1,]

      foo$WeightNumeric <- as.numeric(xmlValue(xmla[["WeightNumeric"]]))
      foo$Description <-xmlValue(xmla[["Description"]])

      # busca el padre para el ContractFolderStatus sube por la cadena entry.ContractFolderStatus.ContractModification
      xmlp <- DamePadre(xmla, "ContractFolderStatus")

      foo$ContractID <- xmlValue( xmlChildren(xmlp)$ContractFolderID )
```

```
68
69          bar <- strsplit(xmlValue( xmlChildren(xmlParent(xmlp))$id ), "/")[[1]] #Split
            la URL por /
70          foo$entryID <- bar[length(bar)]                                    # que
            queda con el utlimo valor
71
72
73          #cat( paste(foo$entryID,", "))
74
75          # acummulamos elementos encontrados
76          Proy <- rbind(Proy,as.data.frame(foo))
77
78        }
79        # cat(paste(" -> Encontrados ",length(buscar),"\n")) #traza
80        Proy
81    }
82
83    #stop cluster
84    stopCluster(cl)
85
86
87    save(Proy, nombres, file = RData_FILE)
88
89    #library(RODBC)
90    #mdb <- odbcDriverConnect(paste0("Driver={Microsoft Access Driver (*.mdb,
      *.accdb)};DBQ=",DB))
91    # mdb <- odbcDriverConnect(paste0("Driver={MariaDB ODBC 3.1
      Driver};user=root;password=123atom123;dbname=atomdb"))
92    #sqlFetch(mdb, "mtcars")
93    #sqlSave(mdb, Proy, tablename = DBtablename, append = T, rownames = F)
94    #odbcClose(mdb)
95
96
97
98    #library(DBI)
99    #con <- dbConnect(RMariaDB::MariaDB(), user="root", dbname="atomdb")
100   #dbWriteTable(con,"AwardingCriteria", Proy,  row.names = F, append=T)
101   #dbDisconnect(con)
102   # dbListTables(con)
103   # dbWriteTable(con, "mtcars", mtcars)
104   # dbReadTable(con, "mtcars")
105
106
```