```r
### SCRIPT 2 - R ###
# Obtencion de modificaciones contratctuales desde ATOM
#
#################################################################
library(XML)
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")

args = commandArgs(trailingOnly=TRUE)
if (length(args)==0){
  # configuración del script sin argumentos
  PATH <- "licitacionesPerfilesContratanteCompleto3_2025
  "
  RData_FILE <- "4.38_ContractModification_2025.Rdata"

} else if (length(args)==2){
  # configuración del script con argumentos
  PATH <- args[1]
  RData_FILE <- args[2]
} else stop("2 argumentos PATH RData_FILE", call. = FALSE)

#DB <- "C:/Users/Joaquin/Desktop/Guille/Atom.accdb"
#DBtablename <-  "ContractModification"

#'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("ID","ContractID","entryID", "updated",
              "ContractModificationDurationMeasure", "ContractModificationDurationMeasure_Uc",
              "FinalDurationMeasure","FinalDurationMeasure_Uc",
              "ContractModificationLegalMonetaryTotal", "ContractModificationLegalMonetaryTotal_c",
              "FinalLegalMonetaryTotal","FinalLegalMonetaryTotal_c")

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"), .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),"ContractModification",
      recursive = T)

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

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

      foo$ID <- as.numeric(xmlValue(xmla[["ID"]]))
      foo$ContractID <- xmlValue(xmla[["ContractID"]])

      foo$ContractModificationDurationMeasure <- as.numeric(xmlValue(xmla[[
      "ContractModificationDurationMeasure"]]))
      if (!is.na(foo$ContractModificationDurationMeasure))
          foo$ContractModificationDurationMeasure_Uc <- xmlGetAttr(xmla[[
          "ContractModificationDurationMeasure"]],"unitCode")
      foo$FinalDurationMeasure <- as.numeric(xmlValue(xmla[["FinalDurationMeasure"]]))
      if (!is.na(foo$FinalDurationMeasure))
          foo$FinalDurationMeasure_Uc <- xmlGetAttr(xmla[["FinalDurationMeasure"]],
          "unitCode")

      foo$ContractModificationLegalMonetaryTotal <- as.numeric(xmlValue(xmla[[
      "ContractModificationLegalMonetaryTotal"]][["TaxExclusiveAmount"]]))
      if (!is.na(foo$ContractModificationLegalMonetaryTotal))
          foo$ContractModificationLegalMonetaryTotal_c <- xmlGetAttr(xmla[[
          "ContractModificationLegalMonetaryTotal"]][["TaxExclusiveAmount"]],
          "currencyID")
      foo$FinalLegalMonetaryTotal <- as.numeric(xmlValue(xmla[[
      "FinalLegalMonetaryTotal"]][["TaxExclusiveAmount"]]))
      if (!is.na(foo$FinalLegalMonetaryTotal))
          foo$FinalLegalMonetaryTotal_c <- xmlGetAttr(xmla[["FinalLegalMonetaryTotal"
          ]][["TaxExclusiveAmount"]],"currencyID")

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

      bar <- strsplit(xmlValue( xmlChildren(xmlp)$id ), "/")[[1]] #Split la URL por /
      foo$entryID <- bar[length(bar)]                                    # que
      queda con el utlimo valor

      foo$updated <- as.POSIXct(xmlValue( xmlChildren(xmlp)$updated ), format=
      "%Y-%m-%dT%H:%M:%OS")

      #cat( paste(foo$entryID," [",foo$ContractID,"], "))

      # acummulamos elementos encontrados
      Proy <- rbind(Proy,as.data.frame(foo))
    }
    # cat(paste(" -> Encontrados ",length(buscar),"\n")) #traza
    Proy
}

#stop cluster
stopCluster(cl)


save(Proy,nombres, file = RData_FILE)


# load(file =  RData_FILE)
#library(DBI)
#con <- dbConnect(RMariaDB::MariaDB(), user="root", dbname="atomdb")
#dbWriteTable(con,DBtablename, Proy,  row.names = F, append=T)
#dbDisconnect(con)
```