

# Guía breve de preparación de datos para el Sistema de Disponibilidad Hídrica Superficial

*Eduardo Jesus Noriega Cabrera*

*Febrero de 2019*

## 1. Antecedentes

El geovisor montado por MMAyA (disponible en <https://sys.sirh.gob.bo/>) agrupa los resultados de los balance hídricos de varias cuencas del territorio boliviano, presentándolos por Unidad Hidrográfica de Reporte y Calibración (modelación y calibración respectivamente) en un formato amigable para el usuario que pretenda obtener información básica de un área de interés.

La información requerida para alimentar al geovisor se encuentra dentro de modelos hidrológicos realizados en **WEAP**<sup>1</sup> de cada cuenca estudiada (ya sea modelada como macrocuenca o a menor escala).

## 2. Alcance de esta guía

Esta guía presenta y describe herramientas para extraer los resultados de las modelaciones hidrológicas que servirán como información de entrada para el geovisor.

Las herramientas comprenden funciones escritas en **R** que permiten crear scripts en formato *.vbs* (Visual Basic Script, formato de scripts usado dentro de WEAP) y otras funciones que transforman, corrigen y dejan lista la información de las variables hidrológicas para ser usadas por el geovisor. Los archivos espaciales (delimitación de cuencas) también son creados con funciones, según los archivos de reporte y calibración de cada variable.

## 3. Herramientas

Las herramientas creadas se dividen en dos tipos:

1. Las que permiten la extracción de información desde WEAP; y
2. las que generan los archivos para el visor modificando y corrigiendo las salidas de WEAP.

Los herramientas producidas (scripts de **R**) están disponibles en el repositorio [https://github.com/noriegaedu/visor\\_git/tree/master/scripts\\_visor](https://github.com/noriegaedu/visor_git/tree/master/scripts_visor).

### 3.1. Extracción de información desde WEAP

Dentro de estas herramientas se diferencian las variables por unidad de reporte. Las variables *PCP*, *ETR*, *ETP*, *BF*, *IN* y *SR* serán presentadas en unidades de lámina de precipitación [*mm*] por lo que para su extracción desde WEAP es necesario dividir los resultados de cada unidad hidrológica entre su área correspondiente. Las variables de Caudal (observado y simulado) serán presentadas en unidades propias de la variable [*m<sup>3</sup>/s*] y su extracción desde WEAP se la hace directamente en estas unidades.

---

<sup>1</sup>WEAP: Water Evaluating And Planning System, Stockholm Environment Institute.

Se creó la función `weap_vb_extraer_todo()`:

```
# @author Eduardo Noriega
#
# @description Modificado de 'weap_vb_extraer_varios.R'
# Con esta nueva funcion se permite generar un solo script vbs para varias variables
#
# @param directorio carpeta donde se almacenara la info de salida
# Ej: 'bh'
# @param Var vector con variables
# *OJO* Por ahora en este orden: c('ETP', 'ETR', 'PCP', 'SR', 'IN', 'BF')
# Ej: 'ETP', 'ETR', 'PCP', 'SR', 'IN', 'BF'
# @param estaciones ruta del archivo Excel exportado desde WEAP con la
# informacion de las unidades hidrologicas: Para generar este archivos seguir estos
# pasos:
# - En WEAP ir al apartado DATA, del arbol de informacion escoger 'Demand and Catchments',
# - En el panel de la derecha escoger 'Land Use/Area'
# - Abrir de la barra de menu 'Edit/Export Expression to Excel...'
#   en la ventana emergente escoger
#   - New Workbook
#   - All Branches
#   - Area Variables
#   - Reference Scenario
# - Guardar la salida de Excel y recordar la ruta.
# Ej. "./Balance Hidrico Ravelo FINAL.xlsx"
# @param nom_salida ruta con el nombre del archivo de salida con extension '.vbs'
#
# @example
# weap_vb_extraer_todo(directorio = 'bh',
#                       Var = c('ETP', 'ETR', 'PCP', 'SR', 'IN', 'BF'),
#                       estaciones = "./Balance Hidrico Ravelo FINAL.xlsx",
#                       nom_salida = "./ravelo_FINAL.vbs")

weap_vb_extraer_todo <- function(directorio, estaciones, nom_salida,
                                Var = c('ETP', 'ETR', 'PCP', 'SR', 'IN', 'BF')){
  require(dplyr)
  require(readxl)

  directorio <- paste0("", directorio, "\\")

  var_A <- paste0('A', '_')

  Var <- paste0(Var, '_')

  val_name_A <- ':Area Calculated[M^2]' # para llevar a mm

  val_name <- c(':ET Potential[m^3]', ':ET Actual[m^3]',
               ':Observed Precipitation[m^3]', ':Surface Runoff[m^3]',
               ':Interflow[m^3]', ':Base Flow[m^3]')

  est <- read_xlsx(estaciones) %>%
    pull(5) %>% unique() %>% na.omit()
```

```

est <- est[-1] # se elimina 'Level 2'

est_variables <- est %>%
  gsub(' ', '_', .)

aa <- paste0(var_A, est_variables)

var_q <- paste0(' ', Var)
est_q <- paste0(est, ' ')

base <- paste("Scenario", "Year", "TS", sep = ' & ", " & ') %>%
  paste0(., ' & ", " & ')

zz <- sapply(seq_along(Var),
  function(y)
    paste(sapply(seq_along(est),
      function(x) paste0(var_q[y], est_q[x],
        collapse = ' & ", " & '))))

# inicio de escritura de vbs
con <- file(nom_salida)

writeLines(
  c('WEAP.ActiveScenario = "Reference"',
    '',
    sapply(seq_along(Var),
      function(x) paste0('salida',
        x,
        ' = WEAP.ActiveArea.Directory & ',
        directorio,
        ' & "',
        paste0(gsub('_', '', Var[x]),
          '.csv')))),
    '',
    'Set objFSO = CreateObject("Scripting.FileSystemObject")',
    '',
    sapply(seq_along(Var) ,
      function(x) paste0('if objFSO.FileExists(salida', x, ') then',
        '\n set objFile', x,
        '= objFSO.OpenTextFile(salida',x,', 8)',
        '\nElse',
        '\n set objFile', x,
        '= objFSO.CreateTextFile(salida',x,')',
        paste0('\n z',x,' = ', base, ' ',
          paste(zz[,x],
            collapse = ' & ", " & ')),
        '',
        paste0('\nobjFile',x, '.WriteLine z',x),
        '\nEnd If',
        '\n')
    ),
    '',
    'For Yr = (BaseYear+1) to EndYear',
    ' For Mes = 1 to NumTimeSteps',

```

```

'',
sapply(seq_along(est),
       function(x)
         paste0(paste0(var_A, est_variables)[x],
                  ' = WEAP.ResultValue("Demand Sites and Catchments\\',
                  est[x],
                  val_name_A,'" , Yr, Mes, WEAP.ActiveScenario)')),
'',
sapply(seq_along(est),
       function(x) paste0(zz[x,] %>%
                           gsub('\"', '\"',.) %>%
                           gsub(' \"', '\"',.),
                           ' = WEAP.ResultValue("Demand Sites and Catchments\\',
                           est[x],
                           val_name,'" , Yr, Mes, WEAP.ActiveScenario)')),
'',
sapply(seq_along(Var),
       function(y) paste0('z', y, ' = ',
                           paste(c('WEAP.ActiveScenario', 'Yr', 'Mes'),
                                sapply(seq_along(est),
                                       function(x)
                                         paste0('round(',
                                                  zz[x,y] %>%
                                                  gsub('\"', '\"',.) %>%
                                                  gsub(' \"', '\"',.), '/',
                                                  aa[x,']',
                                                  '*1000,2)'))),
                           collapse = ' & ", " & '))),
'',
sapply(seq_along(Var),
       function(x) paste0('objFile',x,'.WriteLine z',x)),
'',
' Next',
'Next',
'',
sapply(seq_along(Var),
       function(x) paste0('objFile',x,'.close'))),
con)
close(con)
}

```

Level 1	Level 2	Level 3	Variable	Scenario	Unit	Expression
Demand Sites and Catchments	UH Ravelo		Area	Current Accounts	km <sup>2</sup>	276.260397
Demand Sites and Catchments	UH Ravelo	R6_R25003000	Area	Current Accounts	% share of km <sup>2</sup>	1.436120792
Demand Sites and Catchments	UH Ravelo	R4_R30003500	Area	Current Accounts	% share of km <sup>2</sup>	23.93264533
Demand Sites and Catchments	UH Ravelo	R2_R35004000	Area	Current Accounts	% share of km <sup>2</sup>	29.00456376
Demand Sites and Catchments	UH Ravelo	R1_40004500	Area	Current Accounts	% share of km <sup>2</sup>	1.224346318
Demand Sites and Catchments	UH Ravelo	R3_R35004000cultivos	Area	Current Accounts	% share of km <sup>2</sup>	4.389483665
Demand Sites and Catchments	UH Ravelo	R5_R30003500cultivos	Area	Current Accounts	% share of km <sup>2</sup>	40.01284013
Demand Sites and Catchments	UH Corral Mayu		Area	Current Accounts	km <sup>2</sup>	173.347906
Demand Sites and Catchments	UH Corral Mayu	R4_R35004000cultivos	Area	Current Accounts	% share of km <sup>2</sup>	22.98764543
Demand Sites and Catchments	UH Corral Mayu	R3_R35004000	Area	Current Accounts	% share of km <sup>2</sup>	48.16076578
Demand Sites and Catchments	UH Corral Mayu	R2_R40004500	Area	Current Accounts	% share of km <sup>2</sup>	4.313
Demand Sites and Catchments	UH Corral Mayu	R1_R40004500cultivos	Area	Current Accounts	% share of km <sup>2</sup>	0.14
Demand Sites and Catchments	UH Corral Mayu	R6_R30003500	Area	Current Accounts	% share of km <sup>2</sup>	11.35991224
Demand Sites and Catchments	UH Corral Mayu	R5_R30003500cultivos	Area	Current Accounts	% share of km <sup>2</sup>	13.03848055
Demand Sites and Catchments	UH Maragua		Area	Current Accounts	km <sup>2</sup>	240.616523
Demand Sites and Catchments	UH Maragua	R3_R35004000	Area	Current Accounts	% share of km <sup>2</sup>	49.7992268
Demand Sites and Catchments	UH Maragua	R2_R40004500	Area	Current Accounts	% share of km <sup>2</sup>	43.204
Demand Sites and Catchments	UH Maragua	R4_R30003500	Area	Current Accounts	% share of km <sup>2</sup>	6.687495854
Demand Sites and Catchments	UH Maragua	R1_R45005000	Area	Current Accounts	% share of km <sup>2</sup>	0.308984184
Demand Sites and Catchments	UH Percas Mayu		Area	Current Accounts	km <sup>2</sup>	157.870204
Demand Sites and Catchments	UH Percas Mayu	R5_R30003500	Area	Current Accounts	% share of km <sup>2</sup>	7.454126049
Demand Sites and Catchments	UH Percas Mayu	R4_R30003500cultivos	Area	Current Accounts	% share of km <sup>2</sup>	7.335
Demand Sites and Catchments	UH Percas Mayu	R2_R35004000cultivos	Area	Current Accounts	% share of km <sup>2</sup>	47.35248458
Demand Sites and Catchments	UH Percas Mayu	R3_R35004000	Area	Current Accounts	% share of km <sup>2</sup>	26.36642694
Demand Sites and Catchments	UH Percas Mayu	R1_R40004500	Area	Current Accounts	% share of km <sup>2</sup>	11.49233835

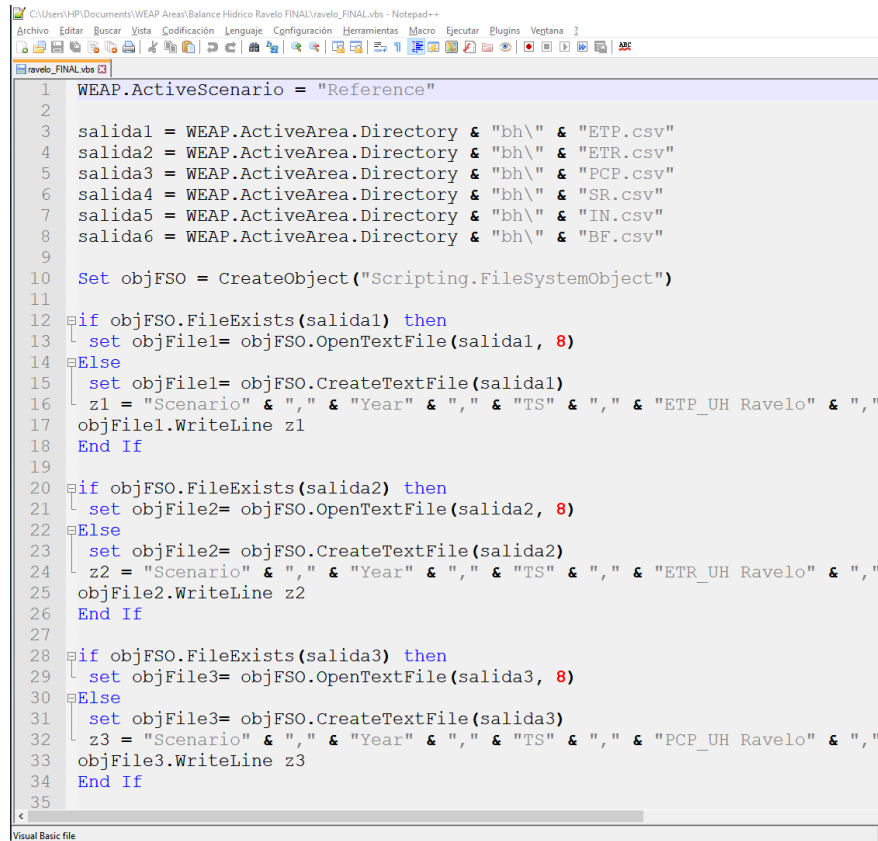
Figura 1: Archivo.xlsx con los nombres de las unidades hidrológicas.

La función crea un archivo *.vbs* que, dentro de WEAP, permite la extracción de la información de las variables *ETP*, *ETR*, *PCP*, *SR*, *IN* y *BF*. Cada argumento tiene una descripción pero a continuación se enumeran dos pasos previos antes de usar la función:

1. Un archivo *.xlsx* extraído desde WEAP con la información de los nombres de cada *Catchment* (unidad hidrológica) presente dentro del modelo. Más detalles de este archivo se encuentra en la descripción de la función (buscar descripción del argumento **estaciones**). Ver la Figura 1.
2. Dentro de la carpeta de trabajo del modelo WEAP crear la carpeta *bh*. En esta carpeta se almacenarán los archivos de salida de WEAP.

Una vez cargada la función, el siguiente código de ejemplo:

```
# Ravelo
weap_vb_extraer_todo('bh',
    './Balance Hidrico Ravelo FINAL.xlsx',
    './ravelo_FINAL.vbs',
    Var = c('ETP', 'ETR', 'PCP', 'SR', 'IN', 'BF'))
```



```
1 WEAP.ActiveScenario = "Reference"
2
3 salida1 = WEAP.ActiveArea.Directory & "bh\" & "ETP.csv"
4 salida2 = WEAP.ActiveArea.Directory & "bh\" & "ETR.csv"
5 salida3 = WEAP.ActiveArea.Directory & "bh\" & "PCP.csv"
6 salida4 = WEAP.ActiveArea.Directory & "bh\" & "SR.csv"
7 salida5 = WEAP.ActiveArea.Directory & "bh\" & "IN.csv"
8 salida6 = WEAP.ActiveArea.Directory & "bh\" & "BF.csv"
9
10 Set objFSO = CreateObject("Scripting.FileSystemObject")
11
12 If objFSO.FileExists(salida1) Then
13     Set objFile1 = objFSO.OpenTextFile(salida1, 8)
14 Else
15     Set objFile1 = objFSO.CreateTextFile(salida1)
16     z1 = "Scenario" & ", " & "Year" & ", " & "TS" & ", " & "ETP_UH Ravelo" & ", "
17     objFile1.WriteLine z1
18 End If
19
20 If objFSO.FileExists(salida2) Then
21     Set objFile2 = objFSO.OpenTextFile(salida2, 8)
22 Else
23     Set objFile2 = objFSO.CreateTextFile(salida2)
24     z2 = "Scenario" & ", " & "Year" & ", " & "TS" & ", " & "ETR_UH Ravelo" & ", "
25     objFile2.WriteLine z2
26 End If
27
28 If objFSO.FileExists(salida3) Then
29     Set objFile3 = objFSO.OpenTextFile(salida3, 8)
30 Else
31     Set objFile3 = objFSO.CreateTextFile(salida3)
32     z3 = "Scenario" & ", " & "Year" & ", " & "TS" & ", " & "PCP_UH Ravelo" & ", "
33     objFile3.WriteLine z3
34 End If
35
```

Figura 2: Salida vbs.

genera el archivo *.vbs* que se muestra en la Figura 2.

Desde WEAP, el archivo *.vbs* generado produce los archivos *ETP.csv*, *ETR.csv*, *PCP.csv*, *SR.csv*, *IN.csv* y *BF.csv* dentro de la carpeta *./bh*. Estos archivos muestran los resultados de cada unidad hidrológica en [mm] según el periodo en el que fue ejecutado el modelo (Ver Figura 3).

PCP.csv - Excel (Error de activación de productos)							
ARCHIVO INICIO INSERTAR DISEÑO DE PÁGINA FÓRMULAS DATOS REVISAR VISTA COMPLEMENTOS ACROBAT							
A1	Scenari						
	A	B	C	D	E	F	G
1	Scenario	Year	TS	PCP_UH Ravelo	PCP_UH Corral Mayu	PCP_UH Maragua	PCP_UH Percas Mayu
2	Reference	2002	1	41.88	1.92	1.19	1.82
3	Reference	2002	2	17.89	49	36.23	49.62
4	Reference	2002	3	56.02	30.9	24.72	30.9
5	Reference	2002	4	36.53	80.4	78.62	80.4
6	Reference	2002	5	59.07	61.4	60.47	61.4
7	Reference	2002	6	134.42	175.4	175.12	175.4
8	Reference	2002	7	61.71	103.9	96.77	103.9
9	Reference	2002	8	96.47	43.11	36.3	46.3
10	Reference	2002	9	0.34	11.2	11.2	11.2
11	Reference	2002	10	0.03	0.6	0.6	0.6
12	Reference	2002	11	3.3	1.8	6.6	6.6
13	Reference	2002	12	0.13	0.16	0.2	0.2
14	Reference	2003	1	0.65	1.25	0.78	1.18
15	Reference	2003	2	147.52	82.18	60.76	83.23
16	Reference	2003	3	76.38	74.9	59.91	74.9
17	Reference	2003	4	96.02	53.5	52.32	53.5
18	Reference	2003	5	138.72	95.6	94.16	95.6
19	Reference	2003	6	75.9	85	84.86	85
20	Reference	2003	7	8.87	97.1	90.44	97.1
21	Reference	2003	8	0.64	1.86	1.57	2
22	Reference	2003	9	1.96	35.2	35.2	35.2
23	Reference	2003	10	0	0	0	0
24	Reference	2003	11	0.22	0.33	1.2	1.2
25	Reference	2003	12	0.08	2.91	3.6	3.6
26	Reference	2004	1	14.79	27.77	17.24	26.3
27	Reference	2004	2	1.05	16	11.83	16.2

Figura 3: Resultados extraídos desde WEAP con archivos vbs.

### 3.1.2. Caudal simulado y observado

Se creó la función `weap_vb_extraer_Q()`:

```
#' @author Edauro Noriega
#' @description La funcion crea un archivo vbs (script de visual basic) para extraer
#' informacion de caudal observado o simulado en WEAP
#'
#' @param directorio caracter, carpeta donde se guardaran la salida
#' @param nom_salida ruta con el nombre para el archivo vbs
#' @param ctrl_csv con la siguiente estructura (sin nombre de columnas):
#' |a|b|c|d|
#' |---|---|---|---|
#' |---|---|---|---|
#' a = nombre estacion hidrometrica (igual a nombre usado en WEAP)
#' Si no existe hidrometrica anhadir NG_ al inicio del nombre
#' b = nombre rio donde esta la estacion hidrometrica
#' c = Nodo (de WEAP) de donde se extrae informacion de caudal
#' d = Existe estacion hidrometrica para nodo del que se extrae
#' informacion? Binario (1 o 0)
#' @param sim_obs caracter para indicar que tipo de caudal se quiere extraer:
#' simulado ('sim') u observado ('obs').
#'
#' @example
#' datos_resumen :
```

```

# ' | Hidro Nujchu | Rio Ravelo Bajo | Sist.Riego RaveloBajo Return | 1 |
# ' | NG_Canchi Mayu | Rio Canchi Mayu | Sist.Riego Canchimayu Return | 0 |
# '
# ' Con este ejemplo se genera el vbs para caudales simulados:
# ' weap_vb_extraer_Q('bh',
# '                      'C:/Users/HP/Documents/WEAP Areas/prueba_q_ojo.vbs',
# '                      "H:/mmayu/datos_q_ravelo.csv",
# '                      sim_obs = 'sim')

weap_vb_extraer_Q <- function(directorio, nom_salida, ctrl, sim_obs = 'sim'){

  base <- paste("Scenario", "Year", "TS", sep = ' & "," & ') %>%
    paste0(., ' & "," & ')

  directorio <- paste0(' ', directorio, '\\')

  ctrl <- read.csv(ctrl, header = FALSE, stringsAsFactors = FALSE)
  ctrl <- list(pto_ctrl = ctrl$V1,
              rio = ctrl$V2,
              nodo_ctrl = ctrl$V3,
              obs = ctrl$V4)

  head_q <- paste(base,
                  paste0(paste0(' ',
                                ctrl$pto_ctrl %>%
                                  gsub(' ', '_', .), ' '),
                          collapse = ' & "," & '))

  if (sim_obs == 'sim') {
    nombre_csv <- "Q_sim.csv"
    extraccion <- sapply(seq_along(ctrl$pto_ctrl),
                          function(x)
                            paste0(ctrl$pto_ctrl[x] %>% gsub(' ', '_', .),
                                    ' = WEAP.ResultValue("Supply and Resources',
                                    '\\River\\',
                                    ctrl$rio[x],
                                    '\\Reaches\\Below ',
                                    ctrl$nodo_ctrl[x],
                                    ':Streamflow[CMS]", Yr, Mes,',
                                    ' WEAP.ActiveScenario'))
  }

  if (sim_obs == 'obs') {
    nombre_csv <- "Q_obs.csv"
    extraccion <- sapply(seq_along(ctrl$pto_ctrl),
                          function(x)
                            ifelse(ctrl$obs[x] == 1,
                                    paste0(ctrl$pto_ctrl[x] %>%
                                              gsub(' ', '_', .),
                                              ' = WEAP.ResultValue("Supply and Resources',
                                              '\\River\\',
                                              ctrl$rio[x],
                                              '\\Streamflow Gauges\\',

```



```

        ctrl$pto_ctrl[x],
        'Streamflow Data[CMS]", Yr, Mes,',
        ' WEAP.ActiveScenario)'),
paste0(ctrl$pto_ctrl[x] %>% gsub(' ', '_', .),
        ' = -999'))
}

con <- file(nom_salida)

writeLines(
  c('WEAP.ActiveScenario = "Reference"',
    '',
    paste0('salida1 = WEAP.ActiveArea.Directory & ', directorio,
           ' & ', nombre_csv),
    '',
    'Set objFSO = CreateObject("Scripting.FileSystemObject")',
    '',
    'if objFSO.FileExists(salida1) then',
    ' set objFile1 = objFSO.OpenTextFile(salida1, 8)',
    'Else',
    ' set objFile1 = objFSO.CreateTextFile(salida1)',
    paste0(' z1 = ', head_q),
    '',
    'objFile1.WriteLine z1',
    'End If',
    '',
    'For Yr = (BaseYear+1) to EndYear',
    ' For Mes = 1 to NumTimeSteps',
    '',
    extraccion,
    '',
    paste0('\nz1 = ', paste(c(c('WEAP.ActiveScenario', 'Yr', 'Mes'),
                              sapply(seq_along(ctrl$pto_ctrl),
                                     function(x) paste0('round(',
                                                         ctrl$pto_ctrl[x] %>%
                                                         gsub(' ', '_', .),
                                                         ',2)'))),
                              collapse = ' & ", " & '))),
    '',
    'objFile1.WriteLine z1',
    '',
    ' Next',
    'Next',
    '',
    'objFile1.close'),
  con)
close(con)
}

```

La función permite crear un archivo *.vbs* que, dentro de WEAP, permite la extracción de los resultados de las variables Caudal Simulado y Caudal Observado. Cada argumento tiene su descripción pero a continuación se enumeran dos pasos previos antes de usar la función:

1. Se necesita crear manualmente un archivo *.csv* según la descripción del argumento **ctrl**. Figura 4.

2. Dentro de la carpeta del modelo WEAP de trabajo crear la carpeta *bh*. En esta carpeta se almacenarán los archivos de salida de WEAP.

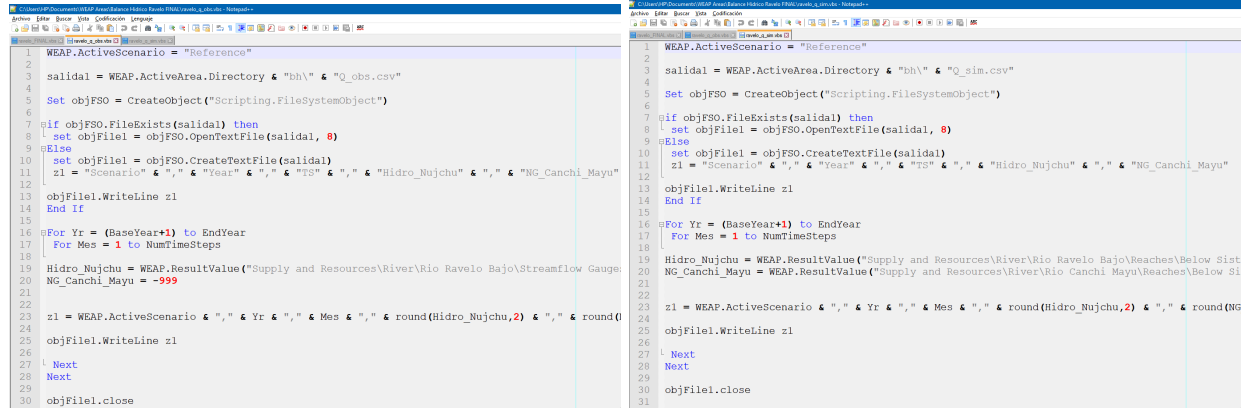


	A	B	C	D	E
1	Hidro Nujchu	Rio Ravelo Bajo	Sist.Riego RaveloBajo Return	1	
2	NG_Canchi Mayu	Rio Canchi Mayu	Sist.Riego Canchimayu Return	0	
3					

Figura 4: Archivo csv con información hidrómetrica.

Una vez cargada la función, el siguiente bucle de ejemplo:

```
# Ravelo
simobs <- c('sim', 'obs')
for (i in simobs) {
  weap_vb_extraer_Q('bh',
    paste0('./ravelo_q_', i, '.vbs'),
    "./datos_q_ravelo.csv",
    sim_obs = i)
}
```



```
1 WEAP.ActiveScenario = "Reference"
2
3 salidal = WEAP.ActiveArea.Directory & "bh\" & "Q_obs.csv"
4
5 Set objFSO = CreateObject("Scripting.FileSystemObject")
6
7 If objFSO.FileExists(salidal) Then
8   Set objFile = objFSO.OpenTextFile(salidal, 8)
9 Else
10  Set objFile = objFSO.CreateTextFile(salidal)
11  z1 = "Scenario" & "," & "Year" & "," & "TS" & "," & "Hidro_Nujchu" & "," & "NG_Canchi_Mayu"
12  objFile.WriteLine z1
13 End If
14
15 For Yr = (BaseYear+1) to EndYear
16   For Mes = 1 to NumTimeSteps
17     Hidro_Nujchu = WEAP.ResultValue("Supply and Resources\River\Rio Ravelo Bajo\Streamflow Gauge")
18     NG_Canchi_Mayu = -999
19     z1 = WEAP.ActiveScenario & "," & Yr & "," & Mes & "," & round(Hidro_Nujchu,2) & "," & round(NG
20     objFile.WriteLine z1
21   Next
22 Next
23 objFile.close
```

```
1 WEAP.ActiveScenario = "Reference"
2
3 salidal = WEAP.ActiveArea.Directory & "bh\" & "Q_sim.csv"
4
5 Set objFSO = CreateObject("Scripting.FileSystemObject")
6
7 If objFSO.FileExists(salidal) Then
8   Set objFile = objFSO.OpenTextFile(salidal, 8)
9 Else
10  Set objFile = objFSO.CreateTextFile(salidal)
11  z1 = "Scenario" & "," & "Year" & "," & "TS" & "," & "Hidro_Nujchu" & "," & "NG_Canchi_Mayu"
12  objFile.WriteLine z1
13 End If
14
15 For Yr = (BaseYear+1) to EndYear
16   For Mes = 1 to NumTimeSteps
17     Hidro_Nujchu = WEAP.ResultValue("Supply and Resources\River\Rio Ravelo Bajo\Reaches\Below Sist
18     NG_Canchi_Mayu = WEAP.ResultValue("Supply and Resources\River\Rio Canchi Mayu\Reaches\Below Si
19     z1 = WEAP.ActiveScenario & "," & Yr & "," & Mes & "," & round(Hidro_Nujchu,2) & "," & round(NG
20     objFile.WriteLine z1
21   Next
22 Next
23 objFile.close
```

Figura 5: Salida vbs luego de usar la funcion: (Izq.) Caudal Observado y (Der.) Caudal Simulado.

genera los archivos *.vbs* que se muestran en la Figura 5.

Dentro de WEAP, los archivos *.vbs* generados producen los archivos *Q\_sim.csv* y *Q\_obs.csv*, los cuales muestran los resultados de cada punto de control (donde sea posible) en  $[m^3/s]$  según el periodo en el que fue ejecutado el modelo. (Ver Figura 6).

	A	B	C	D	E
1	Scenario	Year	TS	Hidro_Nujchu	NG_Canchi_Mayu
2	Reference	1971	1	0.1	0.15
3	Reference	1971	2	0.07	0.12
4	Reference	1971	3	0.03	0.06
5	Reference	1971	4	0.02	0.04
6	Reference	1971	5	0.02	0.03
7	Reference	1971	6	0.01	0.03
8	Reference	1971	7	0.01	0.03
9	Reference	1971	8	0.03	0.05
10	Reference	1971	9	0.03	0.04
11	Reference	1971	10	0.02	0.04
12	Reference	1971	11	0.02	0.03
13	Reference	1971	12	0.02	0.03
14	Reference	1972	1	0.02	0.03
15	Reference	1972	2	0.02	0.02
16	Reference	1972	3	0.01	0.01
17	Reference	1972	4	0.01	0.01
18	Reference	1972	5	0.01	0.01
19	Reference	1972	6	0.01	0.01
20	Reference	1972	7	0.01	0.01

Figura 6: Resultados extraídos desde WEAP con archivos .vbs para el Caudal simulado.

## 3.2. Generación de archivos

En esta sección se agrupan las herramientas que permiten generar los archivos de reporte y calibración.

### 3.2.1. Archivos de reporte

Para generar los archivos de reporte se debe evaluar algunos aspectos de la información disponible del modelo:

- Si el modelo fue ejecutado con el mes de inicio diferente a **Enero**, las fechas de las salidas generadas a partir de los archivos .vbs (figuras 3 y 6) estarán desfasadas respecto a las fechas de reporte. Esto se produce porque el modelo WEAP no distingue entre meses y pasos temporales (por este motivo en las figuras mencionadas no existe la columna **Month** pero si **TS**).
- Se debe verificar la existencia de la delimitación de las cuencas dentro de los documentos del modelamiento de la cuenca de estudio (comúnmente en formato .shp y sus asociados). En caso de no existir se deberá generar uno con las mismas unidades hidrológicas de los archivos .csv generados a partir del modelo.

La generación de los archivos de reporte (resultados del modelo y archivos de delimitación) que serán introducidos al visor son producidos por la función `arch_visor()`.

```
#' @author Eduardo Noriega
#
#' @description Funcion que permite generar archivos csv para alimentar a visor
```

```

#'
#' @param shp_cuencas shape con la informacion espacial de la cuenca modelada
#' Usar el shapefile generado por WEAP
#' Ej.shp_cuencas <- "./Cuenca_Cachimayu_UH_weap.shp"
#' **OJO: Obsoleto**
#' Si se desea usar un shapefile similar al generado por WEAP tener cuidado con errores
#' al usar la funcion
#' Dentro del codigo de esta funcion se incluye un pedazo de codigo que permite salvar
#' errores de lectura de cuencas dentro de shapefile no generado por WEAP.
#' Sin embargo no fiarse y revisar manualmente.
#' Ej. shape no WEAP: "./Calibracion/SIG/CatchmentsNacional.shp"
#' **OJO: Obsoleto**
#' @param var_csv Ruta en la que se encuentran los archivos csv de las variables modeladas
#' Ej. "./bh/"
#' @param abr_var variable modelada de la cual obtener csv para visor
#' EJ. 'PCP'
#' @param salida Ruta de salidas de los archivos csv para visor
#' Ej. "./bh/"
#' @param salida_shp Ruta de salida para los shapefiles asociados a cada variable modelada
#' Ej. "./bh/shp/"
#' @param corregir_ah Logico, si modelo fue corrido con anho inicial diferente de Enero
#' se debe corregir las salidas generadas con vbs.
#' Ej. TRUE para corregir desfase de informacion de meses
#' @param escribir_shp Logico, se da la opcion de escribir los shapes de cada
#' variable (TRUE) o no (FALSE) si es que ya se tienen escritos.
#'
arch_visor <- function(shp_cuencas, var_csv, abr_var, salida, salida_shp,
                      corregir_ah = FALSE, escribir_shp = TRUE){

  require(tidyverse)
  require(sf)

  bhsb_shp <- st_read(shp_cuencas,
                     stringsAsFactors = FALSE)

  #verificar si: shp no tiene crs y si existen coordenadas negativas
  # (lo que indica que es proy geogr) entonces se asiga proy geogr a cuenca leida
  if (is.na(st_crs(bhsb_shp)) & all(st_bbox(bhsb_shp) < 0)) {
    bhsb_shp <- st_set_crs(bhsb_shp, 4326)
  }

  # nombres cuencas:
  nombres_cuencas <- read.csv(paste0(var_csv,abr_var,'.csv'),
                             check.names = FALSE) %>% names
  nombres_cuencas <- nombres_cuencas[-c(1:3)] %>% gsub(paste0(abr_var, '_'), '', .)

  # seleccionar cuencas del shape en base a nombres de cuencas con informacion
  bhsb_shp_reducido <- bhsb_shp %>% filter(Name %in% nombres_cuencas)
  faltantes <- setdiff(nombres_cuencas, bhsb_shp_reducido$Name)

  # **OJO: Obsoleto**
  # # codigo para verificar nombres que no concuerdan entre los nombres de
  # # cuencas y nombres dentro del shape

```

```

# # ver descripcion
# while (length(faltantes) != 0) {
#   # loop para posicion de faltantes en un solo vector
#   pos_faltantes <- sapply(seq_along(faltantes),
#                             function(x) return(agrep(setdiff(nombres_cuencas,
#                                                                bhsb_shp_reducido$Name)[x],
#                                                                bhsb_shp$Name)))) #
#   bhsb_shp[pos_faltantes,3] <- faltantes
#   bhsb_shp_reducido <- bhsb_shp %>% filter(Name %in% nombres_cuencas) # se repite
#                                     # para incluir
#   faltantes <- setdiff(nombres_cuencas, bhsb_shp_reducido$Name)
# }
***OJO: Obsoleto**

bhsb_shp_reducido <- bhsb_shp_reducido %>%
  arrange(Name) %>%
  transmute(NOMBRE = Name,
            PFAF_HYD_ = 999,
            OBSERV_ = NA,
            codigo = paste0('uh', 1:nrow(.)),
            Cuenca = Name)

crear_csvvisor <- read.csv(paste0(var_csv,abr_var,'.csv'),
                           check.names = FALSE) %>%
  dplyr::select(-c(1:3))

if (isFALSE(corregir_ah)) {
  fechas <- read.csv(paste0(var_csv,abr_var,'.csv')) %>%
    dplyr::select(2:3) %>%
    setNames(c('Year', 'Month'))
}

if (isTRUE(corregir_ah)) {
  fechas <- read.csv(paste0(var_csv,abr_var,'.csv')) %>%
    dplyr::select(2:3) %>%
    setNames(c('Year', 'Month'))

  anhos <- fechas$Year %>% unique()
  ini <- anhos %>% head(1)
  fin <- anhos %>% tail(1)

  fechas <- fechas %>%
    mutate(Year = c(rep(ini - 1,4),
                      rep(seq(ini, fin - 1), each = 12),
                      rep(fin,8)),
           Month = rep(c(9:12,1:8), length(anhos)))

  # corregir salida de WEAP
  read.csv(paste0(var_csv,abr_var,'.csv'),
           check.names = FALSE) %>%
    mutate(Year = fechas$Year,
           TS = fechas$Month) %>%

```

```

        write.csv(paste0(var_csv,abr_var,'.csv'),
                  row.names = FALSE)
    }

    write.csv(cbind(fechas,
                    crear_csv_visor %>%
                      dplyr::select(sort(names(.))) %>%
                      setNames(., c('Year', 'Month', bhsb_shp_reducido$codigo)),
                    paste0(salida, abr_var, '_visor.csv'),
                    row.names = FALSE)

    if (isTRUE(escribir_shp)) {
        st_write(bhsb_shp_reducido,
                 paste0(salida_shp, abr_var, '_visor.shp'))
    }
}

```

Con el siguiente bucle se generan los archivos de reporte y los archivos de delimitación en formato *.shp*. Cualquier archivo de delimitación generado puede ser usado como capa base dentro del geovisor.

```

# Ravelo
shp_cuencas <- "./Cuenca_Cachimayu_UH_weap.shp"
var_csv <- "./bh/"
salida <- "./bh/"
salida_shp <- "./bh/shp/"
corregir_ah <- TRUE
escribir_shp <- TRUE

variables <- c('BF', 'ETP', 'ETR', 'IN', 'PCP', 'SR')

for (i in variables) {
    arch_visor(shp_cuencas, var_csv, i, salida, salida_shp, corregir_ah, escribir_shp)
}

# para escurrimiento
library(tidyverse)
lis_esc <- list.files(salida, pattern = 'visor', full.names = TRUE) %>%
  grep('BF_visor|IN_visor|SR_visor', ., value = TRUE) %>%
  lapply(function(x) read.csv(x, stringsAsFactors = FALSE)) %>%
  Reduce('+', .)

lis_esc[,1:2] <- lis_esc[,1:2]/3L

write.csv(lis_esc, paste0(salida, 'esc_visor.csv'), row.names = FALSE)

```

En la primera parte del ejemplo anterior se generan los archivos *ETP\_visor.csv*, *ETR\_visor.csv*, *PCP\_visor.csv*, *SR\_visor.csv*, *IN\_visor.csv* y *BF\_visor.csv*. Ver figura 7. Además, a partir de *SR\_visor.csv*, *IN\_visor.csv* y *BF\_visor.csv* en la segunda parte del ejemplo se genera el archivo *esc\_visor.csv* que corresponde a la variable Escurrimiento en lámina de precipitación [mm].

Nótese que en la figura 7 los nombres de las unidades hidrológicas han sido modificados. Este renombramiento coincide con cada componente dentro del archivo de delimitación de cuenca. En la figura 8 se muestra la representación espacial del archivo de delimitación generado y en el cuadro 1 se muestran los datos del archivo de delimitación generado.

SR\_visor.csv - Excel (Error de activación de productos)

ARCHIVO INICIO INSERTAR DISEÑO DE PÁGINA FÓRMULAS DATOS REVISAR VISTA COMPLEMENTOS ACROBAT Eduardo...

A1 Year

	A	B	C	D	E	F	G	H	I	J	K
	Year	Month	uh1	uh2	uh3	uh4	uh5	uh6	uh7	uh8	uh9
1	Year	Month	uh1	uh2	uh3	uh4	uh5	uh6	uh7	uh8	uh9
2	2001	9	0	0	0	0.01	0.08	0	0	0.02	0
3	2001	10	0.13	1.07	1.54	3.89	0.03	0.26	0.44	7.2	0.0
4	2001	11	0.03	0.59	0.87	2.25	0.29	0.28	0.2	4.06	0.0
5	2001	12	0.85	3.12	6.03	10.18	0.15	1.61	1.57	21.12	0.3
6	2002	1	1.07	1.78	3.6	5.89	0.2	1.05	0.82	12.34	0.3
7	2002	2	1.04	13.12	22.94	36.11	2.05	4.45	7.73	72.26	0.8
8	2002	3	0.03	5.31	8.62	15.74	0.45	0.92	2.72	30.08	0.0
9	2002	4	0.2	0.99	1.52	3.94	1	0.37	0.37	5.7	0.0
10	2002	5	0.01	0.09	0.19	0.35	0	0.01	0.03	0.68	0
11	2002	6	0	0	0	0	0	0	0	0.01	0
12	2002	7	0	0	0.05	0.07	0	0	0.01	0.15	0
13	2002	8	0	0	0	0	0	0	0	0	0
14	2002	9	0	0	0	0	0	0	0	0	0
15	2002	10	0.09	2.92	4	9.97	1.13	0.21	1.37	18.11	0.0

SR\_visor

LISTO

Figura 7: Archivo csv de reporte

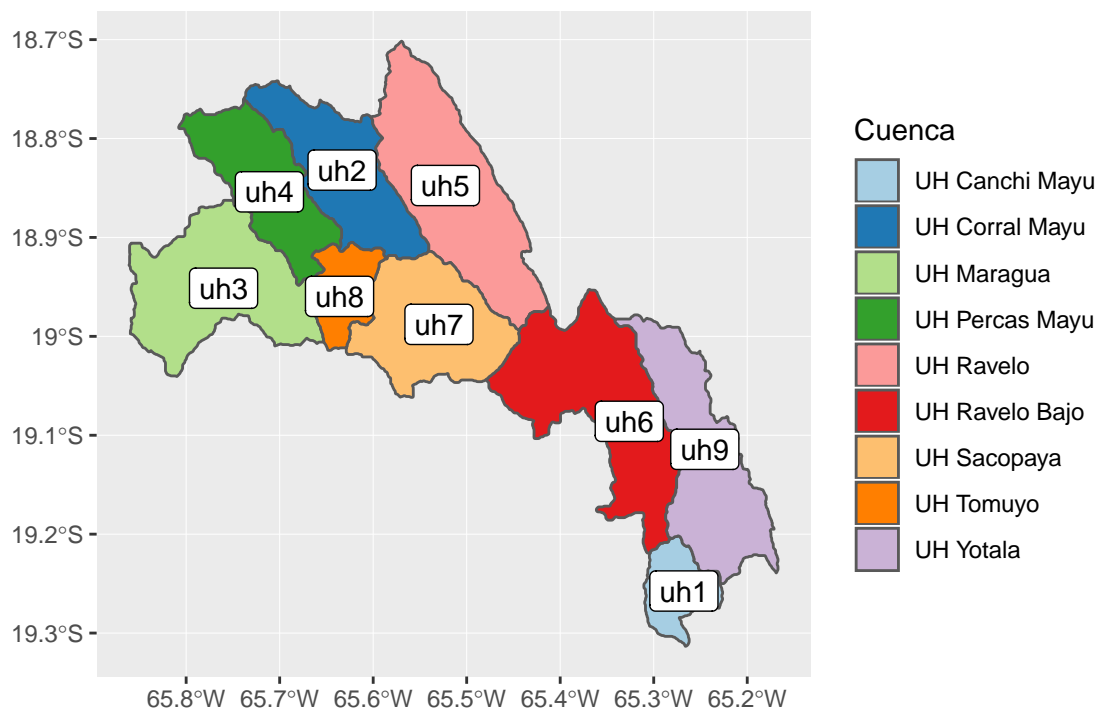


Figura 8: Representación espacial del archivo de delimitación generado

Cuadro 1: Tabla asociada al archivo de delimitación generado.

NOMBRE	PFAF_HYD__	OBSERV__	codigo	Cuenca
UH Canchi Mayu	999	NA	uh1	UH Canchi Mayu
UH Corral Mayu	999	NA	uh2	UH Corral Mayu
UH Maragua	999	NA	uh3	UH Maragua
UH Percas Mayu	999	NA	uh4	UH Percas Mayu
UH Ravelo	999	NA	uh5	UH Ravelo
UH Ravelo Bajo	999	NA	uh6	UH Ravelo Bajo
UH Sacopaya	999	NA	uh7	UH Sacopaya
UH Tomuyo	999	NA	uh8	UH Tomuyo
UH Yotala	999	NA	uh9	UH Yotala

### 3.2.2. Archivos de calibración

A diferencia de los archivos generados para el reporte, los archivos a generar para la calibración no cuentan con un archivo de delimitación de cuenca hasta los puntos donde existe una estación hidrométrica con información. Por este motivo se debe generar los archivos de delimitación a partir de los archivos de delimitación de reporte. Para este trabajo sera de utilidad el archivo *.csv* mostrado en la figura 4. Además, los archivos generados de Caudal (simulado y observado) pueden estar desfasados temporalmente si es que el modelo hidrológico fue ejecutado con el mes de inicio diferente a **Enero**.

Con la función `calibr()` se crean los archivos de delimitación de las cuencas de calibración y se preparan los archivos de calibración. Si es necesario corregir por desfase temporal los archivos de caudal, pasar el argumento `corregir_ah == TRUE`.

```
#' @author Eduardo Noriega
#'
#' @description La funcion permite crear un shapefile de calibracion en base a un
#' shapefile de reporte de un balance hidrico y genera los archivos para el geovisor
#' de las todas las variables (ETP, ETR, PCP, BF, IN, SR, Q_sim y Q_obs).
#' Si es necesario la funcion corrige el desfase temporal de los archivos de caudal.
#'
#' @param datos_resumen csv con la siguiente estructura (sin nombre de columnas)
#' |a-|b-|c-|d-|
#' |---|---|---|---|
#' |---|---|---|---|
#' a = nombre estacion hidrometrica (igual a nombre usado en WEAP)
#' Si no existe hidrometrica anhadir NG_ al inicio del nombre
#' b = nombre rio donde esta estacion hidrometrica
#' c = Nodo (de WEAP) de donde se extrae informacion de caudal
#' d = Existe estacion hidrometrica para nodo del que se extrae informacion? Binario
#' (1 o 0)
#' @param shp_reporte ruta del shapefile de reporte de la cuenca (el que fue usado
#' para visor)
#' @param shp_calibr ruta para escribir shapefile de calibracion
#' @param ruta_q ruta donde se encuentran los archivos de caudal de WEAP
#' @param corregir_ah logico, TRUE para corregir fechas por aho hidrológico
#'
#' @details dentro de la estructura de la funcion se presenta una funcion adicional
#' llamada 'uh_calibr'
#' esta funcion permite asociar la UH del reporte con las UH de la calibracion de forma
#' interactiva al ser llamada, la funcion itera a traves de la cantidad de datos (filas,
```



```

#' uh de calibracion) del archivo 'datos_resumen'. Para cada dato se pide introducir el
#' numero de UH del reporte que le corresponde separado por espacios.
#'
#' @example
#' datos_resumen :
#' | Hidro Nujchu | Rio Ravelo Bajo | Sist.Riego RaveloBajo Return | 1 |
#' | NG_Canchi Mayu | Rio Canchi Mayu | Sist.Riego Canchimayu Return | 0 |
#'
#' calibr("./calibr_input_prueba.csv",
#'         "./bh/shp/PCP_visor.shp",
#'         "./bh/shp/ravelo_calibr_solo2.shp",
#'         reporte = TRUE,
#'         ruta_entrada_reportes = "./bh/",
#'         ruta_salida_reportes = "./bh/",
#'         ruta_q = './bh/',
#'         corregir_ah = TRUE)

calibr <- function(datos_resumen, shp_reporte, shp_calibr, reportes = FALSE,
                   ruta_entrada_reportes, ruta_salida_reportes,
                   ruta_q, corregir_ah = FALSE){

  require(tidyverse)
  require(sf)

  nombres_features <- read.csv(datos_resumen,
                                header = FALSE,
                                stringsAsFactors = FALSE) %>%
    pull(1)

  shp_reporte <- st_read(shp_reporte, quiet = TRUE)

  uh_calibr <- function(){
    message('UH de Reporte de shp que corresponden a cada UH para shp de Calibracion:')
    uh <- lapply(seq_along(nombres_features),
                 function(x) readline(paste0('UH para "',
                                              nombres_features[x],
                                              '" = ')))) %>%
      lapply(function(x) as.numeric(unlist(strsplit(x, ' ')))) %>%
      lapply(function(x) paste0('uh', x)) %>%
      setNames(nombres_features)
    return(uh)
  }

  uh <- uh_calibr()

  lapply(seq_along(nombres_features),
         function(x) shp_reporte %>%
           filter(codigo %in% uh[[x]]) %>% # si basado en shp original usar Name
           st_union() %>%
           st_sf(Name = names(uh)[x], .)) %>%
    do.call(rbind, .) %>%
    transmute(NOMBRE = Name,
              PFAF_HYD_ = 999,

```

```

        OBSERV_ = NA,
        codigo = paste0('uhc', seq_along(nombres_features))) %>%
st_write(shp_calibr)

# otras variables difenrtes de Q

if (reportes == TRUE) {
  reportes_csv <- list.files(ruta_entrada_reportes,
                             pattern = 'visor', full.names = TRUE) %>%
    lapply(read.csv)
  reportes_names <- list.files(ruta_entrada_reportes,
                               pattern = 'visor',
                               full.names = FALSE) %>% gsub('.csv', '', .)

  reportes_calibr <- reportes_csv %>%
    lapply(function(y) lapply(seq_along(uh),
                              function(x) dplyr::select(y, uh[[x]])) %>%
      lapply(function(x) apply(x, 1, sum)) %>%
      setNames(paste0('uhc', seq_along(uh))) %>%
      do.call(cbind, .) %>%
      cbind(y[1:2], .)) %>%
    setNames(reportes_names)

  mapply(function(x, y) write.csv(x, y, row.names = FALSE),
          x = reportes_calibr,
          y = paste0(ruta_salida_reportes,
                     names(reportes_calibr),
                     '_calibr.csv'))
}

# para caudal
qs <- list.files(ruta_q,
                 pattern = 'Q_', full.names = TRUE) %>%
  lapply(read.csv, check.names = FALSE)
qs_names <- list.files(ruta_q,
                      pattern = 'Q_', full.names = TRUE) %>% gsub('.csv', '', .)

if (isFALSE(corregir_ah)) {
  fechas <- lapply(qs, function(x) dplyr::select(x, 2:3)) %>%
    lapply(function(x) setNames(x, c('Year', 'Month'))))
}

if (isTRUE(corregir_ah)) {
  fechas <- lapply(qs, function(x) dplyr::select(x, 2:3)) %>%
    lapply(function(x) setNames(x, c('Year', 'Month'))))

  anhos <- fechas[[1]]$Year %>% unique()
  ini <- anhos %>% head(1)
  fin <- anhos %>% tail(1)

  fechas <- lapply(qs, function(x) dplyr::select(x, 2:3)) %>%
    lapply(function(x) setNames(x, c('Year', 'Month')))) %>%
    lapply(function(x) mutate(x, Year = c(rep(ini - 1, 4),

```

```

rep(seq(ini, fin - 1), each = 12),
rep(fin,8)),
Month = rep(c(9:12,1:8), length(anhos))))

# corregir salida de WEAP
qs_orig <- qs %>%
  lapply(function(x) mutate(x, Year = fechas[[1]]$Year,
                             TS = fechas[[1]]$Month))
mapply(function(x, y) write.csv(x, y, row.names = FALSE),
        x = qs_orig, y = paste0(qs_names, '.csv'))
}

datos <- lapply(qs, function(x) dplyr::select(x, -c(1:3))) %>%
  lapply(function(x) replace(x, x == -999, '')) %>%
  lapply(function(x) setNames(x, paste0('uhc', seq_along(qs))))

csv <- lapply(seq_along(qs), function(x) cbind(fechas[[x]], datos[[x]]))

mapply(function(x, y) write.csv(x, y, row.names = FALSE),
        x = csv, y = paste0(qs_names, '_visor.csv'))
}

```

En el siguiente ejemplo:

```

calibr(datos_resumen = "./datos_q_ravelo.csv",
shp_reporte = "./bh/shp/PCP_visor.shp",
shp_calibr = "./bh/shp/ravelo_calibr_solo2.shp",
reportes = TRUE,
ruta_entrada_reportes = "./bh/",
ruta_salida_reportes = "./bh/",
ruta_q = "./bh/",
corregir_ah = TRUE)

```

```

Console Terminal R Markdown
H:/mmaya/proyectos_R/visor_git/
> calibr(datos_resumen = "/datos_q_ravelo.csv",
+ shp_reporte = "/bh/shp/PCP_visor.shp",
+ shp_calibr = "/bh/shp/ravelo_calibr.shp",
+ reportes = TRUE,
+ ruta_entrada_reportes = "/bh/",
+ ruta_salida_reportes = "/bh/",
+ ruta_q = "/bh/",
+ corregir_ah = TRUE)
UH de Reporte de shp que corresponden a cada UH para shp de Calibración:
UH para "Hidro Nujchu" = 2 3 4 5 6 7 8
UH para "NG_Canchi Mayu" = 1 9
Writing layer 'ravelo_calibr' to data source 'C:/Users/HP/Documents/WEAP Areas/Balance Hidrico Ravelo FINAL/bh/s
hp/ravelo_calibr.shp' using driver 'ESRI Shapefile'
features: 2
fields: 4
geometry type: Polygon
[[1]]
NULL
[[2]]
NULL
>

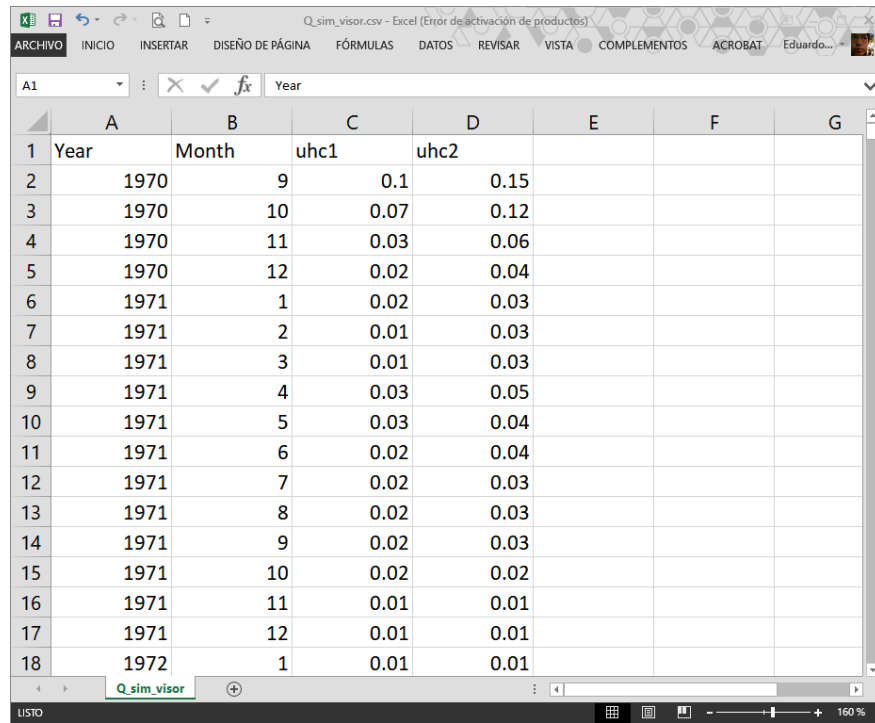
```

Figura 9: Interacción entre usuario y función para asociar UH de calibración.

se genera en la consola una aplicación interactiva con el usuario. En esta pantalla (figura 9) se le pide al usuario que asocie las unidades hidrológicas del modelo ejecutado con las unidades hidrológicas para la calibración basado en la cantidad de datos del archivo .csv de resumen de datos (figura 4). Se ingresan

números enteros separados por espacios según el código asociado a cada unidad hidrológica de la tabla del archivo de delimitación de reporte.

Al terminar de ser ejecutada, si el argumento **reporte == TRUE**, la función genera los archivos *Q\_obs\_visor.csv* y *Q\_sim\_visor.csv* (ver figura 10) y asociado a estos, el archivo de delimitación asociado a los archivos (figura 11 y cuadro 2).



	A	B	C	D	E	F	G
1	Year	Month	uhc1	uhc2			
2	1970	9	0.1	0.15			
3	1970	10	0.07	0.12			
4	1970	11	0.03	0.06			
5	1970	12	0.02	0.04			
6	1971	1	0.02	0.03			
7	1971	2	0.01	0.03			
8	1971	3	0.01	0.03			
9	1971	4	0.03	0.05			
10	1971	5	0.03	0.04			
11	1971	6	0.02	0.04			
12	1971	7	0.02	0.03			
13	1971	8	0.02	0.03			
14	1971	9	0.02	0.03			
15	1971	10	0.02	0.02			
16	1971	11	0.01	0.01			
17	1971	12	0.01	0.01			
18	1972	1	0.01	0.01			

Figura 10: Archivo csv de calibracion.

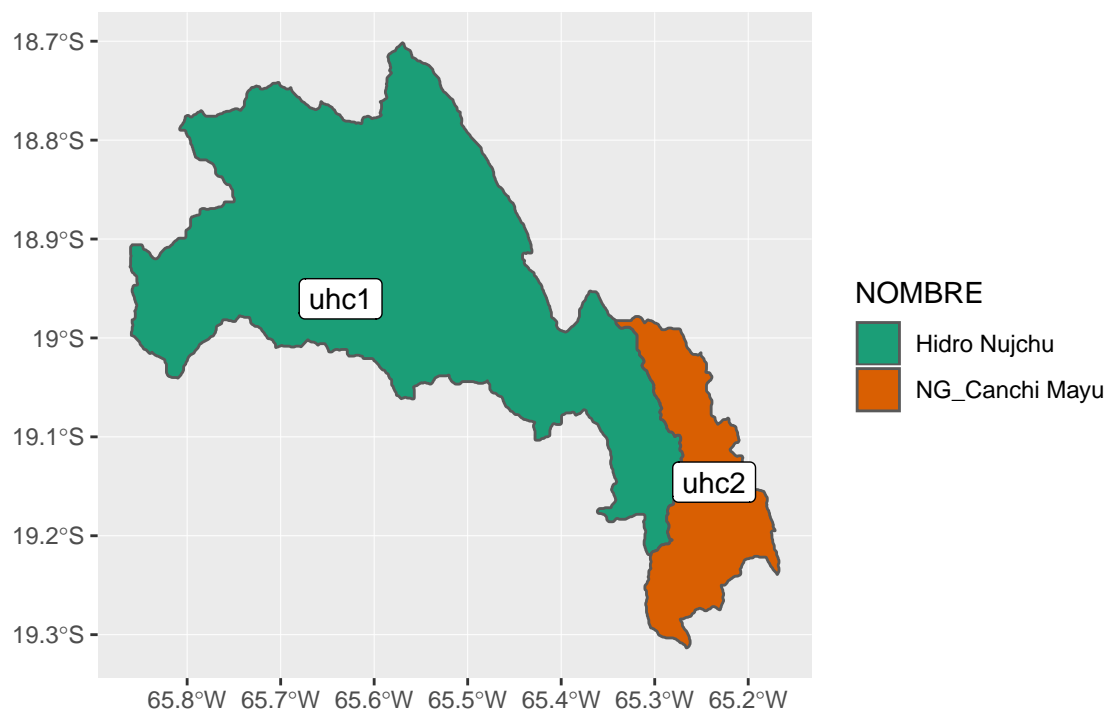


Figura 11: Representación gráfica del archivo de delimitación generado para calibración.

Cuadro 2: Tabla asociada al archivo de delimitación generado para calibración.

NOMBRE	PFAF_HYD__	OBSERV__	codigo
Hidro Nujchu	999	NA	uhc1
NG_Canchi Mayu	999	NA	uhc2