

# Informe de procesamiento de la base del Valor Agregado Bruto (VAB) territorial

Cuentas Satélite de  
Educación (CSE)  
2022

Diciembre, 2023

## 1. Introducción

El Instituto Nacional de Estadísticas y Censos (INEC) dentro de su programación estadística plurianual y en el marco del Sistema Nacional de Planificación viene elaborando las Cuentas Satélite de Educación (CSE), como una herramienta de cuantificación económica de los flujos de oferta y demanda de servicios de educación en el país.

Las CSE son un conjunto de cuentas y cuadros estadísticos elaborados bajo el marco de referencia del SCN, proporciona información económica detallada del sector de la educación, enriquece el análisis sectorial cuando se integran sus datos monetarios con datos físicos, lo cual se complementa con el análisis del sector en el contexto de la economía nacional.

En función al Programa Nacional de Estadística 2021-2025 del INEC, durante el presente año se actualizaron las CSE al periodo de información 2022. La construcción de las CSE deberá estar acorde a las actividades establecidas en cada fase del Modelo de Producción Estadística. El modelo está compuesto por ocho fases: planificación, diseño, construcción, recolección, procesamiento, análisis, difusión y evaluación.

Bajo estas premisas, el presente documento describe el proceso automatizado para la construcción de la base de datos del VAB a nivel provincial periodo 2022, mediante el uso de sintaxis con el software de uso libre "R".

## 2. Objetivos

Detallar el proceso de construcción de la base de datos del VAB con desagregación provincial mediante sintaxis en el programa R y la generación de mapas provinciales sobre el comportamiento de la Producción, CI y VAB de la enseñanza.

## 3. Desarrollo

### 3.1. Aspectos conceptuales

Conforme al marco de buenas prácticas estadísticas, el proceso inicia con la revisión y análisis de metodologías, recomendaciones y prácticas internacionales aplicadas al sistema educativo como la Clasificación Internacional Normalizada de la Educación, 2011, la Metodología de Cuentas Nacionales de la Educación, 2016, el "Manual on regional accounts methods" de Eurostat (2013) y otros documentos técnicos sobre medición regional del PIB, como el Sistema de Cuentas Nacionales de México y Ecuador.

A continuación, el proceso inicia con la obtención de indicadores monetarios de producción, gasto, ingreso, salario, entre otros, a nivel de cada una de las industrias de la educación y desagregado por provincia. Los datos monetarios se complementan con la información de indicadores físicos como el número de

docentes, número de alumnos o beneficiarios según niveles educativos y por cada provincia.

En función de la información disponible para el cálculo de un agregado económico, como la producción o valor agregado bruto a nivel territorial, se combinan métodos ascendentes y descendentes. Estos métodos son utilizados en función de la existencia de información directa para la medición del fenómeno económico a nivel territorial (método ascendente); mientras que, en los casos con ausencia de información que permita la medición directa a nivel territorial se utiliza información clave y relacionada como los indicadores físicos (docentes y alumnos) para la distribución de valores del fenómeno económico a nivel territorial.

Desde esta perspectiva, la información presupuestaria a nivel de micro datos para instituciones educativas y distritos educativos permitió construir las estadísticas territoriales desde un enfoque "ascendente". Mientras la información de indicadores físicos como los alumnos y docentes permitió construir la información estadística territorial desde un enfoque "descendente". Al final, los valores provinciales se reconcilian con el agregado a nivel nacional para que lo provincial sea consistente con la información de las industrias (niveles educativos) a escala nacional.

A continuación, se describe el proceso para construir la base de datos del VAB provincial, mediante una exploración a la sintaxis construida con el software de uso libre "R".

### **3.2. Insumos**

Para la construcción de esta sintaxis se requieren los siguientes insumos:

1. Base de datos unificada: esta base de datos incluye información económica primaria de los sectores institucionales del gobierno general (GG), sociedades no financieras (SNF), instituciones sin fines de lucro (ISFLH) y hogares productores a nivel provincial. No obstante, la información provincial no está disponible para todos los sectores o subsectores, especialmente del sector privado; por lo cual, para estos segmentos, los valores de producción provincial son estimados bajo un método descendente que toma a la distribución provincial de beneficiarios como variable clave de asignación provincial.
2. Matrices de distribución económica: se incluyen matrices previamente distribuidas en los valores económicos a nivel provincial para los programas educativos del MINEDUC, coordinaciones zonales y hogares productores.
3. Matrices de distribución de beneficiarios: estas matrices se generan para todas las industrias a nivel desagregado de las CSE. Por lo cual, se aplican siempre que la información primaria de la base unificada no tenga información provincial suficiente o simplemente carezca de ella.

4. Nomenclaturas de las CSE: clasificador de nomenclaturas que correlaciona los sectores institucionales con las industrias de enseñanza (niveles educativos).
5. Clasificador geográfico: corresponde al clasificador de codificación provincial del INEC.
6. VAB, Producción y Consumo intermedio a nivel nacional y por industrias: esta información se obtiene con el procesamiento de las bases de datos de Economía Total y VAB Nacional.

Con estos insumos se desarrolla el código de programación que homologa e integra los insumos de distribución y posteriormente genera el cálculo de las variables que conforman la base de datos del VAB provincial.

### 3.3. Código de programación

A continuación, se observa en detalle el proceso que contiene la sintaxis elaborada para la construcción de la base de datos de Economía Total.

El procesamiento inicia con el tratamiento de la base unificada (microdato a nivel provincial) y matrices con valores económicos previamente distribuidas según niveles de educación (industrias) y a nivel provincial.

```
# a. Llamar paquetes ----
library(tidyverse)
library(tidyr)
library(openxlsx)
library(reshape2)
library(mice)
library(rpivotTable)
library(cartography)
library(sf)
library(mapsf)

#----- FASE 5. PROCESAMIENTO -----
# VAB TERRITORIAL DE LAS CSE
#-----

ini<-2018
ini_territ = 2020
fin<-2022

#direccion0 <- "C:/Users/HVALDIVIEZO/Desktop/GESTION_GASIN/2023/5_PROC/CSE/RESULTADOS_01"
# Generación de indicadores de producción, consumo intermedio y VAB territorial
# Preparación de insumos

# Insumos VAB corriente constante nacional y matrices con claves de distribución provincial
# 1 Cargar funciones de distribución y consolidación de datos
# 2 Construir matrices de distribución provincial por niveles
# 3 VAB, Prod, CI Nacional
# 4 Procesamiento de matrices con claves de distribución provincial
# 5 Procesamiento Base de datos VAB Provincial

### Procesamiento de base unificada con distribución de niveles de educación a nivel provincial

### 1 Funciones de distribución y consolidación de datos

# Se preparan funciones para consolidar datos, distribuir y ajustar valores

setwd(area_trabajo)
source("Fun_CS_v02.r")
```

```
#archivo_matriz = "3_CSE_MatDist_2007-21_alumnos.xlsx"
archivo_matriz = "3_CSE_MatDist_2007-22_f.xlsx"

### 2. Construir y homologar matrices de distribución

#Clasificador geográfico - provincial
#clasifi_prov = read.xlsx("5_Matriz_Territ_CSE_2018_21_f.xlsx",sheet = "Clasif_Geog",
#
#       rows = c(5:29))
mt_coef_territ = "5_Matriz_Territ_CSE_22_f_dupli.xlsx"

#mt_coef_territ = "5_Matriz_Territ_CSE_22_F.xlsx"

clasifi_prov = read.xlsx(mt_coef_territ,sheet = "Clasif_Geog",
#       rows = c(5:29))
# Matriz a nivel provincial
mtd2 = read.xlsx(archivo_matriz,sheet = 1,startRow = 1)
mtd2$Total = NULL

mtd2 = melt(mtd2,id= c("ejercicio","descripcion","id_registro"),value.name = "distrib",
#       na.rm = T)
mtd2 = mtd2 %>% filter(distrib>0)

#archivo = "BUCSE_20-22.xlsx"
archivo =
"R:/CGTPE/DECON/AS/CS_MPE_2023/CSE_2022/5_Proc/5.1_Crit_integr/5.1.3_Sincr_bas_dat/1_BDUCSE_2022_fase1/BUCSE_20-
22.xlsx"
bdp<- read.xlsx(archivo)
bdp$ejercicio<- as.numeric(bdp$ejercicio)

#bdp<- bdp %>% rename(cod_provincia = cod_provincia_f, descr_provincia = descr_provincia_f)
#Recodificar a nacional datos incompletos en gobiernos locales y ciclo corto

bdp<- bdp %>% mutate(cod_provincia=ifelse(codigo_N6 == "S13.01.05.01.02.01","00",cod_provincia),
#       cod_provincia=ifelse(substr(codigo_N6,1,6) == "S13.02","00",cod_provincia),
#       cod_provincia=ifelse(codigo_N6 == "S13.01.01.02.01.05","00",cod_provincia))

sum(bdp$devengado,na.rm = T)
table(is.na(bdp$cod_provincia))
if (class(bdp$cod_provincia)=="integer") {
#       bdp = bdp %>% mutate(cod_provincia=ifelse(is.na(cod_provincia),99,cod_provincia),
#       cod_provincia = ifelse(cod_provincia<10,paste0("0",cod_provincia),
#       as.character(cod_provincia)))
}
table(is.na(bdp$cod_provincia))
table((bdp$cod_provincia))
unique(bdp$cod_provincia)

nrow(bdp)

#### S14
bdp1 = bdp %>% filter(substr(codigo_N6,1,6)=="S14.01" & ejercicio>=ini)
mtd1 = read.xlsx(mt_coef_territ,sheet = "S14",startRow = 6)
mtd1 = mtd1 %>% select(ejercicio,codigo_N6,cod_provincia,descr_provincia,valor)
mtd1 = mtd1 %>% mutate(cod_provincia = as.numeric(cod_provincia))

mtd1 = mtd1 %>% mutate(cod_provincia=ifelse(is.na(cod_provincia),NA,cod_provincia),
#       cod_provincia = ifelse(cod_provincia<10,paste0("0",cod_provincia),
#       as.character(cod_provincia)))
table(is.na(mtd1$cod_provincia))
unique(mtd1$cod_provincia)

bdp1 = bdp1 %>% select(ejercicio,tipo,codigo_N1,codigo_N6,descr_codigo_N6,cod_CN,devengado)
sum(bdp1$devengado)
bdp1 = bdp1 %>% left_join(mtd1, multiple = "all")

bdp1 = bdp1 %>%
#       group_by(ejercicio,codigo_N1,codigo_N6,descr_codigo_N6,tipo,cod_CN) %>%
#       nest() %>%
#       mutate(mod_obj = map(data, ~ participa(., "cod_provincia", "valor",
#       "devengado", nuevo_df = 2), id="ver")) %>%
#       select(ejercicio,codigo_N1,codigo_N6,descr_codigo_N6,tipo,cod_CN,mod_obj) %>%
```

```

unnest(mod_obj)
sum(bdp1$devengado_dist, na.rm = T)
bdp1$devengado = bdp1$devengado_dist
bdp1 = bdp1 %>% select(ejercicio, tipo, codigo_N1, codigo_N6, descr_codigo_N6, cod_CN,
                      cod_provincia, devengado)
bdp1 = bdp1 %>% left_join(clasifi_prov[, 1:2])
bdp1$fuente = "BBD unificada hogares"

bdp = bdp %>% filter(! (codigo_N6 %in% unique(bdp1$codigo_N6) & ejercicio>=ini))
bdp = bind_rows(bdp, bdp1)
sum(bdp$devengado, na.rm = T)

table(is.na(bdp$cod_provincia))
unique(bdp$cod_provincia)

#### universidades distribución por sedes
sum(bdp$devengado, na.rm = T)

bdp1 = bdp %>% filter(id_registro %in% c("1034", "1031", "1029", "1044", "1041", "1042",
                                         "1038", "1002", "1079", "1012", "1058") & ejercicio>=ini)

mtd1 = rbind(read.xlsx(mt_coef_territ, sheet = "C6.01.01", startRow = 6),
              read.xlsx(mt_coef_territ, sheet = "C6.02.01", startRow = 6))
mtd1 = mtd1 %>% filter(id_registro %in% c("1034", "1031", "1029", "1044", "1041", "1042",
                                         "1038", "1002", "1079", "1012", "1058"))

# UTPL se gestiona desde información salarial de las sedes, si los centros tienen más de 5
# trabajadores se considera la producción
mtd1 = mtd1 %>%
  mutate(beneficiarios = ifelse(id_registro=="1031", ifelse(round(as.numeric(docentes))>4,
                                                            round(as.numeric(docentes)), 0), beneficiarios))
# Epoch se reduce el costo de la planta administrativa de la sede central y el resto se distribuye
# según alumnos, el porcentaje identificado es 24%.
mtd1 = mtd1 %>%
  mutate(beneficiarios = ifelse(id_registro=="1002" & cod_provincia!="06",
                                beneficiarios*(1-0.24), beneficiarios))

mtd1 = mtd1 %>% select(ejercicio:beneficiarios)

bdp1 = bdp1 %>% group_by(ejercicio, id_registro, tipo, codigo_N1, codigo_N6, descr_codigo_N6, cod_CN) %>%
  summarise(devengado=sum(devengado, na.rm = T))

sum(bdp1$devengado)
bdp1 = bdp1 %>% left_join(mtd1, multiple = "all")

bdp1 = bdp1 %>%
  group_by(ejercicio, id_registro, codigo_N1, codigo_N6, descr_codigo_N6, tipo, cod_CN) %>%
  nest() %>%
  mutate(mod_obj = map(data, ~ participa(., "cod_provincia", "beneficiarios",
                                         "devengado", nuevo_df = 2))) %>%
  select(ejercicio, id_registro, codigo_N1, codigo_N6, descr_codigo_N6, tipo, cod_CN, mod_obj) %>%
  unnest(mod_obj)

sum(bdp1$devengado_dist, na.rm = T)
bdp1$devengado = bdp1$devengado_dist
bdp1 = bdp1 %>% select(ejercicio, id_registro, tipo, codigo_N1, codigo_N6, descr_codigo_N6, cod_CN,
                      cod_provincia, devengado)

bdp1 = bdp1 %>% left_join(clasifi_prov[, 1:2])
mtd1 = mtd1 %>% group_by(id_registro, descr_entid_inst) %>% summarise(n())
bdp1 = bdp1 %>% left_join(mtd1[, 1:2])
bdp1$fuente = "BBD universidades sede distribuida"

bdp = bdp %>% filter(! (id_registro %in% c("1034", "1031", "1029", "1044", "1041", "1042",
                                         "1038", "1002", "1079", "1012", "1058") & ejercicio>=ini))
bdp = bind_rows(bdp, bdp1)
sum(bdp$devengado, na.rm = T)

### Distribución coordinaciones zonales y planta central SENESCYT
sum(bdp$devengado, na.rm = T)

```

```
ids_senescyt = c("10111413158","10111414159","10111415154","10111416150","10111417155",
  "10111418150","10111419156","10111420151","10111421150")

bdp1 = bdp %>% filter(id_cs %in% ids_senescyt & ejercicio>=ini)
glimpse(bdp1)

bdp1 = bdp1 %>% mutate(id_cs = "10111421150")

mtd1 = read.xlsx(mt_coef_territ,sheet = "C1.01.01_senescyt",startRow = 6)

mtd1 = mtd1 %>% select(ejercicio:administrativo)

bdp1 = bdp1 %>% group_by(ejercicio,id_cs,tipos,codigo_N1,codigo_N6,descr_codigo_N6,cod_CN) %>%
  summarise(devengado=sum(devengado,na.rm = T))

sum(bdp1$devengado)
bdp1 = bdp1 %>% left_join(mtd1)

bdp1 = bdp1 %>%
  group_by(ejercicio,id_cs,codigo_N1,codigo_N6,descr_codigo_N6,tipos,cod_CN) %>%
  nest() %>%
  mutate(mod_obj = map(data, ~ participa(., "cod_provincia", "administrativo",
    "devengado", nuevo_df = 2))) %>%
  select(ejercicio,id_cs,codigo_N1,codigo_N6,descr_codigo_N6,tipos,cod_CN,mod_obj) %>%
  unnest(mod_obj)

sum(bdp1$devengado_dist,na.rm = T)
bdp1$devengado = bdp1$devengado_dist
bdp1 = bdp1 %>% select(ejercicio,id_cs,tipos,codigo_N1,codigo_N6,descr_codigo_N6,cod_CN,
  cod_provincia,devengado)

bdp1 = bdp1 %>% left_join(clasifi_prov[,1:2], multiple = "all")
bdp1$fuente = "BBD senescyt distribuida"
# bdp1$id_cs = as.character(bdp1$id_cs)

bdp = bdp %>% filter(!(id_cs %in% ids_senescyt & ejercicio>=ini))

bdp = bind_rows(bdp,bdp1)
sum(bdp$devengado,na.rm = T)

glimpse(bdp1)

## Distribución coordinaciones zonales MINEDUC

# Distribución de zonales en distritos
mtd3 = read.xlsx(archivo_matriz,sheet = 2,startRow = 1)

bdp2 = bdp %>% filter(id_registro %in% unique(mtd3$id_registro) & ejercicio>=ini_territ)
bdp = bdp %>% filter(!(id_registro %in% unique(mtd3$id_registro) & ejercicio>=ini_territ))

bdp2 = bdp2 %>% group_by(ejercicio,tipos,codigo_N6,id_registro,id_cs,
  cod_provincia,cod_CN) %>% summarise(devengado=round(sum(devengado,na.rm = T)))
# Valor que se imputa a las coordinaciones zonales, similar a distritales = 0,10661957
bdp2 = bdp2 %>% group_by(ejercicio,tipos,codigo_N6,id_registro,id_cs,
  cod_provincia,cod_CN) %>% summarise(devengado=round(sum(devengado,na.rm = T)))
bdp2 = bdp2 %>% mutate(devengado1 = devengado - round(devengado*(1-0.10661957)),
  devengado = devengado - devengado1)
zonales = bdp2 %>% select(-devengado,devengado = devengado1)
zonales$fuente = "Distribuido desde BBD unificada zonales"

bdp2 = bdp2 %>% select(-devengado1)

bdp2 = bdp2 %>% mutate(descripcion = ifelse(cod_CN %in% c("P.2","P.51"),"Alumnos","Docentes"))
sum(bdp2$devengado,na.rm = T)

bdp2 = merge(as.data.frame(bdp2),mtd3,by = c("ejercicio","descripcion","id_registro"),all.x = T)

bdp2 = bdp2 %>%
  mutate(id_registro = cod_distrito)
```

```

bdp2 = bdp2 %>%
  group_by(ejercicio,descripcion,tipo,codigo_N6,id_cs,cod_provincia,cod_CN) %>%
  nest() %>%
  mutate(mod_obj = map(data, ~ participa(., "id_registro", "Porcentaje",
    "devengado", nuevo_df = 2), id = "ver")) %>%
  select(ejercicio,descripcion,tipo,codigo_N6,id_cs,cod_provincia,cod_CN,mod_obj) %>%
  unnest(mod_obj)
bdp2$devengado = bdp2$devengado_dist
bdp2$codigo_N6 = "S13.01.01.02.01.02"
sum(bdp2$devengado, na.rm = T)
# bdp2$cod_provincia = bdp2$cod_provincia %>% replace_na(99)
table(is.na(bdp2$cod_provincia))
bdp2$fuente = "Distribuido desde BBD unificada zonales"

zonales = bind_rows(zonales, bdp2)
zonales = zonales %>% select(-c("descripcion", "Porcentaje", "clave_dist", "devengado_dist"))
sum(zonales$devengado)
bdp = bind_rows(bdp, zonales)

#####
# S13, S11 y S15
mtd_distr_prov =
  read.xlsx(mt_coef_territ, sheet = "Distrito_prov", startRow = 1)

mtd_distr_prov = mtd_distr_prov %>% filter(cod_distrito != "00000") %>% select(cod_distrito, cod_provincia)
mtd_distr_prov = mtd_distr_prov %>% group_by(cod_distrito) %>% distinct(cod_provincia)

bdp2 = bdp %>% filter(id_registro %in% unique(mtd2$id_registro) & ejercicio >= ini_territ)

bdp = bdp %>% filter(!(id_registro %in% unique(mtd2$id_registro) & ejercicio >= ini_territ))
nrow(bdp) + nrow(bdp2)

bdp2 = bdp2 %>% select(ejercicio, tipo, codigo_N1, codigo_N6, descr_codigo_N6, id_registro,
  id_cs, cod_provincia, descr_provincia, cod_CN, devengado)

bdp2 = bdp2 %>% group_by(ejercicio, tipo, codigo_N6, id_registro, id_cs,
  cod_provincia, cod_CN) %>% summarise(devengado = round(sum(devengado, na.rm = T)))

temp = bdp2 %>% filter(id_registro %in% unique(mtd_distr_prov$cod_distrito))
temp$cod_provincia = NULL

temp = merge(temp, mtd_distr_prov, by.x = "id_registro", by.y = "cod_distrito")
temp = temp %>% group_by(ejercicio, tipo, codigo_N6, id_registro, id_cs,
  cod_provincia, cod_CN) %>% summarise(devengado = round(sum(devengado, na.rm = T)))

bdp2 = bdp2 %>% filter(!(id_registro %in% unique(mtd_distr_prov$cod_distrito)))
bdp2 = bind_rows(bdp2, temp)

bdp2 = bdp2 %>% mutate(descripcion = ifelse(cod_CN %in% c("P.2", "P.51"), "Alumnos", "Docentes"))
sum(bdp2$devengado, na.rm = T)

bdp2 = merge(as.data.frame(bdp2), mtd2, by = c("ejercicio", "descripcion", "id_registro"), all.x = T)
bdp2 = bdp2 %>%
  mutate(id_registro = ifelse(id_registro %in% c("S11.01.0X", "S11.01.0E", "S15.01.02",
    "S15.01.03", "S15.01.04", "S15.01.0E"),
    id_cs, id_registro))

bdp2 = bdp2 %>% mutate(descripcion = ifelse(cod_CN %in% c("P.2", "P.51"), "Alumnos", "Docentes"))
sum(bdp2$devengado, na.rm = T)

bdp2 = mutate(bdp2, v_dist = (devengado * distrib))
sum(bdp2$v_dist)

write.xlsx(bdp2, "revisión_terr.xlsx")
table(is.na(bdp2$cod_provincia))

# Ajustar diferencias por distribución
Sys.time()
bdp2 = bdp2 %>%
  group_by(ejercicio, tipo, codigo_N6, cod_provincia, id_registro, id_cs, cod_CN, descripcion) %>%

```

```

nest() %>%
mutate(mod_obj = map(data, ~ ajusta_dist(., "devengado", "v_dist"), id="ver")) %>%
select(ejercicio, tipo, codigo_N6, cod_provincia, id_registro, id_cs, cod_CN, descripcion, mod_obj) %>%
unnest(mod_obj)
Sys.time()

bdp2 = bdp2[bdp2$v_dist != 0,]
sum(bdp2$v_dist, na.rm = T)

sum(bdp2$v_dist, na.rm = T)

# View(bdp2)
bdp2$id_grupo = NULL
bdp2$codigo_N6 = bdp2$variable
bdp2$codigo_N1 = substr(bdp2$codigo_N6, 1, 3)
bdp2$devengado = bdp2$v_dist
bdp2$variable = NULL
bdp2$v_dist = NULL
bdp2$distrib = NULL
bdp2$descripcion = NULL
bdp2$fuente = "ValorDistribuidoNiveles"
bdp = bind_rows(bdp, bdp2)

# incluir matriz de programas educativos

bdp1 = read.xlsx("5_Progr_MINEDUC_2020-22.xlsx", sheet = 1, startRow = 1)
bdp1 = pivot_longer(bdp1, $13.01.02.02.01.01:$13.01.04.02.03.01, names_to = "codigo_N6",
  values_to = "devengado")
bdp1 = bdp1 %>% select(ejercicio, descr_programa=programa, id_registro, cod_provincia,
  descr_provincia, codigo_N6, devengado)
bdp1 = bdp1 %>% mutate(codigo_N1 = substr(codigo_N6, 1, 3), cod_CN = "P.2",
  fuente = "Programas educativos MINEDUC", tipo = "2")
bdp = bind_rows(bdp, bdp1)

dir_terri = paste0(direccion0, "/Prov")
dir.create(dir_terri)

setwd(dir_terri)
write.csv(bdp, "1_bdg_distr_prov_niveles.csv")

#pivotTable(bdp)

setwd(direccion0)

# Cargar insumos para correr la función ----
# Base del VAB
VAB <- read.xlsx("VAB22.xlsx")
setwd(area_trabajo)

# Archivo de nomenclaturas
clasif <-
read.xlsx("R:\\CGTPE\\DECON\\AS\\CS_MPE_2022\\CSE_2020_21\\5_Proc\\5.7_Finali_archiv_dat\\5.7.2_Compil_prod_ant
\\6_Base_VAB\\0_Arch_Procesam\\1_Deli_CSE_2020-21.xlsx")

if (is.null(bdp)) { # Se requiere base de datos con microdato a nivel provincial
  stopifnot(is.null(VAB)) # clasif <-
  read.xlsx("R:\\CGTPE\\DECON\\AS\\CS_MPE_2022\\CSE_2020_21\\5_Proc\\5.7_Finali_archiv_dat\\5.7.2_Compil_prod_ant
\\6_Base_VAB\\0_Arch_Procesam\\1_Deli_CSE_2020-21.xlsx")
  clasif <- read.xlsx("2_Deli_CSE_2022.xlsx", sheet = "1.12 correspondencia2022", startRow=8, cols = c(1:31, 34:43))
  clasif <- clasif[!duplicated(clasif$codigo_N6),]

  #if (is.null(bdp)) { # Se requiere base de datos con microdato a nivel provincial
  # stopifnot(is.null(VAB))

  ### 2 Cargar VAB, Prod, CI Nacional
  VAB_N = VAB %>% filter(PRODUCTOS %in% c("P.1", "P.2", "B.1b") & ejercicio >= ini_territ) %>%
  group_by(ejercicio, codigo_industria=cod_industria_N3, PRODUCTOS) %>%
  summarise(Corriente=sum(Corriente, na.rm = T), Constante=sum(Constante, na.rm = T)) %>%
  gather(key = "Tipo", value = "Total", Corriente, Constante) %>%
  pivot_wider(names_from = c( PRODUCTOS, Tipo), values_from = Total)
  VAB_N$ejercicio = as.numeric(VAB_N$ejercicio)

```

### ### 3.1 Procesar Matrices de distribución desde microdato

```
bdp2 = bdp %>% filter(cod_provincia %in% c("01","02","03","04","05","06","07","08","09",
      "10","11","12","13","14","15","16","17","18",
      "19","20","21","22","23","24"))

bdp2 = bdp2 %>% filter((cod_CN=="P.11" | (tipo==2 & cod_CN %in% c("D.11","D.121",
      "D.122","D.29","P.2")))) %>%
select(ejercicio,codigo_N6,cod_provincia,tipo,cod_CN,devengado)

clasif_temp = clasif[!duplicated(clasif$codigo_N6),c("codigo_N6","codigo_N1","cod_industria_N3","descr_industria_N3")]

bdp2 = bdp2 %>% left_join(clasif_temp,by="codigo_N6")

bdp2 = as.data.frame(bdp2)
glimpse(bdp2)

bdp2 = bdp2 %>%
  group_by(ejercicio,codigo_industria=cod_industria_N3,descr_codigo_industria=descr_industria_N3,
    codigo_N1,codigo_N6,cod_provincia,tipo) %>%
  summarise(devengado=sum(devengado,na.rm=T)) %>%
  pivot_wider(names_from = tipo,values_from=devengado,names_prefix="dev") %>%
  mutate(produccion=ifelse(codigo_N1 %in% c("S11","S14"),dev1,dev2)) %>%
  select(-starts_with("dev"))

rpivotTable(bdp2)
```

Una vez procesadas las bases de datos con información económica a nivel provincial, el siguiente paso se orienta a procesar una base de datos con claves de distribución basada en datos físicos de beneficiarios y docentes en cada industria de la educación y a nivel provincial.

Al finalizar el proceso se tiene una matriz consolidada de distribución que recoge las claves de distribución basada en: i) valores económicos provinciales obtenidos desde el microdato, ii) valores de alumnos (beneficiarios) distribuidos según industria y provincia, iii) valores de docentes distribuidos al igual que los alumnos.

Como se mencionó previamente, los valores económicos registran un nivel de cobertura alto para el sector público, mientras que para el sector privado se observó falta de cobertura de información en varias provincias. Por ende, el sector privado en gran medida se estima mediante las claves de alumnos o docentes a nivel provincial.

### # 3.2 Generación de matrices distribuidas

```
#setwd(area_trabajo)
#Datos de docentes y estudiantes a nivel territorial
setwd(area_trabajo)

wb = mt_coef_territ
nombres = getSheetNames(wb)
vhojas = nombres[c(3:20)]

MDist = consolidaHojas(wb,vhojas,finicio=6)
unique(MDist$id_hoja)
MDist = MDist %>% filter(id_hoja!="C6.01.01_2")

MDist = drop_na(MDist,id_hoja)

MDist = MDist %>% mutate(cod_provincia=ifelse(is.na(cod_provincia),NA,cod_provincia),
  cod_provincia = ifelse(as.numeric(cod_provincia)<10,paste0("0",as.numeric(cod_provincia)),
    as.character(cod_provincia)))

glimpse(MDist)
MDist = MDist %>% mutate(docentes = as.numeric(docentes))
```

```

MDist = MDist %>% group_by(ejercicio,cod_provincia,codigo_industria,
  descr_codigo_industria) %>%
  summarise(beneficiarios = sum(beneficiarios,na.rm = T),
    docentes = sum(docentes,na.rm = T),produccion = sum(produccion,na.rm = T)) %>%
  drop_na(ejercicio)

MDist22 = clasif %>%
  group_by(codigo_industria=cod_industria_N3, cod_industria_N2, descr_industria_N2,
    cod_industria_N1, descr_industria_N1, sector) %>%
  summarise(n()) %>% filter(codigo_industria!="99999") %>%
  select("-n()") %>% right_join(MDist, multiple = "all")

# base de datos para VAB por alumno
Alumnos_prov = MDist %>% filter(substr(codigo_industria,1,2) %in% c("02","03","04","05","06")) %>%
  select(1:5)

clasif_ind = clasif %>%
  group_by(codigo_industria=cod_industria_N3, descr_codigo_industria= descr_industria_N3,
    cod_industria_N2,descr_industria_N2,
    cod_industria_N1, descr_industria_N1) %>% summarise(n()) %>% select(-7)

Alumnos_prov = Alumnos_prov %>% left_join(clasif_ind)

write.xlsx(Alumnos_prov,"Alumns_prov_res22.xlsx")

Alumnos_prov_origen = Alumnos_prov

# Imputación valores faltantes

# Se añade un valor mínimo por pérdida de declaraciones en provincias con bajo nivel de formalidad

xx = MDist %>% group_by(ejercicio,codigo_industria,descr_codigo_industria) %>%
  mutate(beneficiarios = imputa_aditivo(beneficiarios,0.002,0.25 ),
    docentes = imputa_aditivo(docentes,0.002,0.25 ))

# Datos de producción a nivel territorial
temp1 = expand.grid(ejercicio=2018:fin,codigo_industria=c("08.01.0","09.01.0","01.01.01",
  "10.01.0","11.01.0","12.01.0","13.01.0","02.01.01","02.01.02","02.02.02","03.01.01",
  "03.02.01","04.01.01","04.01.02","04.02.01","04.02.02","06.01.01","06.02.01","07.01.01",
  "07.02.01","02.02.01","05.01.01","05.02.01"),
  cod_provincia=c("01","02","03","04","05","06","07","08","09","10","11","12","13",
    "14","15","16","17","18","19","20","21","22","23","24"))
temp1$ejercicio = as.numeric(temp1$ejercicio)
temp1$codigo_industria = as.character(temp1$codigo_industria)

# Matriz a nivel de industria con valores de producción desde la fuente

temp = bdp2 %>% group_by(ejercicio,cod_provincia,codigo_industria) %>%
  summarise(valor = sum(produccion,na.rm = T)) %>%
  filter(ejercicio>=ini)
# Matriz a nivel de industria con valores de producción y otras variables para distribución
temp = temp %>% right_join(temp1)

temp = temp %>% left_join(MDist[,c("ejercicio","cod_provincia",
  "codigo_industria","produccion","beneficiarios","docentes")])

temp = clasif %>% group_by(codigo_industria=cod_industria_N3, descr_codigo_industria = descr_industria_N3, tipo_activ)
%>%
  summarise(n()) %>% filter(codigo_industria!="99999") %>%
  select(codigo_industria, descr_codigo_industria, tipo_activ) %>% right_join(temp) %>%
  left_join(clasifi_prov[,1:2])

xxx = temp
temp = xxx

#temp$valor[is.na(temp$valor)] = 0
# No se distribuyen
MDist = temp %>% filter(!{tipo_activ=="Característico" & !(codigo_industria %in% c("01.01.01","02.02.01",
  "05.01.01","05.02.01","06.01.01","06.02.01"))})

```

```

# Se distribuyen
temp = temp %>%
  filter(tipo_activ=="Característico" & !(codigo_industria %in% c("01.01.01","02.02.01",
    "05.01.01","05.02.01","06.01.01","06.02.01")))

# Imputación valores faltantes

temp = temp %>% mutate(valor = ifelse(valor==0 & (beneficiarios>0 | docentes > 0),NA,valor),
  beneficiarios = ifelse(beneficiarios==0 & docentes > 5 ,NA,beneficiarios),
  docentes = ifelse(docentes>5 & beneficiarios > 0,NA,docentes))

sapply(temp, function(x) sum(is.na(x)))

cor(temp$valor,temp$beneficiarios,use = "complete.obs")
cor(temp$valor,temp$docentes,use = "complete.obs")

# Ejecutamos una imputación múltiple: 1) Generamos imputaciones 2) Estimamos coeficientes puntuales
# 3) Agrupamos los estimadores 4) Predecimos valores
# Generamos imputaciones múltiples m=100

init = mice(temp, maxit=0)
meth = init$method
predM = init$predictorMatrix
predM[, c("valor")]=0
meth[c("codigo_industria","descr_codigo_industria","tipo_activ",
  "ejercicio","cod_provincia","produccion","descr_provincia")]=""
temp1 = mice(temp, method=meth, predictorMatrix=predM, m=50,seed = 2021)
summary(temp1)
densityplot(temp1,subset = cod_provincia!="17" & cod_provincia!="09" & cod_provincia!="13")

#Excluimos provincias grandes para observar los resultados de la imputación
#xyplot(temp1,valor ~ beneficiarios | .imp,pch = c(1, 10), cex = c(0.5, 0.5),
# subset = cod_provincia!="17" & cod_provincia!="09" & cod_provincia!="13" )
xyplot(temp1,valor ~ beneficiarios | cod_provincia,pch = c(1, 10), cex = c(0.5, 0.5),
  subset = cod_provincia!="17" & cod_provincia!="09" & cod_provincia!="13")
xyplot(temp1,valor ~ beneficiarios | codigo_industria,pch = c(1, 10), cex = c(0.5, 0.5),
  subset = cod_provincia!="17" & cod_provincia!="09"& cod_provincia!="13")
xyplot(temp1,valor ~ beneficiarios,pch = c(1, 20), cex = c(0.5, 1.5),
  subset = cod_provincia!="17" & cod_provincia!="09"& cod_provincia!="13")

#Estimación del modelo predictivo en cada imputación
lm_valor=with(temp1,lm(valor~-1+beneficiarios+docentes))
lm_benefi=with(temp1,lm(beneficiarios~-1+docentes))
lm_docent=with(temp1,lm(docentes~-1+beneficiarios))

lm_valor=pool(lm_valor)
lm_benefi=pool(lm_benefi)
lm_docent=pool(lm_docent)
summary(lm_valor)
summary(lm_benefi)
summary(lm_docent)

# Opción 1 de imputación, tomar valores promedio de las multiples imputaciones

temp1 = complete(temp1,action="long",include=T)
temp1 = temp1 %>% group_by(ejercicio,tipo_activ,codigo_industria,cod_provincia,
  descr_codigo_industria,descr_provincia,.id ) %>%
  summarise(valor=mean(valor,na.rm=T),beneficiarios=mean(beneficiarios,na.rm=T),
    docentes=mean(docentes,na.rm=T),produccion=mean(produccion,na.rm=T)) %>%
  select(-.id)
temp1$produccion = temp1$valor
sapply(temp1, function(x) sum(is.na(x)))

# Opción 2 de imputación, predecir valores con el modelo ajustado
x1 = lm_benefi$pooled$estimate[1]
x2 = lm_docent$pooled$estimate[1]
x3 = lm_valor$pooled$estimate[1:2] #1 beneficiarios 2 docentes

temp = temp %>% mutate(beneficiarios = ifelse(is.na(beneficiarios),docentes*x1,beneficiarios),
  docentes = ifelse(is.na(docentes),beneficiarios*x2,docentes),
  valor = ifelse(is.na(valor),beneficiarios*x3[1]+docentes*x3[2],valor))

```

```

temp$produccion = temp$valor

sapply(temp, function(x) sum(is.na(x)))

par(mfrow=c(1,2))
plot(round(temp$beneficiarios/1000),round(temp$valor/1000))
plot(round(temp$docentes),round(temp$valor/1000))

dev.off()

# Se trabajará con la opción 2 de imputación por ser robusta
# -----

# Organización de base de datos
MDist = bind_rows(MDist,temp)
MDist = MDist %>% mutate( valor = ifelse(is.na( valor),0,valor), produccion = valor )
MDist$valor = round(MDist$valor)

MDist$valor = NULL
MDist = MDist %>% pivot_longer(cols=c(beneficiarios,docentes,produccion),names_to = "clave",
                                values_to = "valor")
MDist = MDist %>% left_join(clasifi_prov)

#MDist = bind_rows(MDist,temp)
MDist = MDist %>% group_by(ejercicio,cod_region,descr_region,cod_provincia,descr_provincia,
                           tipo_activ,codigo_industria,descr_codigo_industria,clave) %>%
  summarise(valor = sum(valor,na.rm = T))

MDist = full_join(MDist,VAB_N,by=c("ejercicio","codigo_industria"))
MDist = MDist %>% mutate(prioridad = ifelse(clave=="produccion",1,0))

MDist = MDist %>% group_by(ejercicio,codigo_industria,clave) %>%
  mutate(valor_ag = round(sum(valor,na.rm = T)),cobert=round(valor_ag/(P.1_Corriente*1000),2))
# Si el valor de producción de la información territorial de base cubre a menos el 70%
# del valor real de la producción a nivel nacional, se toma como fuente de distribución
# en cada sub-industria de la educación.

MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & cobert>=0.70,1,0))
#MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="01.01.01",0,prioridad))

# Se trabaja con el dato de empleo en desarrollo infantil público, al no existir desagregación provincial
MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="02.02.01",0,prioridad))
MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="02.01.01",0,prioridad))
MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="02.01.02",0,prioridad))

# si bien existe buena cobertura en valores de producción los ITTs privados, existen muchas provincias
# sin datos (13) por lo cual se trabajará con datos de empleo desde la base del directorio de empresas
MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="05.01.01",0,prioridad))
MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="05.02.01",0,prioridad))

# En el sector privado y público luego de observar varios casos con información ausente a nivel provincial
# cuando se usan datos de producción
MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="03.01.01",0,prioridad))
#MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="3.02.01",0,prioridad))
MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="04.01.01",0,prioridad))
MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="04.01.02",0,prioridad))
MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="05.02.01",0,prioridad))
MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="07.01.01",0,prioridad))
MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="07.02.01",0,prioridad))

# Universidades se trabaja con el dato de producción y con ajuste adicional con alumnos en casos de
# universidades con sedes en otras provincias
#MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="06.02.01",1,prioridad))
MDist = MDist %>% mutate(prioridad = ifelse(prioridad==1 & codigo_industria=="06.01.01",0,prioridad))

#identifico las sub-industria que no tienen clave de distribución (producción) válida, estos casos
# se redistribuyen con los datos de alumnos docentes
temp = MDist %>% filter( prioridad==0 )
# Filtro matriz de claves basadas en la producción donde prioridad == 1
MDist = MDist %>% filter(prioridad==1)

```

```
temp = temp %>% filter(!(codigo_industria %in% unique(MDist$codigo_industria)))

unique(temp$codigo_industria)
head(temp)

## Identificar matriz de claves basadas en los docentes como prioridad == 2
## Identificar matriz de claves basadas en los docentes como prioridad == 2
# Se toma como siguiente variable clave de prioridad al dato de docentes, no obstante debido
# a que se trata de registros nuevos, se trabaja con datos consolidados entre el 2020 y 2021
# y ese valor se aplicará en la distribución de cada año.
#temp = temp %>%
# mutate(clave = ifelse(codigo_industria=="05.01.01" & clave=="beneficiarios","docentes1",
#                       ifelse(codigo_industria=="05.01.01" & clave=="docentes","beneficiarios",clave)))

temp = temp %>%
  mutate(clave = ifelse(codigo_industria=="01.01.01" & clave=="beneficiarios","docentes1",
                       ifelse(codigo_industria=="01.01.01" & clave=="docentes","beneficiarios",clave)))

tempx = temp
xxx = MDist
#if (nrow(temp) > 0) {
# for (i in 2019:2021) {
#   temp = tempx %>% filter(clave=="beneficiarios" & ejercicio %in% c(i,i-1)) %>% group_by(codigo_industria,cod_provincia)
#   %>%
#   #   mutate(valor=sum(valor,na.rm = T))
#   temp$prioridad = 2
#   #if (i > 2019){
#   # temp = temp %>% filter(ejercicio == i)
#   #}
#   #MDist = bind_rows(MDist,temp )
#   #}

#}

if (nrow(temp) > 0) {
  for (i in 2019:2022) {
    if (i < 2020){
      temp = tempx %>% filter(clave=="beneficiarios" & ejercicio %in% c(i,i-1)) %>% group_by(codigo_industria,cod_provincia)
    } %>%
      mutate(valor=sum(valor,na.rm = T))
      temp$prioridad = 2
    }
    else {
      temp = tempx %>% filter(clave=="beneficiarios" & ejercicio == i)
      temp$prioridad = 2
    }
    MDist = bind_rows(MDist,temp )
  }
}

#tempx = temp

#if (nrow(temp) > 0) {
# for (i in 2018:2021) {
#   temp = tempx %>% filter(clave=="beneficiarios" & ejercicio == i) %>% group_by(codigo_industria,cod_provincia) %>%
#   #   mutate(valor=sum(valor,na.rm = T))
#   temp$prioridad = 2
#   #MDist = bind_rows(MDist,temp )
#   #}

#}

MDist$cobert = NULL
MDist$valor_ag = NULL

MDist$cobert = NULL
MDist$valor_ag = NULL
```

Una vez que se tienen las matrices de distribución basadas en datos monetarios y físicos, se generan los agregados a nivel territorial y se estructura la base de datos del VAB provincial. Además, se genera el mapa provincial del valor de la producción de los servicios de educación a nivel provincial, el cual también incluye información de la población en cada provincia.

```
### 4 Calcular VAB provincial

temp = NULL

for (h in c("P.1_Corriente","P.2_Corriente","B.1b_Corriente")){
  VAB_Prov = MDist %>%
    group_by(ejercicio,prioridad,codigo_industria,descr_codigo_industria) %>%
    nest() %>%
    mutate(mod_obj = map(data, ~ participa[, "cod_provincia", "valor",
      h, nuevo_df = 2, dec = 10L, id = "ver"])) %>%
    select(ejercicio,prioridad,codigo_industria,descr_codigo_industria,mod_obj) %>%
    unnest(mod_obj)
  VAB_Prov = VAB_Prov %>% left_join(clasifi_prov)
  VAB_Prov = clasifi %>% group_by(codigo_industria=cod_industria_N3, tipo_activ, sector) %>%
    summarise(n()) %>% filter(codigo_industria!="99999") %>%
    select(codigo_industria, tipo_activ, sector) %>% right_join(VAB_Prov)
  val_dist = paste0(h, "_dist")
  VAB_Prov = VAB_Prov %>% mutate(indicador = h) %>%
    rename(agregado = h, agregado_dist = val_dist)
  temp = bind_rows(VAB_Prov, temp)
}
VAB_Prov = temp %>% filter(tipo_activ=="Característico")
View(VAB_Prov)

dir_terri = paste0(direccion0, "/Prov")
dir.create(dir_terri)
setwd(dir_terri)

VAB_Prov = clasifi %>%
  group_by(codigo_industria=cod_industria_N3, cod_industria_N2, descr_industria_N2,
    cod_industria_N1, descr_industria_N1) %>%
  summarise(n()) %>% filter(codigo_industria!="99999") %>% right_join(VAB_Prov, multiple = "all")

setwd(direccion1)
VAB_Prov2019 = read.xlsx("Prov/5_VAB_Prov.xlsx")
names(VAB_Prov2019)

VAB_Prov2019 = VAB_Prov2019 %>%
  rename(cod_industria_N2=codigo_industria_agg, descr_industria_N2=descr_codigo_industria_agg,
    cod_industria_N1=codigo_industria_agg_2, descr_industria_N1= descr_codigo_industria_agg_2, tipo_activ=producto)

VAB_Prov2019 = VAB_Prov2019 %>% mutate(cod_industria_N1 = ifelse(cod_industria_N1 == "05", "05-06", cod_industria_N1))

VAB_Prov2019 = VAB_Prov2019[VAB_Prov2019$ejercicio<=2019,]
VAB_Prov = VAB_Prov[VAB_Prov$ejercicio>2019,]
VAB_Prov = VAB_Prov %>% bind_rows(VAB_Prov2019)

setwd(dir_terri)
write.xlsx(MDist22, "Mdist2022.xlsx")
write.xlsx(MDist, "2_ind_provinciales22.xlsx", overwrite=T)
write.xlsx(VAB_Prov, "5_VAB_Prov22.xlsx", overwrite = T)
write.xlsx(VAB_N, "4_VAB_N22.xlsx", overwrite = T)

gc()
```

El archivo de sintaxis (script completo) que automatiza la construcción de la base de datos del VAB provincial y genera los tabulados se encuentra ubicado en la siguiente ubicación de la carpeta compartida de la unidad:

R:\CGTPE\DECON\AS\CS\_MPE\_2023\CSE\_2022\5\_Proc\5.4\_Deriv\_variab\5.4.2\_Ca  
l\_variab\_deriv\1\_Scripts\_Result

**Nombre del archivo:** 5\_BS\_VAB\_TERRI22

### 3.4. Estructura de la base de datos procesada del VAB provincial

Una vez que se ejecuta la sintaxis se obtiene una base de datos con una estructura de 17 variables, como se muestra en la siguiente tabla:

**Tabla 1:** Descripción de variables de la base datos del VAB provincial.

Nro.	Variables	Descripción de variables
1	Ejercicio	Ejercicio o año de la información en la base de datos.
2	Sector	Sector al que pertenece la institución (público o privado).
3	Producto	Tipo de producto de las CSE (característico o conexo).
4	Codigo_industria_agg	Código de industria de las CSE a nivel agregado
5	Descr_codigo_industria_agg	Descripción del código de industria de las CSE a nivel agregado.
6	Codigo_industria	Código de la industria de las CSE.
7	Descr_codigo_industria	Descripción del código de la industria de las CSE.
8	Cod_region	Código de las regiones naturales del Ecuador.
9	Descr_region	Descripción del código de las regiones naturales del Ecuador.
10	Cod_provincia	Código de la provincia según división política administrativa.
11	Descr_provincia	Descripción del código de la provincia según división política administrativa.
12	Indicador	Tipo de agregado económico (P.1 = producción, P.2 = consumo intermedio y B.1b=VAB).
13	Agregado	Valor del agregado económico a nivel nacional.
14	Prioridad	Indica el tipo de variable usada como clave de distribución (1: variable monetaria y 2: variable física).
15	Valor	Valor absoluto de la clave de distribución.
16	Clave_dist	Es el porcentaje que asigna la clave de distribución para cada provincia.
17	Agregado_dist	Valor del agregado económico a nivel provincial.

La base de datos finalizada del VAB provincial se encuentra ubicada en la siguiente ubicación de la carpeta compartida de la unidad:

R:\CGTPE\DECON\AS\CS\_MPE\_2023\CSE\_2022\5\_Proc\5.7\_Finali\_archiv\_dat\5.7.  
2\_Compil\_prod\_ant\5\_Bases\_result

**Nombre del archivo:** "2\_ind\_provinciales22"

## 4. Conclusiones

- Los indicadores territoriales enriquecen la información de las CSE pues presentan información relevante de agregados económicos (producción, consumo intermedio y valor agregado bruto) por cada provincia.

- La construcción del tabulado se la realiza en el software estadístico R ya que promueve la investigación reproducible y sus scripts son oportunos para la optimización de tiempos y recursos.
- Dado que el software estadístico R usa una secuencia de comandos y se puede acceder al historial de las acciones realizadas dentro de la construcción de cada una de las variables, hace más fácil la detección y solución de errores dado el caso de presentarse dentro de la fase de estructuración y construcción del tabulado.

FIRMAS DE APROBACIÓN	
ELABORADO POR:	REVISADO POR:
<p>Miembro de Equipo Unidad de Gestión de Análisis de Síntesis</p> <p><b>Nombre:</b> Mireya Barrera</p>	<p>Jefe de Unidad de Gestión de Análisis de Síntesis</p> <p><b>Nombre:</b> Henry Valdiviezo</p>

