



Informe de procesamiento de la base de datos de síntesis: **VAB territorial**

..... Cuentas Satélite de Educación
..... (CSE) 2024

DICIEMBRE · 2025

1 NOMBRE DE LA BASE DE DATOS

Valor Agregado Bruto territorial (VAB territorial)

2 TIPO DE BASE DE DATOS

Base de datos de síntesis

3 SECTOR INSTITUCIONAL DENTRO DE LA COBERTURA DE LA BASE DE DATOS: N2

S11 Sociedades no financieras-SNF

- S11.01 Sociedades no financieras característicos

13 Gobierno general

- S13.01 Gobierno central
- S13.02 Gobierno local

S14 Hogares

- S14.01 Hogares productores
- S14.02 Hogares consumidores

S15 Instituciones sin fines de lucro que sirven a los hogares – ISFLSH

- S15.01 Instituciones sin fines de lucro que sirven a los hogares

4 OBJETIVO DE LA BASE DE DATOS

Determinar la información económica desagregada a nivel territorial para estimar la producción, consumo intermedio y valor agregado bruto del sector educativo a nivel provincial, asegurando la consistencia con los agregados nacionales y el resguardo de la confidencialidad estadística.

5 FUENTE DE DATOS PRINCIPAL

- Matriz de distribución de alumnos Mineduc
- Matriz territorial
- Base de datos unificada 2022-2024
- Programas Mineduc
- Valor agregado bruto (VAB24)
- VAB provincial 2023

(Ver anexo1 ruta archivo de insumos)

6 FUENTE DE DATOS SECUNDARIA/VALIDACIÓN

- VAB provincial 2023

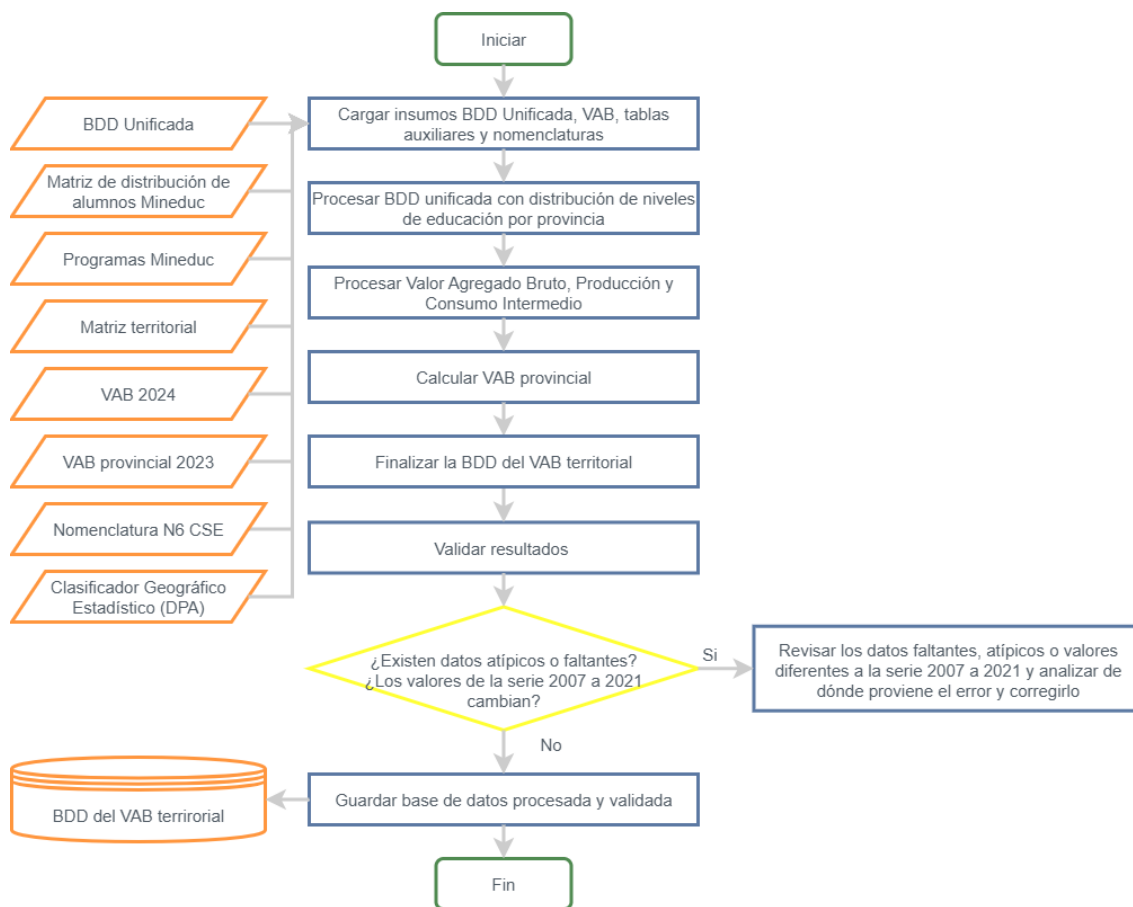
(Ver anexo1 ruta archivo de insumos)

7 NOMENCLATURAS A USAR

- Nomenclatura CSE: 1.12 correspondencia2024
- Clasificador Geográfico Estadístico (División Político Administrativa)

(Ver anexo1 ruta archivo de insumos)

8 FLUJO DEL PROCESO DE CONSTRUCCIÓN DE LA BASE DE DATOS



9 COBERTURA INSTITUCIONAL/INDUSTRIA/PRODUCTO DE LA BASE DE DATOS:

codigo_industria	descr_codigo_industria
01.01.01	Actividades de regulación y administración de servicios de enseñanza
02.01.01	Actividades de servicios de enseñanza de desarrollo infantil privado
02.01.02	Actividades de servicios de enseñanza preprimaria privado
02.02.01	Actividades de servicios de enseñanza de desarrollo infantil público
02.02.02	Actividades de servicios de enseñanza preprimaria público
03.01.01	Actividades de servicios de enseñanza primaria privado
03.02.01	Actividades de servicios de enseñanza primaria público
04.01.01	Actividades de servicios de enseñanza secundaria baja privado
04.01.02	Actividades de servicios de enseñanza secundaria alta privado
04.02.01	Actividades de servicios de enseñanza secundaria baja público
04.02.02	Actividades de servicios de enseñanza secundaria alta público
05.01.01	Actividades de servicios de enseñanza superior de ciclo corto privado
05.02.01	Actividades de servicios de enseñanza superior de ciclo corto público
06.01.01	Actividades de servicios de enseñanza superior privado
06.02.01	Actividades de servicios de enseñanza superior público
07.01.01	Actividades de servicios de otros tipos de enseñanza y de apoyo a la enseñanza privado
07.02.01	Actividades de servicios de otros tipos de enseñanza y de apoyo a la enseñanza público

10 CODIGOS DE CUENTAS NACIONALES QUE CONFORMAN LA BASE DE DATOS

Tipo	cod_CN	desc_cod_CN
Ingreso	B.1b	Valor agregado bruto (corriente)
Ingreso	P.1	Producción (corriente)
Gasto	P.2C	Consumo intermedio (corriente)

11 VARIABLES QUE CONFORMAN LA BASE DE DATOS

Variable	Descripción de variable
cod_industria	Código a nivel 3 de industrias de las CSE
cod_industria_N2	Código a nivel 2 de industrias de las CSE
descr_industria_N2	Descripción del código a nivel 2 de las industrias de las CSE
cod_industria_N1	Código a nivel 1 de industrias de las CSE
descr_industria_N1	Descripción del código a nivel 1 de las industrias de las CSE
n()	
tipo_activ	Tipo de actividad: característica o conexa
sector	Sector al que pertenece: público o privado
ejercicio	Año al que corresponde la información
prioridad	Prioridad de uso de los coeficientes de la matriz territorial (producción, alumnos, docentes)
descr_codigo_industria	Descripción del código a nivel 3 de las industrias de las CSE

cod_provincia	Código de la provincia
valor	Valor presupuestario devengado
agregado	Valor agregado
clave_dist	Coeficiente usado para la distribución territorial
agregado_dist	Valor del agregado distrital
descr_provincia	Descripción del código de la provincia
cod_region	Código de la región
descr_region	Descripción del código del cantón: 01 Sierra, 02: Costa, 03: Oriente; 04 Insular
indicador	Tipo de indicador: B.1b, P.1 y P.2

12 LIMITACIONES TÉCNICAS/OBSERVACIONES

Ninguna.

13 PERIODICIDAD DE LA BASE

Anual

14 DISPONIBILIDAD DE LOS DATOS

2007-2024

15 NIVEL DE DESAGREGACIÓN

Geográfico: Nacional

Institucional: Unidades institucionales S11, S13, S14 y S15.

Industria/producto: Industria a nivel N3

16 FECHA DE PROCESAMIENTO

Agosto 2025

17 FECHA DE LA ÚLTIMA ACTUALIZACIÓN

Septiembre 2025

18 NOVEDADES EN EL PROCESAMIENTO DE LA INFORMACIÓN

El procesamiento de la información inició en agosto de 2025, usando para este fin insumos duplicados con el objetivo de verificar que no existan errores en las herramientas utilizadas para el procesamiento de la información. Una vez que se verificó que el script para el procesamiento no tenía errores, se fueron incorporando uno a uno los insumos actualizados. Esto permitió identificar insumos en los que existió un cambio en el formato, impidiendo que los resultados se generaran sin novedad.

En total, se realizó la corrida de los resultados en 10 ocasiones, y, posterior a cada una de ellas, se llevó a cabo la validación de los resultados; para ello, se verificó

que, de los años 2018 a 2021, los resultados fueran los mismos que los generados en la base de datos el año anterior. Adicionalmente, se analizó la serie temporal de las principales variables con el fin de identificar la consistencia de los datos en relación con la serie temporal.

Finalmente, después de la incorporación de todos los insumos actualizados y de la validación de la consistencia de la información, finalizó el procesamiento con la corrida de los Resultados 10.

19 RUTA DE LA SINTAXIS PARA LA CONSTRUCCION DE LA BDD DE SÍNTESIS

R:\CGTPE\DECON\AS\CS_MPE_2025\CSE_2022_24\5_Proc\5.6_Deriv_nuev_variab
\5.6.2_Deriv_unidad\1_Proc

20 NOMBRE ARCHIVO SINTAXIS

2_BS_VAB_TERRI_CSE24

21 RUTA DE LA BASE DE SÍNTESIS

R:\CGTPE\DECON\AS\CS_MPE_2025\CSE_2022_24\5_Proc\5.8_Final_archiv_dato\
5.8.1_Reun_result_prelim

22 NOMBRE DE LA BASE DE SÍNTESIS

5_VAB_Prov24

23 PROCESADO POR

Nikole Pepinós

24 REVISADO POR

Henry Valdiviezo

ANEXOS

ANEXO 1: RUTA DE ARCHIVO DE INSUMOS

Nombre del insumo	Nombre archivo	Ruta archivo	Fuente
Matriz de distribución de alumnos Mineduc	CSE_MatDist_2007-2024_alumnos	R:\CGTPE\DECON\AS\CS_MPE_2025\CSE_2022_24\5_Proc\5.5_Editar_imput\5.5.2_Imput_datos\2_TABLAS_AUXIL\Matriz_distr_MINEDUC	Instituto Nacional de Estadística y Censos (INEC)
Matriz territorial	Matriz_Territ_CSE_2024	R:\CGTPE\DECON\AS\CS_MPE_2025\CSE_2022_24\5_Proc\5.5_Editar_imput\5.5.2_Imput_datos\2_TABLAS_AUXIL\Matriz_territorial	Instituto Nacional de Estadística y Censos (INEC)
Base de datos unificada	BUCSE_22-24	R:\CGTPE\DECON\AS\CS_MPE_2025\CSE_2022_24\5_Proc\5.5_Editar_imput\5.5.2_Imput_datos\3_BDD_UNIFICADA	Instituto Nacional de Estadística y Censos (INEC)
Programas Mineduc	Prog_mineduc_2024	R:\CGTPE\DECON\AS\CS_MPE_2025\CSE_2022_24\5_Proc\5.5_Editar_imput\5.5.2_Imput_datos\2_TABLAS_AUXIL\Programas_MINEDUC	Instituto Nacional de Estadística y Censos (INEC)
VAB territorial	5_VAB_Prov23	R:\CGTPE\DECON\AS\CS_MPE_2025\CSE_2022_24\5_Proc\5.6_Deriv_nuev_variab\5.6.2_Deriv_unidad\0_Arch_trab	Instituto Nacional de Estadística y Censos (INEC)
VAB 2024	VAB24	R:\CGTPE\DECON\AS\CS_MPE_2025\CSE_2022_24\5_Proc\5.8_Final_archiv_dato\5.8.1_Reun_result_prelim	Instituto Nacional de Estadística y Censos (INEC)
Nomenclatura CSE	1_Deli_CSE_2024 (1.12 correspondencia2024)	R:\CGTPE\DECON\AS\CS_MPE_2025\CSE_2022_24\2_Dis\2.2_Dis_variab\2.2.3_Ident_clasificad_nom	Instituto Nacional de Estadística y Censos (INEC)
Clasificador Geográfico Estadístico (División Político Administrativa)	2_cge_2024	R:\CGTPE\DECON\AS\CS_MPE_2025\CSE_2022_24\2_Dis\2.2_Dis_variab\2.2.3_Ident_clasificad_nom	Instituto Nacional de Estadística y Censos (INEC)

ANEXO 2: SINTAXIS

```
#
# Unidad de Gestión de Análisis de Síntesis
# Cuentas Satélite de Educación
# Ejercicio: 2022-2024
# Fecha: 01/07/2025
# ---- FASE 5. PROCESAMIENTO ----
# VAB TERRITORIAL DE LAS CSE
#
# Generación de indicadores de producción, consumo intermedio y VAB territorial

#
# 0. Cargar librerías de trabajo y definir directorios ----
#

## a. Llamar paquetes ----
#
```



```
library(tidyverse)
library(openxlsx)
library(reshape2)
library(mice)
library(rpivotTable)
library(cartography)
library(sf)
library(mapsf)

## b. Definir directorio ----
#-----
# area_trabajo<-
#"R:/CGTPE/DECON/AS/CS_MPE_2025/CSE_2022_24/5_Proc/5.6_Deriv_nuev_variab/5.6.2_Deriv_unidad/1_Proc"
# insumos<-
#"R:/CGTPE/DECON/AS/CS_MPE_2025/CSE_2022_24/5_Proc/5.6_Deriv_nuev_variab/5.6.2_Deriv_unidad/0_Arch
_trab"
#
# direccion0 <- "D:/Respaldos/5_DECON/2_CSE/2024/Procesamiento/Res_24/RESULTADOS_10"
# if (dir.exists(direccion0)==FALSE){
# dir.create(direccion0)
# }

hora_inicio = Sys.time()

ini<-2018
ini_territ = 2018
fin<-2024

#-----
# 1. Procesar BD unificada con distribución de niveles de educación por provincia ----
#-----
# 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

## 1.1. Preparar funciones para consolidar datos, distribuir y ajustar valores ----
#-----
setwd(area_trabajo)
source("Fun_CS_v02.r")

setwd(insumos)
archivo_matriz = "CSE_MatDist_2007-2024_alumnos.xlsx"
mt_coef_territ = "Matriz_Territ_CSE_2024.xlsx"
clasifi_prov = read.xlsx("1_Deli_CSE_2024.xlsx",sheet ="1.14 Prov_reg")

## 1.2. Construir y homologar matrices de distribución ----
#-----

# Cargar Matriz de distribución a nivel provincial (hoja: Distritos)
setwd(insumos)
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)

# Cargar Base de Datos Unificada
archivo = "BUCSE_22-24.xlsx"
bdp<- read.xlsx(archivo)
bdp$ejercicio<- as.numeric(bdp$ejercicio)

# Recodificar a nacional datos incompletos en gobiernos locales y ciclo corto
bdp<- bdp %>%
mutate(cod_provincia=ifelse(codigo_N6 == "$13.01.05.01.02.01","00",cod_provincia),
cod_provincia=ifelse(substr(codigo_N6,1,6) == "$13.02","00",cod_provincia),
cod_provincia=ifelse(codigo_N6 == "$13.01.01.02.01.05","00",cod_provincia))
sum(bdp$devengado,na.rm = T)
```

```
table(is.na(bdp$cod_provincia))
bdp$cod_provincia <- as.character(bdp$cod_provincia)
bdp = bdp %>%
  mutate(cod_provincia = ifelse(is.na(cod_provincia), "99",cod_provincia),
  cod_provincia = ifelse(nchar(cod_provincia) == 1, paste0("0",cod_provincia),as.character(cod_provincia)))

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

nrow(bdp)
#pivotTable(bdp)

#### S14 ----
#-----
bdp1 = bdp %>%
  filter(substr(codigo_N6,1,6)=="S14.01" & ejercicio>=ini)

mtd1 = read.xlsx(mt_coef_territ, sheet = "S14")
mtd1 = mtd1 %>%
  select(ejercicio,codigo_N6,cod_provincia,descr_provincia,valor) %>%
  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,tipos,codigo_N1,codigo_N6,descr_codigo_N6,cod_CN,devengado)

sum(bdp1$devengado, na.rm = T)
sum(mtd1$valor, na.rm = T)

bdp1$ejercicio <- as.double(bdp1$ejercicio)
mtd1$ejercicio <- as.double(mtd1$ejercicio)

bdp1 = bdp1 %>%
  left_join(mtd1, multiple = "all")
sum(bdp1$devengado, na.rm = T)
sum(bdp1$valor, na.rm = T)

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

#### 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_a"),
  read.xlsx(mt_coef_territ,sheet = "C6.02.01"))

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) %>%
  mutate(ejercicio = as.numeric(mtd1$ejercicio))

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

sum(bdp1$devengado)

bdp1$ejercicio <- as.double(bdp1$ejercicio)
mtd1$ejercicio <- as.double(mtd1$ejercicio)

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

bdp1 = bdp1 %>%
  group_by(ejercicio,id_registro,codigo_N1,codigo_N6,descr_codigo_N6,tipos,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,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_registro,tipos,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]) %>%
  mutate(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)
#pivotTable(bdp)
```

```
#### 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_b")

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

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$ejercicio <- as.double(bdp1$ejercicio)
mtd1$ejercicio <- as.double(mtd1$ejercicio)

bdp1 = bdp1 %>%
  left_join(mtd1)
bdp1$beneficiarios <- NULL

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","docentes","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") %>%
  mutate(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)
mtd3$id_registro = gsub("ZONAL", "ZONAL ", mtd3$id_registro)

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,tipo,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,tipo,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) %>%
  mutate(fuente = "Distribuido desde BBD unificada zonales")

bdp2 = bdp2 %>%
  select(-devengado1) %>%
  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) %>%
  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$codigo_N6 = "S13.01.01.02.01.02"
bdp2$fuente = "Distribuido desde BBD unificada zonales"
#bdp2$cod_provincia = bdp2$cod_provincia %>% replace_na(99)
sum(bdp2$devengado,na.rm = T)
table(is.na(bdp2$cod_provincia))

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

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

#### S13, S11 y S15 ----
#-----
mtd_distr_prov = read.xlsx("Distrito_prov.xlsx")

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

setwd(direccion0)
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)

#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)
bdp$ejercicio <- as.numeric(bdp$ejercicio)

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

aggregate(devengado~ ejercicio, bdp, sum)

# Incluir matriz de programas educativos
setwd(insumos)
bdp1 = read.xlsx("Prog_mineduc_2024.xlsx", sheet = 1, startRow = 1)
bdp1 = pivot_longer(bdp1, S13.01.02.02.01.01:S13.01.04.02.03.01, names_to = "codigo_N6",
  values_to = "devengado")
bdp1 = bdp1 %>%
  select(ejercicio, descr_programa=programa, cod_provincia, descr_provincia, codigo_N6, devengado) %>%
```

```
mutate(codigo_N1 = substr(codigo_N6,1,3),cod_CN = "P.2", fuente ="Programas educativos MINEDUC", tipo =
"2",
ejercicio = as.numeric(bdp1$ejercicio))

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

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

setwd(dir_terri)
write.csv(bdp,"1_bdg_distr_prov_niveles_24.csv")
rpivotTable(bdp)

#
# 2. Procesar VAB, Prod, CI Nacional ----
#

## 2.1. Cargar insumos ----
#
# Base del VAB
setwd(direccion0)
VAB <- read.xlsx("VAB24.xlsx")
#rpivotTable(bdp)
aggregate(Corriente~ ejercicio, VAB, sum)
aggregate(Constante~ ejercicio, VAB, sum)
#Archivo de nomenclaturas

setwd(insumos)
clasif <- read.xlsx("1_Deli_CSE_2024.xlsx",sheet = "1.12 correspondencia",startRow=13,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))

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)
rpivotTable(VAB_N)

## 2.2. 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"))
```

```
#aggregate(produccion ~ ejercicio, bdp2, sum )
#rpivotTable(bdp2)

## 2.3. Generar matrices distribuidas ----
#-----

#### Datos de docentes y estudiantes a nivel territorial ----
#-----
setwd(insumos)

wb = mt_coef_territ
nombres = getSheetNames(wb)
nombres
vhojas = nombres[c(3:21)]

MDist = consolidaHojas(wb,vhojas,finicio=1)
unique(MDist$id_hoja)
MDist = MDist %>%
  filter(id_hoja!="C6.01.01_b")

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)) %>%
  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)

MDist24 = 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")
  rpivotTable(MDist24)

#### 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)

Alumnos_prov$cod_industria_N2 <- ifelse(Alumnos_prov$codigo_industria == "04.01.01",
  Alumnos_prov$cod_industria_N2 == "04.01", Alumnos_prov$cod_industria_N2)
table(Alumnos_prov$codigo_industria)

setwd(direccion0)
#rpivotTable(Alumnos_prov)
write.xlsx(Alumnos_prov,"Alumns_prov_res24.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)
pivotTable(temp)

#### Matriz a nivel de industria con valores de producción y otras variables para distribución ----
#
temp = temp %>%
right_join(temp1)
MDist$ejercicio = as.numeric(MDist$ejercicio)

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")))

#### Imputar 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))

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

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

# Ejecutar una imputación múltiple:
# 1) Generar imputaciones
# 2) Estimar coeficientes puntuales
# 3) Agrupar los estimadores
```

```
# 4) Predecir valores
# Generar 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 = 2022)
summary(temp1)
densityplot(temp1, subset = cod_provincia!="17" & cod_provincia!="09" & cod_provincia!="13")

# Excluir 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 múltiples 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
# 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 2019:2024) {
    if (i < 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$prioridad = 2
    }
    else {
      temp = tempx %>%
        filter(clave=="beneficiarios" & ejercicio == i)
      temp$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

#}
#pivotTable(MDist)

# ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo
# 3. Calcular VAB provincial ----
# ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

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 = clasif %>% 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)

VAB_Prov = clasif %>%
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(insumos)
VAB_Prov2020 = read.xlsx("5_VAB_Prov23.xlsx")
names(VAB_Prov2020)

VAB_Prov2020 = VAB_Prov2020 %>%
mutate(cod_industria_N1 = ifelse(cod_industria_N1 == "05", "05-06", cod_industria_N1))

VAB_Prov2020 = VAB_Prov2020[VAB_Prov2020$ejercicio<=2021,]
VAB_Prov = VAB_Prov[VAB_Prov$ejercicio>2021,]

unique(VAB_Prov$ejercicio)
unique(VAB_Prov2020$ejercicio)

VAB_Prov = VAB_Prov %>% bind_rows(VAB_Prov2020)

setwd(dir_terri)
write.xlsx(MDist24, "Mdist2024.xlsx")
write.xlsx(MDist, "2_ind_provinciales24.xlsx", overwrite=T)
write.xlsx(VAB_Prov, "5_VAB_Prov24.xlsx", overwrite = T)
write.xlsx(VAB_N, "4_VAB_N24.xlsx", overwrite = T)

gc()

print("Proceso bases territorial de las CSS 2018-2024 finalizado")
setwd(area_trabajo)

##### FIN #####

# Copiar los resultados a la carpeta compartida ----
# ::::::::::::::::::::::::::::::::::::::::::::

setwd(direccion0)
archivos = dir(recursive = T)
temp = dir(pattern = "*.png", recursive = T)
archivos = archivos[!(archivos %in% temp)]
ruta = substr(direccion0, nchar(direccion0)-1, nchar(direccion0))
```

```

ruta =
paste0("R:/CGTPE/DECON/AS/CS_MPE_2025/CSE_2022_24/5_Proc/5.8_Final_archiv_dato/5.8.1_Reun_result_pr
elim/RESULTADOS_",ruta)

dir.create(ruta)

file.copy(archivos, ruta, overwrite = T)
setwd(areatrabajo)

#.....#
#### FIN ####
#.....#

```

ELABORADO POR:	REVISADO y APROBADO POR:
<p>Miembro de Equipo de la Gestión de Análisis de Síntesis</p> <p>Nombre: Nikole Pepinós</p>	<p>Responsable de la Gestión de Análisis de Síntesis</p> <p>Nombre: Henry Valdiviezo</p>



Buenas cifras,
mejores vidas



@InecEcuador



@ecuadorencifras



@ecuadorencifras



INECEcuador