Islas Canarias
Del 15 al 19 de noviembre de 2021

Estadística Castilla y León
Junta de Castilla y León
Consejería de Economía y Hacienda
Dirección General de Presupuestos y Estadística

R y shiny para la difusión de datos e indicadores

Jesús M. Rodríguez Rodríguez
D.G. de Presupuestos y Estadística de la Junta de Castilla y león
jesusmaria.rodriguez@jcyl.es

Introducción

Desde hace muchos años se está utilizando el Sistema de Información Estadística (SIE) de la Junta de Castilla y León como una herramienta tanto para el trabajo diario como para la difusión de datos a distintos niveles (municipal, provincial, Comunidad) y de distintos temas.

El interfaz de dicho sistema es a veces algo complejo y los resultados se obtienen siempre en un fichero con formato csv.

En la mayoría de las ocasiones no es necesario utilizar todas las posibilidades que ofrece el sistema y además los datos nos los piden en otros formatos, generalmente como una hoja electrónica.

Para agilizar la consultas más habituales y lograr una mayor rapidez a la hora de presentar los datos se han desarrollado distintas aplicaciones en R [1], para las que en muchos casos se utiliza shiny [2] como interfaz, lo cual permite una forma más sencilla de hacer las consultas y obtener los resultados en distintos formatos.

Entre estas aplicaciones, en esta ponencia se va a hablar de tres de ellas: Consulta de datos de comercio exterior, consulta de indicadores municipales y la difusión de indicadores de la Agenda 2030.

Todas ellas se han puesto en un servidor de shiny y por lo tanto pueden ser utilizadas por cualquiera con acceso a dicho servidor, sin necesidad de tener instalado R. Las dos primeras se desarrollaron hace tiempo, pero recientemente se han desarrollado los interfaces con shiny.

La de los indicadores para la Agenda 2030 se ha desarrollado cuando se han ido obteniendo resultados del grupo de trabajo de las comunidades sobre este tema y de momento es la única a la que se tiene acceso desde el exterior.

En cada una de las tres se obtienen los datos de base de distintas formas: con consulta a las bases de datos, mediante la utilización del API del SIE o con datos preparados expresamente para la aplicación.

Objetivos

Los objetivos principales que se han intentado lograr con el desarrollo de estas aplicaciones son:

- Mayor rapidez a la hora de responder a peticiones que se repiten periódicamente.
- Que las aplicaciones sean lo suficientemente versátiles para poder adaptarlas a distintos tipos de peticiones.
- Facilidad de uso para que puedan trabajar con ellas técnicos que no necesariamente trabajen con R.
- El uso de estas aplicaciones por parte de los ciudadanos a través de internet.

Los cuatro primeros objetivos se han logrado en todos los casos que se comentan en este trabajo, pero el último solamente se ha alcanzado en una de las tres aplicaciones presentadas.

Metodología

Para la implementación en R de las distintas aplicaciones se han desarrollado dos paquetes, uno interno que facilita el acceso a las bases de datos de Oracle, DGPYEN, y otro que en su día se subió al CRAN, tablaxlsx [3], para facilitar la creación de ficheros en excel con un formato específico.

Siempre que ha sido posible se han utilizado las distintas funciones que se tenían ya desarrolladas para los trabajos que se realizan habitualmente.

Las aplicaciones se han colocado en un servidor con CENTOS en el que se ha instalado R y Shiny Server.

En general las aplicaciones constan de las siguientes partes claramente diferenciadas:

- El interfaz, que en las aplicaciones presentadas en este trabajo se ha desarrollado en “shiny”.
- Funciones para la consulta de los datos en las fuentes disponibles: Oracle, SIE, Microdatos, Internet, …
- Funciones para los cálculos intermedios necesarios. Esta parte a veces va junto a la anterior, sobre todo en los caos en los que la consulta de los datos es sencilla.
- Funciones para la escritura de los resultados en los formatos de interés: Excel, Word, …

Resultados

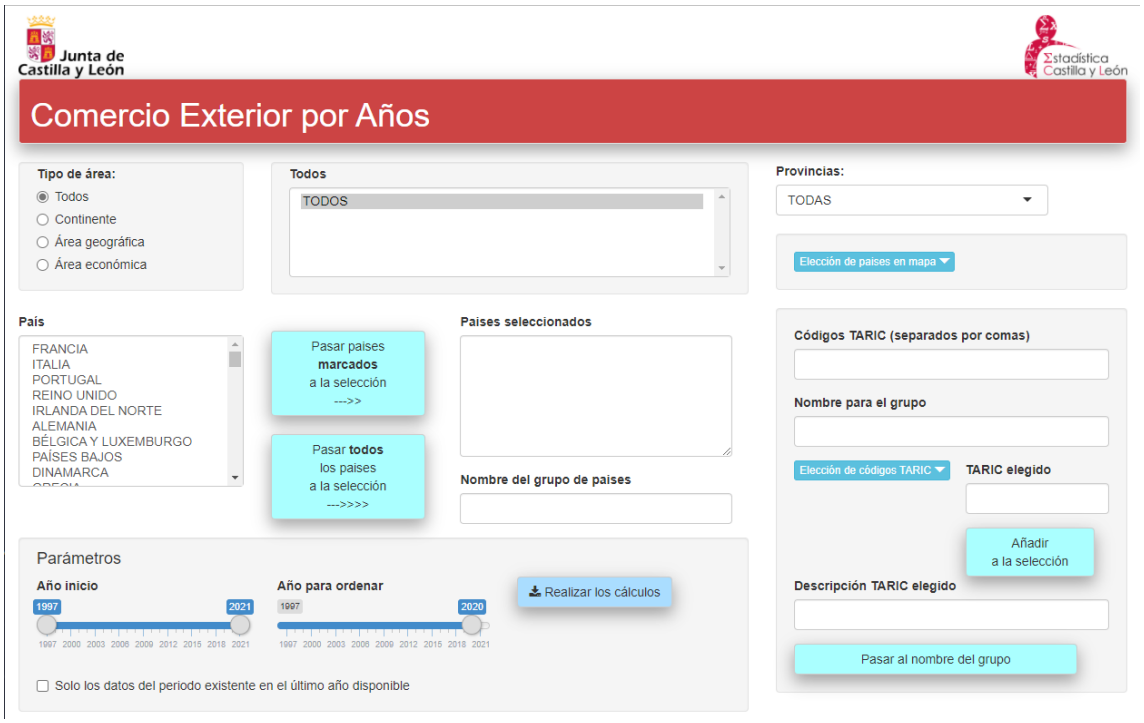
A continuación se describen las tres aplicaciones desarrolladas.

Comercio exterior

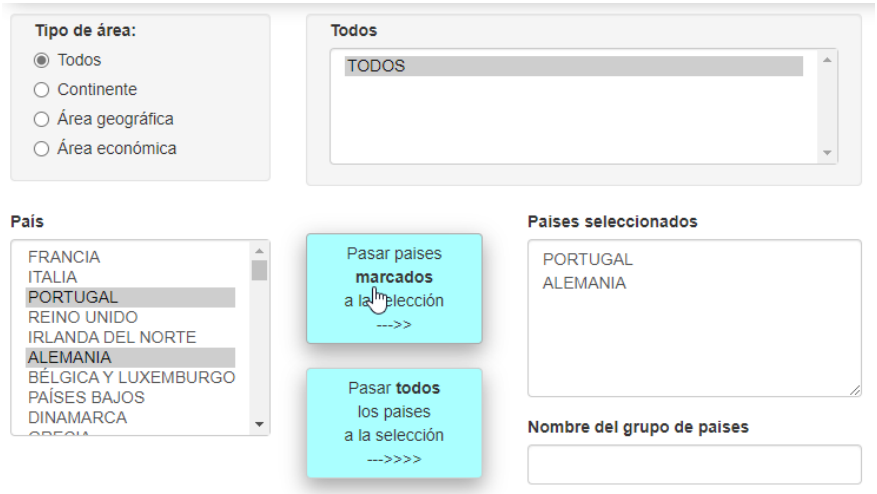
Las consultas más habituales que recibimos sobre comercio exterior se refieren a algún sector o país determinado y solo de datos anuales o datos de la parte de los años de la que hay datos en el último año.

Con la aplicación desarrollada, que se ha ido modificando a lo largo del tiempo para adaptarla a las nuevas peticiones, se pretende cubrir gran parte de las mismas.

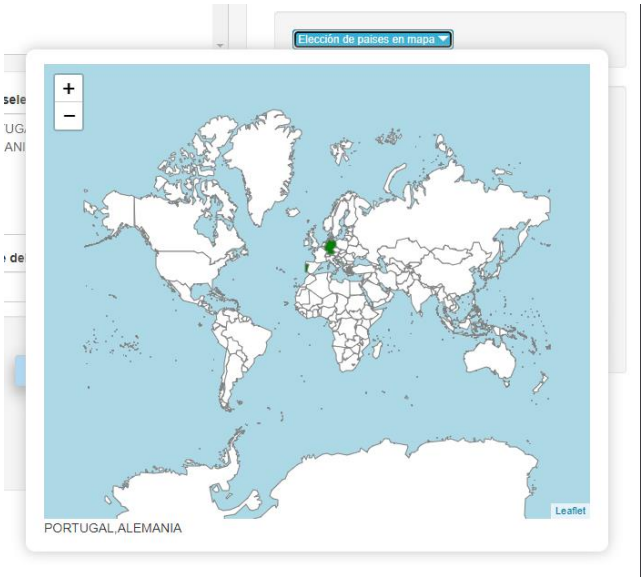
En el interfaz desarrollado se pueden observar los diversos aspectos que se contemplan:



En la parte superior izquierda se pueden elegir los países de varias formas, incluso escribiendo su nombre en el cuadro que pone "**Países seleccionados**".



En la parte superior derecha, debajo del cuadro para seleccionar la provincia, hay un botón que despliega un mapa en el que también se pueden seleccionar.



Los países se pueden marcar y desmarcar y automáticamente se ponen en el cuadro de "**Países seleccionados**".

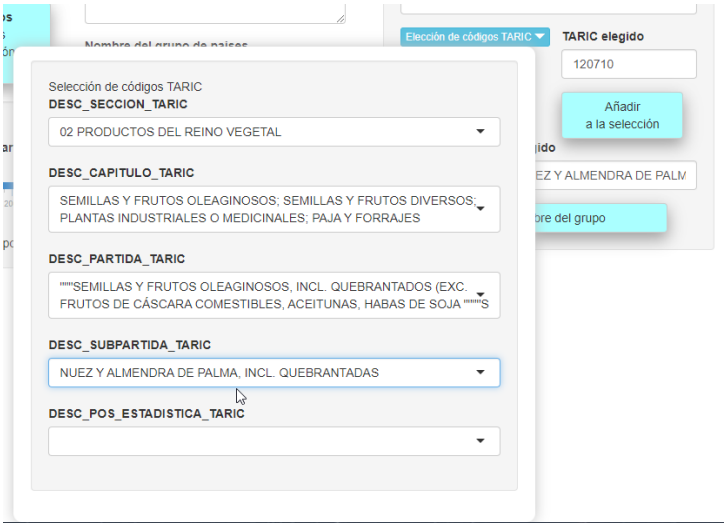
El utilizar el mapa tiene un pequeño inconveniente, en el mapa están solo los países actuales (o los que había cuando se hizo el mapa) y esta lista no coincide con la que disponemos para esta estadística, que es más amplia. Esta situación ha creado algunos problemas a la hora de trasladar lo seleccionado mediante los cuadros al mapa y viceversa. De hecho hay países que solo se pueden seleccionar a partir de los cuadros de texto y no aparecen en el mapa.

El cuadro que pone "Provincias:" permite seleccionar una provincia para obtener solo datos relativos a la misma.

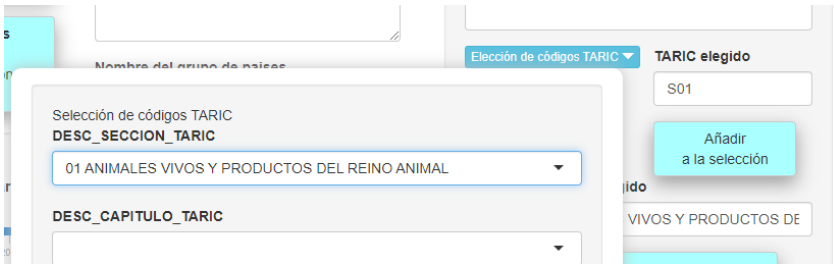
Si se selecciona TODAS, que es el defecto, los datos se refieren a Castilla y León.

En cuanto a los códigos TARIC se pueden elegir a través de la estructura o escribir directamente en el cuadro "**Códigos TARIC separados por comas**".

Cuando se selecciona a través de los cuadros que salen al pulsar el botón "**Elección de códigos TARIC**" se muestra una ventana con un cuadro de selección por cada nivel de la nomenclatura, de forma que código elegido es el que corresponde al último nivel no en blanco.



Para el caso se secciones se ha añadido una S delante para diferenciarlo del capítulo.



Con el botón "Añadir a la Selección" se añade el código elegido a lo que estuviera seleccionado antes.

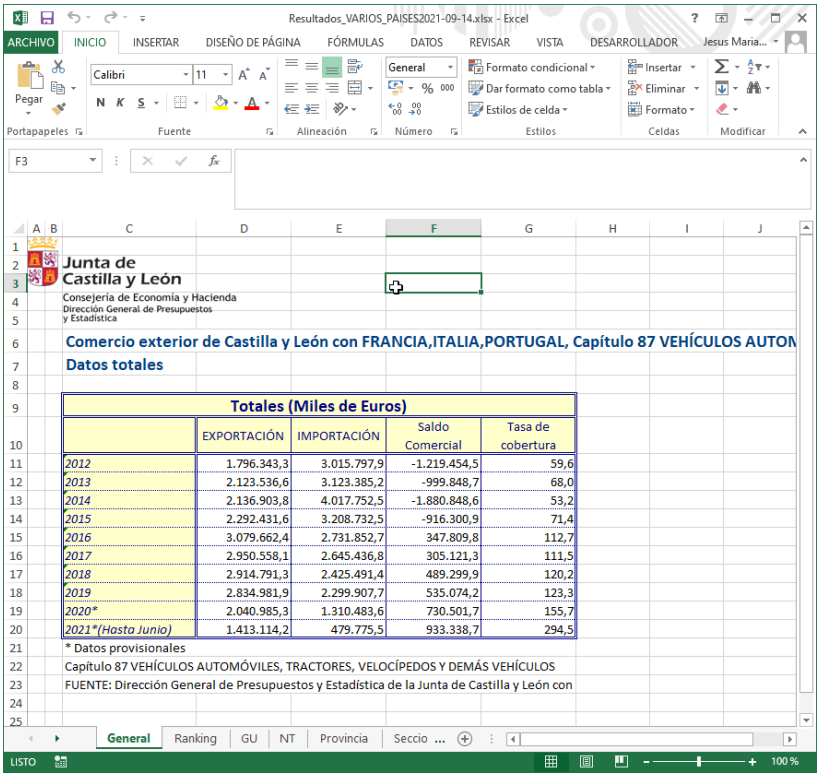
El cuadro "**Códigos TARIC (separados por comas)**" se puede editar para añadir o quitar códigos.

El cuadro "**Nombre para el grupo**" sirve para identificar el grupo de códigos seleccionados en los resultados. El botón "**Pasar al nombre del grupo**" permite pasar la descripción del último código elegido, que está en "**Descripción TARIC elegido**".

En la parte de abajo se puede seleccionar el intervalo de años a sacar, escala de la izquierda, y el año por el que se van a ordenar los datos, escala de la derecha.

También hay un cuadro, parte inferior, para sacar los datos solo de los meses de los que hay datos el último año.

Al pulsar el botón de “Realizar los cálculos” se obtiene una hoja excel con varias pestañas.



Esta aplicación hace una consulta directa a las bases de datos de comercio exterior que tenemos en Oracle utilizando el paquete DGPYEN, desarrollado por nosotros, que a su vez hace uso del paquete ROracle [4], desarrollado por Oracle. Asimismo se han utilizado los paquetes sf [5], shinyjs [6], shinyWidgets [7], leaflet [8] y tablaxlsx.

Este programa se basa en dos funciones que se habían realizado con anterioridad y a las que se ha añadido el interfaz con shiny.

Una de ellas es la que se encarga de acceder a la base de datos de Oracle, leer lo que se necesite y devuelve una lista con las tablas necesarias:

calculoTablas=function(codigospais, annoOrden,
ainicio="1997", afin=format(Sys.Date(),"%Y"),
periodo=FALSE, taric=NULL, lprovincia="00")

Lo que se pasa a esta función proviene de los distintos controles de la página.

Devuelve una lista con todas las tablas y datos necesarios:

```
return(list(codigosPaises=codigospais, AnnoOrden=annoOrden,
        totales=totales, totalesPaises=totalesPaises,
        rankExpor=rankExpor, rankImpor=rankImpor,
        posExpor=posExpor, posExport=posExport,
        posImpor=posImpor, posImport=posImport,
        totalesProv=totalesProv, totalesSeccion=totalesSeccion,
        totalesCapitulo=totalesCapitulo, totalesPartida=totalesPartida,
        totalesGU=totalesGU, totalesNT=totalesNT,
        TextoAnnos=textoAnnos,
        lprovincia=lprovincia))
```

Esta lista es la que posteriormente se pasa a la función que lo pone todo en una hoja de Excel:

```
escribirTablas=function(listaTablas,
            fescudo="escudoDGPYE(Economia y Hacienda)1.png",
            fuente="FUENTE: Dirección General de Presupuestos y Estadística de
                la  Junta de Castilla y León con datos de la AEAT, Dpto. de
                Aduanas e Impuestos Especiales.",
            cabecera="Totales (Miles de Euros)",
            titulo1=NULL, titulo2=NULL,
            notas="",
            cabtablaA=createStyle(fontColour = "#004080", fgFill = "#FFFECD",
                halign = "center", valign = "center", textDecoration = "bold",
                border= "TopBottomLeftRight",
                borderColour="#004080",wrapText=TRUE),
            tituloA=createStyle(fontColour = "#004080", fontSize="14",
                halign = "left", valign = "center", textDecoration = "bold")
            )
```

Como se puede observar además de la lista de tablas se pasan estilos para las tablas en Excel.

Datos básicos municipales

Otro tipo de consultas frecuentes son las que datos de uno o varios municipios, que en muchas ocasiones se repite de forma periódica.

Estos datos se pueden obtener del SIE, pero hay que seleccionar cada vez los municipios, por lo que nos planteamos diseñar una aplicación para facilitar estas consultas.

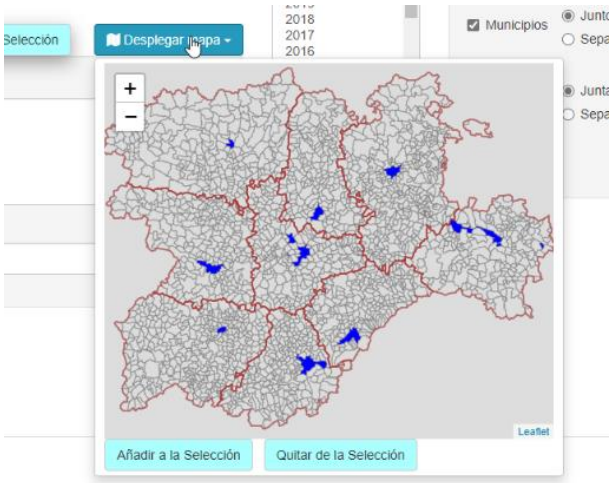
En este caso se decidió realizar la consulta utilizando el API del SIE, por lo que lo que se obtiene es exactamente lo mismo que lo que proporcionamos a través de la página web.

Se ha intentado que el interfaz sea sencillo, lo único que hay que hacer es seleccionar los municipios, bien a través de un mapa, eligiéndolos a través de unos cuadros despegables o escribiendo sus códigos INE. En el segundo caso hay que seleccionar primero la provincia y después el municipio dentro de la misma. Hay que tener en cuenta que en Castilla y León hay 2.248 municipios, por lo que una selección directa con un solo desplegable no es conveniente.

Lo que pasa al cuadro de "**Municipios seleccionados**", cuadro que se puede editar directamente, son los códigos INE.

En la parte superior están los cuadros para seleccionar los municipios mediante los cuadros despegables. Los dos primeros botones debajo de estos cuadros sirven para añadir o quitar el municipio que se haya seleccionado.

El tercero de los botones sirve para desplegar un mapa en el que se pueden elegir los municipios marcándolos en el mismo y utilizando los dos botones de la parte inferior para añadirlos o quitarlos de la selección.

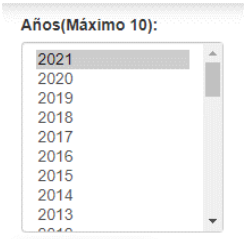


Los municipios que se muestran en azul son las capitales de provincia.

También se pueden indicar con sus códigos INE separados por comas en el cuadro que pone "**Municipios seleccionados"**, lo que permite tener almacenados los códigos en algún sitio y con copiarlos y pegarlos en el cuadro pertinente estaría hecha la selección.

Se ha implementado un sistema sencillo para poder guardar estas consultas en la misma aplicación y así poder reutilizarlas. Los botones "Leer municipios" y "Grabar municipios" leen y graban un fichero con los códigos de los municipios seleccionados y le pone el nombre que haya en el cuadro "**Nombre del grupo de municipios**".

En la parte superior central se eligen los años de los que se desean datos. El máximo de 10 es porque que con más la consulta puede tardar mucho.

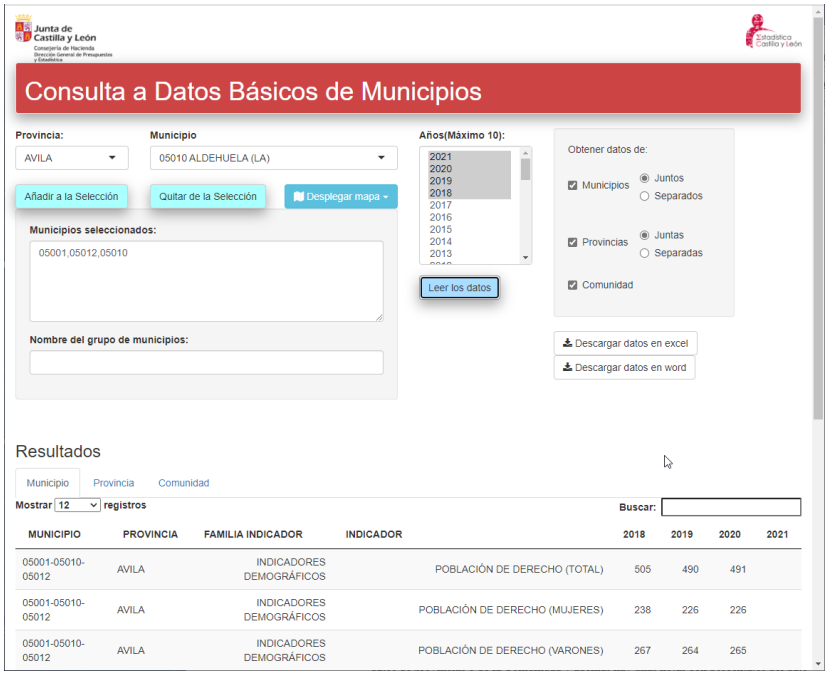


En la parte superior derecha hay una serie de opciones que permiten sacar los datos de los municipios agrupados o una hoja por municipio o incluso no sacarlos.

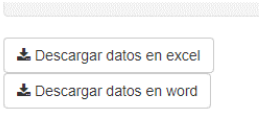
Además de los datos del municipio se pueden obtener los de las provincias en las que están, marcando el cuadro correspondiente, y en la misma zona se puede indicar si se quieren agrupados o una hoja por provincia.

Asimismo se puede indicar que se saquen o no los datos de la Comunidad.

Con el botón “**Leer los datos**” se realiza la consulta y los resultados se sacan en la parte inferior, donde pone **Resultados.**



Una vez leídos los datos aparecen dos botones en la parte derecha que permiten descargar los datos en formato excel o word.

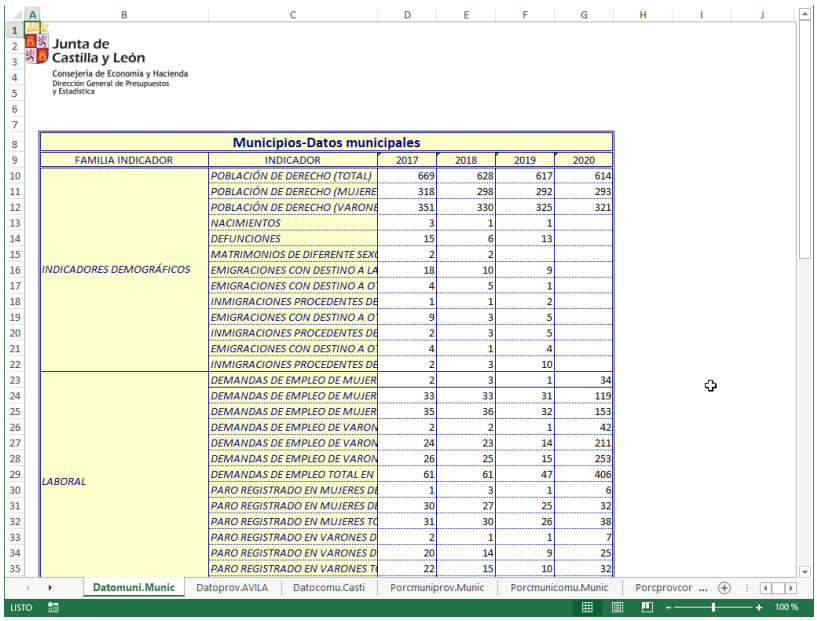


Lo que se escriba en “**Nombre del grupo de municipios**“, sirve para hacer referencia al grupo de municipios en los resultados.

En excel, además de los datos solicitados y siempre que se hayan pedido, se obtienen los datos de provincias o de la Comunidad y además hay unas hojas con porcentajes del dato del municipio sobre la provincia y la Comunidad y del de la provincia sobre la Comunidad.

Para obtener la hoja excel se utiliza el paquete tablaxlsx, que tiene como base el paquete openxlsx.

Para el fichero word se utilizan los paquetes officer [9] y flextable [10].

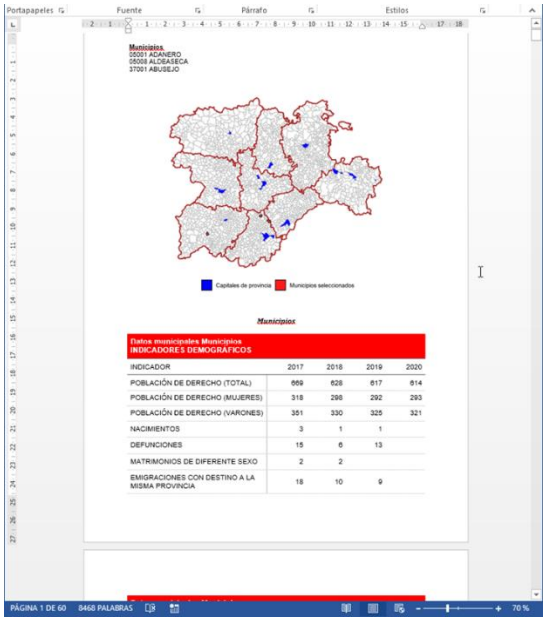


Junta de Castilla y León
Consejería de Economía y Hacienda
Dirección General de Presupuestos y Estadística

Municipios-Datos municipales

FAMILIA INDICADOR	INDICADOR	2017	2018	2019	2020
INDICADORES DEMOGRÁFICOS	POBLACIÓN DE DERECHO (TOTAL)	669	628	617	614
	POBLACIÓN DE DERECHO (MUJERE	318	298	292	293
	POBLACIÓN DE DERECHO (VARONE	351	330	325	321
	NACIMIENTOS	3	1	1	
	DEFUNCIONES	15	6	13	
	MATRIMONIOS DE DIFERENTE SEX	2	2		
	EMIGRACIONES CON DESTINO A LA	18	10	9	
	EMIGRACIONES CON DESTINO A O	4	5	1	
	INMIGRACIONES PROCEDENTES DE	1	1	2	
	EMIGRACIONES CON DESTINO A O	9	3	5	
	INMIGRACIONES PROCEDENTES DE	2	3	5	
	EMIGRACIONES CON DESTINO A O	4	1	4	
	INMIGRACIONES PROCEDENTES DE	2	3	10	
LABORAL	DEMANDAS DE EMPLEO DE MUJER	2	3	1	34
	DEMANDAS DE EMPLEO DE MUJER	33	33	31	119
	DEMANDAS DE EMPLEO DE MUJER	35	36	32	153
	DEMANDAS DE EMPLEO DE VARON	2	2	1	42
	DEMANDAS DE EMPLEO DE VARON	24	23	14	211
	DEMANDAS DE EMPLEO DE VARON	26	25	15	253
	DEMANDAS DE EMPLEO TOTAL EN	61	61	47	406
	PARO REGISTRADO EN MUJERES DE	1	3	1	6
	PARO REGISTRADO EN MUJERES DE	30	27	25	32
	PARO REGISTRADO EN MUJERES TO	31	30	26	38
	PARO REGISTRADO EN VARONES D	2	1	1	7
	PARO REGISTRADO EN VARONES D	20	14	9	25
	PARO REGISTRADO EN VARONES T	22	15	10	32

Datomuni.Munic | Datoprov.AVILA | Datocomu.Casti | Porcmuniprov.Munic | Porcmunicomu.Munic | Porcprovcor ...



Municipios
05001 ADANERO
05008 ALDEASECA
37001 ABUSEJO

Capitales de provincia | Municipios seleccionados

Municipios

Datos municipales Municipios
INDICADORES DEMOGRÁFICOS

INDICADOR	2017	2018	2019	2020
POBLACIÓN DE DERECHO (TOTAL)	669	628	617	614
POBLACIÓN DE DERECHO (MUJERES)	318	298	292	293
POBLACIÓN DE DERECHO (VARONES)	351	330	325	321
NACIMIENTOS	3	1	1	
DEFUNCIONES	15	6	13	
MATRIMONIOS DE DIFERENTE SEXO	2	2		
EMIGRACIONES CON DESTINO A LA MISMA PROVINCIA	18	10	9	

PÁGINA 1 DE 60 8468 PALABRAS 70 %

Para este programa se han empleado funciones realizadas con anterioridad, en particular la función que se encarga de leer los datos desde el Sistema de Información Estadística de Castilla y León, mediante su API:

leerTablas=function(annos,municipios=NULL, provincias=NULL, comunidad=FALSE,
variables=NULL,
juntarMun=TRUE, juntarProv=TRUE,
nombreGrupoMun=NULL, nombreGrupoProv=NULL)

A esta función se le pasan los datos de los controles de página creada con shiny y devuelve una tabla con los datos obtenidos.

Esta tabla es la que se muestra en la parte inferior de la página y también es la que se utiliza para la función que crea el fichero de Excel:

```
escribirFicheroxlsx=function(tablas,wb=NULL,escudo=NULL,fuente=NULL,
                    campos=c("FAMILIA INDICADOR", "INDICADOR", "FECHA",
                              "Valor"),
                    ...)
```

Los parámetros adicionales que se pueden pasar, … , se refieren al formato con el que se van a poner las tablas en Excel y son los que se utilizan en la función *escribirTabla* del paquete *tablaxlsx*:

```
escribirTabla(tabla, wb=NULL, hoja=NULL, fichero=NULL,
          limpiarFilas=TRUE, limpiarColumnas=TRUE, limpiarValores=NA,
          cabecera = "",  fuente = "",  notas = "",
          fila = 7,  columna = 3, decimales = 1, porcentaje = FALSE,
          cabecerasFila = TRUE, cabecerasColumna = TRUE,
          cabeceraFilas="",cabeceraColumnas="",
          anchoCabecera = 10,anchoDatos = 14,
          escudo = NULL,posEscudo=c(1,1),
          tamEscudo=c(2.7,4.5),unidadesEscudo="cm",
          estilos=options("tablaxlsx.Estilos")[[1]],
          bordes = c("TABLA", "CABECERA", "CABECERASFILA",
                     "CABECERASCOLUMNA", "DATOS"),
          estilosBordes = NULL)
```

La creación del fichero Word se hace mediante la función:

```
escribirFichas=function(tablas,doc=NULL,campoZona=1,cabecera1="",
           campoG1="FAMILIA INDICADOR",
           camposG2=c("INDICADOR"),
           campoF="FECHA", campoV="Valor",
           colorCabFondo="red", colorCabTexto="white",
           documentoBase="modeloFicha.docx", fuente="")
```

que se basa en el paquete *officer*.

Indicadores de la Agenda 2030

A raíz del grupo de trabajo creado entre las CCAA para el desarrollo a nivel de comunidad de los indicadores de los ODS2030, se planteó la necesidad de buscar una forma de difundir dichos indicadores.

Inicialmente se han utilizado hojas excel que están en el portal de Estadística, pero para lograr algo más dinámico, se ha desarrollado una aplicación con shiny, a la que se ha dado acceso desde el exterior: https://servicios4.jcyl.es/ods2030/

Esta aplicación parte de los datos de los indicadores en formato csv, no necesariamente con todos los campos que se han consensuado para el formato de intercambio entre comunidades.

Aunque de momento solo presenta los datos de Castilla y León, se pretende ampliarlo para que pueda presentar también los de otras comunidades, por separado o conjuntamente.

El interfaz es sencillo:

En la primera página sale una pequeña introducción.

La línea de iconos de la parte superior y el botón “Descargar todos los datos en csv”, de la parte inferior, se mantienen en todas las páginas.

Cuando se pulsa en alguno de los objetivos se obtienen una página con sus metas e indicadores, solo de los que hay algún dato.

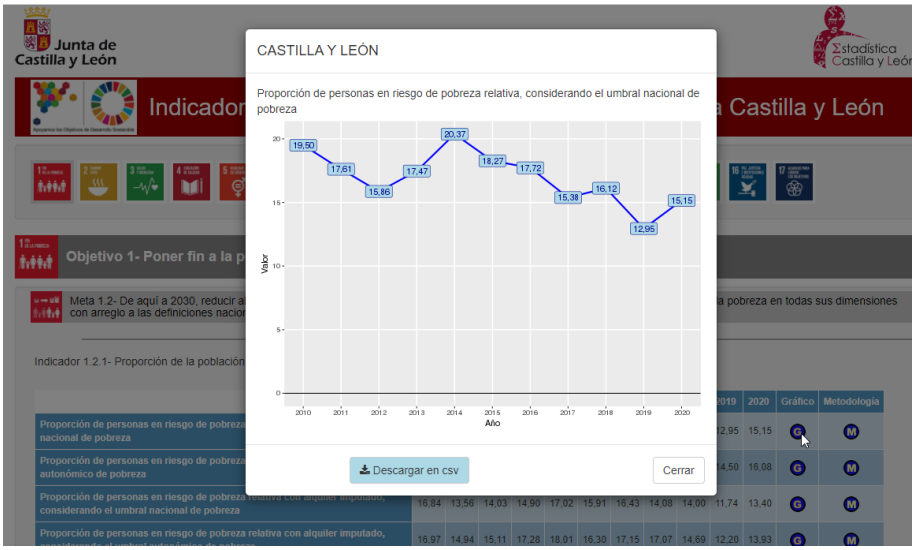
	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	Gráfico	Metodología
Proporción de personas en riesgo de pobreza relativa, considerando el umbral nacional de pobreza	19,50	17,61	15,86	17,47	20,37	18,27	17,72	15,38	16,12	12,95	15,15	G	M
Proporción de personas en riesgo de pobreza relativa, considerando el umbral autonómico de pobreza	19,55	19,07	18,42	19,09	21,63	19,27	18,34	19,63	18,40	14,50	16,08	G	M
Proporción de personas en riesgo de pobreza relativa con alquiler imputado, considerando el umbral nacional de pobreza	16,84	13,56	14,03	14,90	17,02	15,91	16,43	14,08	14,00	11,74	13,40	G	M
Proporción de personas en riesgo de pobreza relativa con alquiler imputado, considerando el umbral autonómico de pobreza	16,97	14,94	15,11	17,28	18,01	16,30	17,15	17,07	14,69	12,20	13,93	G	M

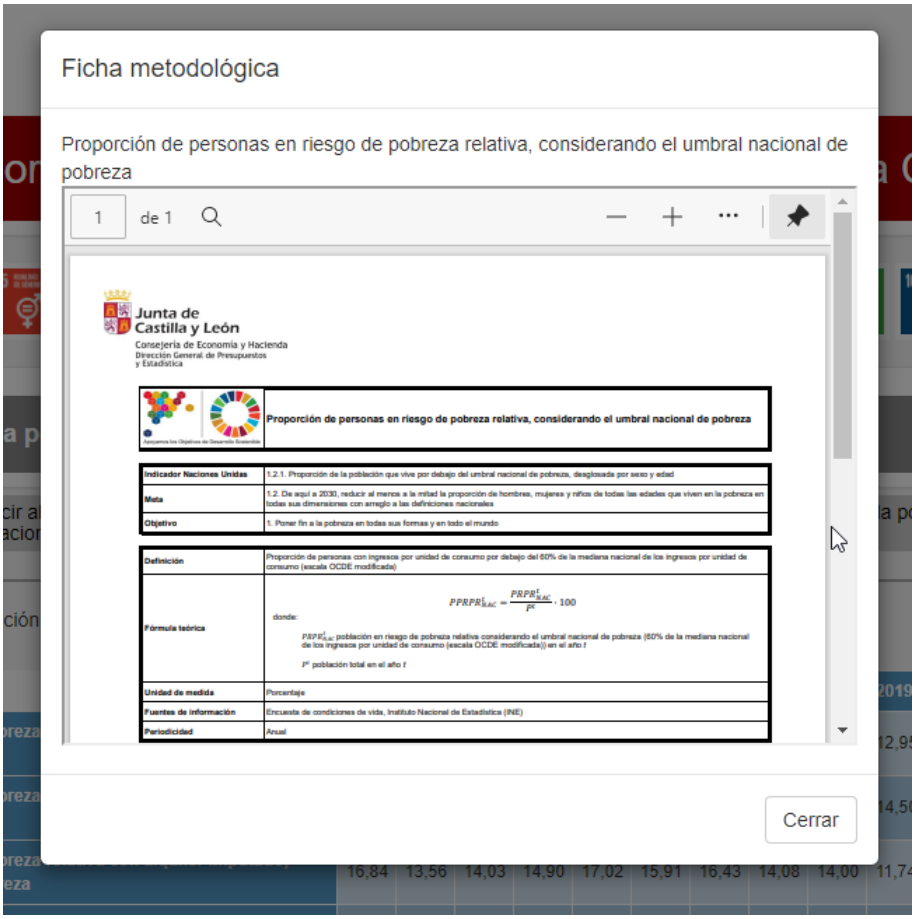
Los datos salen agrupados por meta e indicador y se presentan en forma de tabla.

La implementación en Shiny de esta presentación se ha hecho de forma que se crean dinámicamente los componentes y los eventos asociados a los mismos. Para ello se ha usado el paquete shinyjs y componentes del tipo uiOutput para contener los elementos que se van generando en el programa.

En cada línea se han colocado dos iconos: uno para obtener un gráfico de los datos que están en la línea y otro que saca, en una hoja aparte, un pdf con la metodología.

En la ventana del gráfico se pueden descargar los datos en formato csv.





La aplicación se ha desarrollado pensando en futuras ampliaciones, por lo que se ha intentado que fuera fácilmente adaptable a otras situaciones con distintito número de objetivos y metas.

Además de leer los datos de un fichero csv se leen tablas con los objetivos, las metas, los indicadores y los subindicadores. Dichas tablas contienen los textos, la numeración y otras características de los mismos.

La creación de la barra de iconos se ha hecho de forma dinámica:

```
output$RejillaObj=renderUI({
   lapply(1:nrow(objetivos), function(i){
    sel1= paste0("background-color:",color1[as.numeric(i==nObj())+1],";")
    img(id=paste0("Objetivo",i),src=paste0('imagenesMetas1/Objetivo ',i,'.png'),
              title=paste0("Objetivo ",i,". ",objetivos[as.character(i),"Objetivo"]),
        alt=paste0("Objetivo ",i,". ",objetivos[as.character(i),"Objetivo"]),
        width=55,height=55,style=paste0("padding:4px;",sel1))
   })
  })
```

Y la funcionalidad, al pulsar en cada icono, se ha hecho de la misma forma, utilizando una variable reactiva, *nObj*, para marcar el objetivo a presentar:

```
shinyjs::onclick("Presentacion",{
    nObj(0)
  })

  a1=lapply(1: nrow(objetivos), function(i){
          shinyjs::onclick(paste0("Objetivo",i),{
                              nObj(i)
          })
  })
```

Lo mismo se hace con las metas y los indicadores. Para la funcionalidad a los botones que muestran el gráfico y la metodología se emplean dos variables reactivas, *subI* y *subIM*, que son las que provocan que se muestren.

```
a1=lapply(rownames(subindicadores), function(nsi){
        shinyjs::onclick(paste0("Bgraf_",gsub("[.]","_",nsi)),{
                          subI(nsi)
        })
 })

  a2=lapply(rownames(subindicadores), function(nsi){
          shinyjs::onclick(paste0("Meto_",gsub("[.]","_",nsi)),{
                              subIM(nsi)
          })
  })
```

Conclusiones

La principal conclusión es que el empleo de herramientas del tipo de R y shiny facilitan el trabajo repetitivo y hacen posible que el conocimiento sobre las mismas sea también aprovechado por aquellos que no las dominan.

Pero hay que tener en cuenta que el desarrollo de cualquier aplicación exige unos conocimientos de las herramientas que no todos los que trabajan en estadística tienen, si bien se pueden abordar mediante equipos multidisciplinares.

Por otro lado es importante valorar el tiempo que se dedica al desarrollo, ya que si un proceso va a ser ejecutado pocas veces puede no ser “rentable” intentar abordarlo de esta forma.

Referencias

[1] R, Core Team, «R: A Language and Environment for Statistical Computing,» Vienna, 2021.

[2] W. Chang, J. Cheng, J. J. Allaire, C. Sievert, B. Schloerke, Y. Xie, J. Allen, J. McPherson, A. Dipert y B. Borges, «shiny: Web Application Framework for R,» 2021.

[3] J. M. R. Rodríguez, «tablaxlsx: Write Formatted Tables in Excel Workbooks,» 2020.

[4] D. Mukhin, D. A. James y J. Luciani, «ROracle: OCI Based Oracle Database Interface for R,» 2016.

[5] E. Pebesma, «Simple Features for R: Standardized Support for Spatial Vector Data,» *The R Journal*, vol. 10, p. 439–446, 2018.

[6] D. Attali, «shinyjs: Easily Improve the User Experience of Your Shiny Apps in Seconds,» 2020.

[7] V. Perrier, F. Meyer y D. Granjon, «shinyWidgets: Custom Inputs Widgets for Shiny,» 2021.

[8] J. Cheng, B. Karambelkar y Y. Xie, «leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library,» 2021.

[9] D. Gohel, «officer: Manipulation of Microsoft Word and PowerPoint Documents,» 2021.

[10] D. Gohel, «flextable: Functions for Tabular Reporting,» 2021.