

Paquete esindex^{*}

Javier Bezos^{**}

2019-01-25

This package defines the command `\esindex` which eases writing Spanish index entries:

`\esindex{cañón}`

is equivalent to

`\index{can^^ffon@cañón}`

As you can see, the package generates the sort key within \TeX itself, which has a number of advantages.

Although it is a specifically Spanish tool (so most of the documentation is in Spanish), the idea behind this package may be applied to other languages, and tools to adapt it are provided, too. They are explained below.

Version 1.5 provides tools for other languages and adds support for `luatex` and `xetex`.

1. Spanish

Este paquete ha sido diseñado para facilitar la escritura de índices correctamente alfabetizados en castellano. Su principal orden es `\esindex`, que convierte a una forma adecuada su argumento. Así por ejemplo,

`\esindex{cañón}`

equivale a

`\index{can^^ffon@cañón}`

^{*}Este archivo está actualmente en la versión 1.5 beta con fecha 2019-01-25. Esta copia del manual se compuso el 26 de enero del 2019.

^{**}Para comentarios y sugerencias: <http://www.texnia.com>.

No es necesario usar **babel** salvo, lógicamente, si los acentos están escritos en forma de abreviaciones ('a, 'e, etc.) en lugar de con los caracteres reales. En este último caso, el paquete utiliza ciertas órdenes internas de **babel**, por lo que no puedo garantizar su funcionamiento correcto con versiones distintas a las 3.6 a 3.27. En caso de que **esindex** sea incompatible con futuras versiones de **babel** (lo cual no es realmente probable) intentaré adaptarlo en el menor tiempo posible.

Salvo el carácter **actual** (normalmente @) se pueden usar todos los caracteres especiales de *MakeIndex*. Se pueden aplicar convenciones diferentes a las normales, pero en este caso hacen falta ajustes adicionales en caso de que los modificados sean **actual**, **encap**, **level** o **quote**. En ese caso basta con indicar los caracteres que hay que usar como opciones de paquete. Por ejemplo, si para **quote** decidimos usar \$ en nuestro archivo .ist particular, tendríamos que llamar al paquete del siguiente modo:

```
\usepackage[quote=$]{esindex}
```

Es importante observar que, a diferencia de la opción para alemán de *MakeIndex*, el uso de " en abreviaciones como "u es completamente legítimo, ya que el paquete reconoce tal combinación y la trata aparte. Lo mismo vale para ' o ~ en caso de que se usaran como carácter especial. Es decir

```
\esindex{"!' }Cig"ue'nas{"!}|textbf}
```

equivale a

```
\index{"!' }Ciguen^^ffas{"!}@{"!' }Cig\"ue\"nas{"!}|textbf}
```

Sin embargo, el uso del carácter **quote** ante **encap** o **level** no se detecta a menos que el grupo esté encerrado entre llaves. Por ejemplo, en lugar de `\esindex{Pleca: "|}` debe escribirse `\esindex{Pleca: {"|}}`. (En realidad en este caso podría haberse usado `\index`. Es tan sólo un ejemplo.)

Aunque el hecho de que @ no se pueda usar en `\esindex` hace que todavía algunas entradas se tengan que hacer a mano, la mayor parte del trabajo se ve considerablemente simplificado.

Finalmente, hay que señalar que con este paquete no se crea en el índice una entrada propia para la palabras que empiezan por ñe, sino que tan sólo se añaden al final de la ene. En el rarísimo caso de que hubiera palabras que empiezan por ñe habría que modificar el archivo .ind a mano o bien redefinir de algún modo las entradas generadas.

La versión 1.3 elimina una incompatibilidad con recientes versiones de L^AT_EX y añade nuevas funciones:

- Opción de paquete **ignorespaces**: al formar la clave de ordenación se suprimen los espacios, de forma que:

adentro < a donde = adonde.

- Opción de paquete `replaceindex`: el comportamiento de `\index` se reemplaza por el de `\esindex`, aunque en este caso no es posible introducir entradas que no se adapten a lo requerido por `\esindex`.
- La orden `\ignorewords` da una lista de palabras separadas por comas que no se cuentan en la ordenación. Por ejemplo, con `\ignorewords{de}` tendríamos:

pino albar < pino laricio < pino de montaña.

Distingue la caja, por lo que las formas con mayúsculas hay que darlas explícitamente, si hicieran falta.

Algunas funciones adicionales son:

- La lista de tókenes `\everyesindex` permite dar definiciones locales para establecer el comportamiento de otras órdenes. Por ejemplo:

```
\everyesindex{\renewcommand\emph[1]{#1}}
```

elimina de la clave esa orden. Con:

```
\everyesindex{\renewcommand\emph[1]{#1'}}
```

la entrada en cursiva iría detrás de la redonda, si la hubiera. Es una técnica que se puede emplear en otros casos para reajustar el orden de entradas idénticas.

- La orden `\esindexsort` permite predefinir claves asociadas a entradas concretas, para ajustar su ordenación (lo que normalmente se consigue añadiendo texto adicional para que `makeindex` lo tenga en cuenta). Estas correspondencias deben darse antes de la aparición del primer `\esindex` con ese término, y las claves se procesan posteriormente con `ignorespaces`, `\ignorewords` y `\everyesindex`, si están activadas. Por ejemplo:

```
\esindexsort{adonde}{adonde'1}
\esindexsort{adónde}{adonde'2}
\esindexsort{a donde}{a donde'7}
\esindexsort{a dónde}{a donde'8}
```

daría el orden *adonde*, *adónde*, *a donde*, *a dónde*, con `ignorespaces`, o bien *a donde*, *a dónde*, [probablemente otros términos], *adonde*, *adónde*, sin `ignorespaces`.

2. Other languages

First, you very likely want to ignore de Spanish specific settings with the package option `nospanish`. What `\esindex` does is the following:

1. Replacements set by `\esindexsort`. See an example above.
2. `\esindexreplace` in `\everyesindex`. See an example below.
3. Replacements by L^AT_EX (protected) expansion, including redefinitions in `\everyesindex` and, in Spanish, `\‘`, `\"` and `\~`. You may force protected expansion at any point inside `\everyesindex` with `\esindexexpandkey`. In addition, the sort key is stored in `\esindexkey`, which can be manipulated directly.
4. Removal of words listed in `\ignoredwords` (maybe not the logical place, but real life is not always logical). For example, with `\ignoredwords{de,la}` the words “de” and “la” (preceded and followed by spaces) are removed from the key.
5. Removal of spaces if `ignorespaces`. Very often this is similar to the `-l` option of *MakeIndex*.
6. Replacement of ‘actual’ as set by `\esindexactual`, but still based on the original `\esindex` argument, without changes. For example, with:

```
\esindexactual{Felipe II}{Felipe II, \textit{rey de España}}
```

just write `\esindex{Felipe II}` (as many times as you want), and the entry will show “Felipe II, *rey de España*” (the sort key is still based on “Felipe II”, of course).

Do you find it chaotic? Well, you are right. After all this package is for Spanish indexes with some readjustments for it to be adapted to other languages. A general solution deserves another package. Feel free to create one based on this package (MIT license), if you like.

Here is an example of an `\everyesindex`. Like other `\every...`’s, it is a token register:

```
\everyesindex{%  
  \renewcommand\"[1]{#1e}%  
  \renewcommand\textit[1]{#1}%  
  \esindexexpandkey  
  \esindexreplace{ä}{ae}%  
  \esindexreplace{ü}{ue}%  
  \esindexreplace{ö}{oe}}
```

What it does is:

1. Redefines `\textit` so that it is removed to build the key. With `\everyesindex{\renewcommand\textit[1]{#1'}}` italics would be sorted after upright.
2. Redefines `\"`. Of course, this only works correctly if used for this precise expansion.
3. Applies the previous changes with a protected expansion.
4. Make a direct replacement of some characters.

Remember these changes are not shown – they are used by *MakeIndex* to sort the entries.

As a convenience tool, `\esindexlastchar` is `^^ff` or `^^^^ffff`, depending on the engine.