

Paquete `esindex`*

Javier Bezos**

2008-09-28

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

```
\esindex{cañón}
```

is equivalent to

```
\index{can^^ffon@cañón}
```

Since it is an specifically Spanish tool, the documentation is in 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}
```

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 3.8. En caso de que `esindex` sea incompatible con futuras versiones de `babel` intentaré adaptarlo en el menor tiempo posible.

Salvo el carácter `actual` (normalmente `@`) se pueden usar todos los caracteres especiales de *MakeIndex*. Se pueden usar convenciones diferentes a las normales, pero en este caso hacen falta ajustes adicionales en caso de que los

*Este archivo está actualmente en la versión 1.4 con fecha 2008-09-28. Esta copia del manual se compuso el 13 de febrero del 2010.

**Para comentarios y sugerencias: <http://www.tex-tipografia.com>.

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 `eñe`, sino que tan sólo se añaden al final de la `ene`. En el rarísimo caso de que hubiera palabras que empiezan por `eñe` habría que modificar el archivo `.ind` a mano.

La versión 1.3 elimina una incompatibilidad con recientes versiones de \LaTeX 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.

La versión 1.4 elimina un pequeño bicho con "U y añade nuevas funciones:

- 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`.