

Matemáticas en emacs

Mario Román

<2016-09-26 Mon 15:07>

Actualizado a 06 de agosto de 2017.

Apuntes a ordenador

Tomar apuntes de matemáticas con el ordenador es una tarea difícil por lo complejo que es transcribir la notación (símbolos, letras en otros alfabetos, índices...) y la velocidad necesaria. Además, los diagramas, dibujos, flechas o anotaciones no textuales que tomamos parecen imposibles de transcribir fielmente a un fichero.

Latex permite notación matemática pero no facilita la velocidad (al fin y al cabo, Latex estaría pensado para edición de libros, no para tomar apuntes). **Markdown**, por otro lado, simplifica mucho el poder escribir, pero pierde mucha potencia respecto a Latex. Así que una solución es usar **org-mode** como lenguaje de marcado; es relativamente simple y fácilmente legible, como markdown, mientras que permite visualización de latex conforme se edita, programación literaria y exportación a latex y html, permitiendo la inclusión de código latex o html arbitrario para exportarlo.

Ventajas de org-mode

org-mode tiene internamente un lenguaje de marcado similar al de markdown, con la ventaja para el usuario de Emacs de que está adaptado especialmente al editor. Permite escribir fórmulas en Latex y exportar luego a **.tex** y **.pdf**, controlando las opciones de Latex.

Una fórmula en Latex puede escribirse directamente en **org-mode** incluyéndola entre `\(...\)`, si está dentro de una línea de texto (como en $i * x = x$); o entre `\[... \]`, cuando queremos que se muestre aparte del texto como en el siguiente ejemplo

$$\sum_{n=0}^{\infty} \frac{1}{2^n}.$$

Cuando terminamos de escribirla podemos [previsualizarla](#) directamente con `C-c C-x C-l`, como si fuera un editor [WYSIWYG](#).

Si vamos a acabar exportando a latex podemos insertar entornos como los que ofrece la biblioteca de la AMS con

```
#+begin_theorem
[texto del teorema]
#+end_theorem
```

o incluso definir nuestros propios entornos y usarlos después de la misma forma. Además, también permite la inclusión de bloques de código con

```
#+begin_src ruby
[código en ruby]
#+end_src
```

en la mayoría de lenguajes de programación y algunos especialmente útiles para matemáticas como [Sage](#).

Zoom

Un problema menor (y quizá sólo mío) al configurar todo esto es que las fórmulas previsualizadas parecen demasiado pequeñas. Aunque estén en proporción con el texto, cuesta más leerlas; y cuando aumentamos el tamaño del texto con `C-x C-+`, las fórmulas no se amplían con él. Para conseguir que lo hagan hay que incluir el siguiente trozo de código en el archivo de configuración de Emacs, mezcla de dos respuestas de [thisirs](#) y [Mark](#) en Stack Overflow:

```
(defun update-org-latex-fragment-scale ()
  (let ((text-scale-factor
        (expt text-scale-mode-step text-scale-mode-amount)))
    (plist-put org-format-latex-options
               :scale (* 1.2 text-scale-factor)))
  )
(add-hook
```

```
'text-scale-mode-hook  
'update-org-latex-fragment-scale)
```

Después de volver a cargar el archivo de configuración, las fórmulas nuevas deberían volver a ajustarse con el texto.

Aumentando la velocidad de escritura

Mi objetivo principal con todo esto era escribir matemáticas más rápidamente, así que [pregunté sobre autocompletado](#) y concluí en usar `latex-math-mode`. Esto permite incluir comandos de Latex con atajos de teclado. En su configuración original usa el caracter ‘ para acceder a ellos, así que ‘-a escribe `\alpha`. Yo he decidido cambiar el acento invertido, que ya cuesta dos pulsaciones en el teclado en español, por la ζ , que no la suelo usar. Además de los que incluye el paquete por defecto, se pueden escribir atajos propios.

Además de `latex-math-mode`, `cdlatex` es un modo de Emacs escrito por el mismo creador de `org-mode`, que facilita la escritura rápida de símbolos de latex. Puede añadirse a `org-mode` usando

```
(use-package cdlatex  
  :ensure t)  
  
(add-hook 'org-mode-hook 'turn-on-org-cdlatex)
```

y las instrucciones del paquete pueden encontrarse [aquí](#).

Por otro lado, empecé a usar `yasnippets` en Emacs. Son cómodos, fáciles de programar, y me permiten simplificar tareas como escribir diagramas conmutativos o complejos simpliciales en pocos pasos.

Diagramas conmutativos

Ahora estoy escribiendo sobre álgebra homológica y teoría de categorías, así que la mayoría de lo que escribo usa secuencias exactas y diagramas conmutativos.

Para las secuencias exactas, por ejemplo, tengo simplemente una plantilla con `yasnippet`, que me deja incluirlas escribiendo `complex_` y pulsando `<tab>`:

```
# -*- mode: snippet -*-
# name: complex
# key: complex_
#--
\begin{aligned*} $1 \overset{$6}{\longrightarrow}
$2 \overset{$7}{\longrightarrow}
$3 \overset{$8}{\longrightarrow}
$4 \overset{$9}{\longrightarrow}
$5 \end{aligned*}
```

Para los diagramas conmutativos, la solución es un poco más compleja. El paquete **tikz** de Latex es muy útil para escribirlos pero tiene una sintaxis es demasiado recargada; así que existe **tikz-cd**, que simplifica esa sintaxis para centrarla en diagramas conmutativos. Para usarlo, hay que empezar por incluir en el archivo de configuración `init.el` las siguientes líneas

```
(add-to-list
 'org-latex-packages-alist '("" "tikz" t))

(eval-after-load "preview"
 '(add-to-list
  'preview-default-preamble
  "\\PreviewEnvironment{tikzpicture}"
  t))
```

que pueden modificarse y escribirse de forma análoga si queremos que Emacs use internamente otros paquetes de Latex.

Además en mi caso, tuve que cambiar el programa con el que generaba las imágenes. Parece funcionar sólo **imagemagick** cuando queremos usar diagramas conmutativos

```
(setq org-latex-create-formula-image-program 'imagemagick)
```

Cabeceras en Latex

Cuando necesitamos funcionalidad adicional que ofrece Latex en bibliotecas aparte, como usar **tikz-cd**, podemos incluirlas en la cabecera del archivo `org` como:

```
+latex_header : amsthm+latex_header : amsmath+latex_header : tikz - cd
```

Además, si queremos que sea funcionalidad que sólo se use en la exportación de latex pero no en la previsualización, podemos incluirla con `#+latex_header_extra`.

Archivos de configuración de org

Para evitar tener que repetir varias veces la misma cabecera en varios archivos, podemos usar un sólo archivo para escribir matemáticas y fraccionarlo en secciones temáticas. Cuando necesitamos tratar una sección, podemos usar la funcionalidad de *narrowing* de **org** para tratar sólo una sección.

Otra opción es la de tener un sólo archivo de configuración `math.setup` con reglas de la forma

```
+latex_header : amsthm+latex_header : amsmath+latex_header : tikz - cd
```

y cargarlo en cada archivo `org` con `#+SETUPFILE: math.setup`.

Ejemplos

Pueden encontrarse ejemplos de uso en [este repositorio](#) con mis apuntes de matemáticas. Mi archivo de configuración de Emacs está disponible en [mroman42/emacs.d](#).