

Capturando links en org-mode

Mario Román

<2017-03-23 Thu 14:54>

Capturando links

El objetivo de este post es describir el proceso que uso para capturar links desde Firefox y almacenarlos en un archivo de org-mode, que puede leerse luego desde el propio navegador.

Org-capture

La primera parte será activar `org-capture` y `org-protocol` en Emacs. `org-capture` puede ser usado también para escribir pequeñas ideas y almacenarlas en ficheros org rápidamente; pero, en este caso, lo usaremos sólo para recibir el link desde Firefox.

```
(require 'org-protocol)
```

Lo primero que debemos hacer es definir la plantilla que queremos usar para capturar los links. La sintaxis usada se explica en la [documentación](#) de org-mode, así como las opciones que usa después:

```
(setq org-capture-templates
  (quote (
    ("x" "org-protocol" entry (file "~/links.org")
      "** %c %?" :kill-buffer t :prepend t))))
```

En este caso, guardará los links en un archivo llamado `links.org` en el directorio `home`. Pueden añadirse [opciones](#) para explicitar en qué punto exacto del archivo queremos insertar el link una vez lo capturemos.

Extensión de firefox

El uso de `org-protocol` desde Firefox puede gestionarse más detalladamente usando `marcadores` que activen la captura. En nuestro caso, como sólo necesitamos la configuración básica, podemos ahorrarnos este trabajo y dejárselo a la extensión `org-capture for Firefox`.

La extensión nos deja elegir en sus opciones si queremos que use una nueva ventana (me parece lo más conveniente y además veremos cómo controlarla luego con `i3`); la letra asignada a la plantilla (en nuestro caso hemos usado la `x` en la configuración anterior); y el atajo de teclado para activarla, que por defecto es `Ctrl-Alt-r`.

Una vez activemos la extensión, podremos capturar enlaces pulsando el atajo.

Cerrando ventanas

Un problema menor al usar esta extensión es que crea ventanas de Emacs que no cierra al terminar. Otro problema es que suele crear también un buffer inicial al lado de nuestro buffer de captura. Podemos usar `hooks` para forzarlo a corregir ese comportamiento:

```
(add-hook 'org-capture-mode-hook 'delete-other-windows)
(add-hook 'org-capture-after-finalize-hook 'delete-frame)
```

Integrándolo en i3

Para los usuarios de `i3`, puede ser útil poder controlar dónde y cómo se crea la ventana de captura de Emacs. Un ejemplo de configuración es la siguiente, que coloca la ventana como flotante y en el centro de la pantalla; como si fuera un popup:

```
for_window [class="Emacs" title="CAPTURE"] floating enable
for_window [class="Emacs" title="CAPTURE"] resize set 1880 480
for_window [class="Emacs" title="CAPTURE"] move position center
```

Mostrando los links en el navegador

Para que además los links se muestren en el navegador, pueden usarse también **hooks** que se activen al guardado para exportar a HTML. En este post de [@rafaelleru](#) se explica el proceso en detalle.

Un detalle sin solucionar

En la plantilla de captura, me gustaría poder usar además `%~g`, que crea un pequeño diálogo donde insertar tags al link que se está guardando. El problema con esto es que, hasta que ese diálogo no ha terminado, no se ejecutan los hooks y el frame no se hace único (ni se reposiciona en i3).