

## TODO

Las listas TODO simplemente son tareas que hay que hacer ( `to-do`, se pueden usar con fechas de entrega) y sirven para ser más productivos. Org nos permite crear items TODO mientras escribimos notas.

Cualquier nivel se convierte en un item de `TODO`, si empieza con la palabra `TODO`:

```
*** TODO Escribir reporte de la práctica.
```

## Comandos TODO

`C-c C-t` Rota el estado de el ítem (TODO-DONE- (desmarcado)<sup>4</sup>)

`C-c / t` Este comando ya lo vimos antes, solo estamos agregando la opción `t`, con la cual podemos buscar todos los `TODO` que tengamos en el documeto.

`S-M-RET` Inserta un nuevo `TODO` debajo del actual.

## Extras

Org nos permite hacer un montón de cosas más que no cabrían en este tríptico, pero aquí abajo enlistamos otras que seguro te ayuden sin entrar tanto en detalle.

## Exportar

GitHub y GitLab nos permiten visualizar un documento org, pero si por alguna razón quisieras tener un documento de  $\LaTeX$ , org nos permite crearlo de la siguiente manera:

`C-c C-e` Despliega un menú con las opciones a las cuales podemos exportar. Por ejemplo, la opción `l`, nos permite exportar a  $\LaTeX$ , dentro de estas opciones, la opción `p`, nos crea un archivo `.tex` y el resultado en `.pdf`. Otra opción es `h`, que nos permite exportar a `HTML`, y

<sup>4</sup>Se pueden agregar más apartados, pero es tu tarea ver cómo

dentro de estas opciones, `h`, nos crea un archivo `.html`, listo para abrir en un navegador. Hay más opciones para exportar, pero a ti te toca probar esas.

## Código

Org nos permite escribir código en el mismo archivo y poner los resultados a un lado. Para algunos lenguajes tendremos que editar nuestra configuración de Emacs<sup>5</sup>. Por lo que para el ejemplo, usaremos el lenguaje `emacs-lisp` y nuestro código solo será una multiplicación.

```
#+NAME: multiplicaPorDos
#+BEGIN_SRC emacs-lisp
(* 2 3)
#+END_SRC
```

Aquí, `#NAME:` es el nombre de nuestra función. `#BEGIN_SRC` y `#END_SRC` comienza y termina el bloque de código respectivamente. `emacs-lisp` es el nombre del lenguaje a utilizar. `(* 2 3)` es el código que queremos ejecutar.

Para ver los resultados, usamos uno de los siguientes comandos: `C-c C-c` o `C-c C-v e` y obtendremos un resultado en el mismo archivo como sigue:

```
#+RESULTS: multiplicaPorDos
: 6
```

## Vínculos

Podemos agregar un vínculo a alguna página de la siguiente manera:

`[[LINK] [DESCRIPTION]]`, esto nos permite 'Acortar el link', y que se vea legible. También, podemos hacer vínculos entre nuestros archivos, por ejemplo

`[./practica1.tex] [práctica1]]`, nos permite agregar un vínculo al archivo `practica1.tex`, y podríamos abrir desde org tal archivo. Acepta rutas relativas y absolutas.

<sup>5</sup>Puedes encontrar información sobre esto googleando 'org-babel'

# Universidad Nacional Autónoma De México

## Facultad De Ciencias

### Propedéutico

#### org-mode

Org es un modo para hacer notas, tener listas TODO y planear proyectos. Pero también es un sistema de creación y publicación, en que podemos trabajar con código.

Org está disponible para varios editores de texto, pero comenzó como un modo de edición para el editor de texto Emacs, por lo que los comandos empleados serán del editor de texto Emacs.

## Instalación

Las versiones recientes de emacs tienen integrado el modo org. Por lo que basta con crear un archivo con terminación `.org`: `C-x C-f ejemplo.org`

## Documentos

Org permite crear documentos con una estructura jerárquica y nos permite esconder o 'doblar' niveles para tener un documento más organizado, esto es con `M-x org-cycle`, que es análogo a `TAB` en el nivel que se quiera doblar.

Los niveles se representan de la siguiente manera: Y se puede escribir texto entre cada nivel.

```
* Primer nivel
** Segundo nivel
TEXT0
*** Tercer nivel
MÁS TEXT0
* Otro primer nivel
```

`TAB` Es la tecla de tabulador.

## Navegación

<code>C-c C-n</code>	Nivel siguiente.
<code>C-c C-p</code>	Nivel anterior.
<code>C-c C-f</code>	Título siguiente del mismo nivel.
<code>C-c C-b</code>	Título anterior del mismo nivel.
<code>C-c C-u</code>	Regresar al nivel más alto.

## Edición de estructura

<code>M-RET</code>	Inserta un nuevo título con el mismo nivel que el título actual; Si se hace a mitad de una línea, se parte la línea, y el resto se vuelve un nuevo título.
<code>M-S-RET</code>	<sup>1</sup> Inserta una entrada de TODO con el mismo nivel que el título actual.
<code>M-LEFT</code>	Incrementa el nivel en el título actual.
<code>M-RIGHT</code>	Decrementa el nivel en el título actual.
<code>M-UP</code>	Mueve una sección entera con la sección anterior del mismo nivel.
<code>M-DOWN</code>	Mueve una sección entera con la sección posterior del mismo nivel.
<code>TAB</code>	En una entrada sin título, cambia entre niveles que tienen sentido.
<code>C-c /</code>	Nos permite elegir algunas opciones para buscar en nuestro documento, las resalta y solo 'abre' los niveles que contienen lo que se busca. Para quitar el resaltado basta con <code>C-c C-c</code> .

<sup>1</sup>La S, se refiere a la tecla Shift  
TODO se verá más adelante.  
RET se refiere a la tecla de enter.  
LEFT/RIGHT/UP/DOWN Son las teclas de las flechas.

## Listas

Aunque se pueden usar los niveles como listas, org permite la creación de listas dentro de algún nivel.

- Desordenadas: Los items comienzan con '-', '+' o '\*'.
- Ordenadas: Los items comienzan con '1.' o '1'
- Descriptivas: Se usa '::' para separar el término de la descripción.

### Ejemplo

```
* Vacaciones
  Lugares a visitar (En ese orden):
  1. Chapultepec
  2. Coyoacan
    + La Casa Azul
    + Teatro La Capilla
  3. Centro
    Cosas importantes a realizar:
    - comer :: Salón Luz
    - tomar :: Bar La Ópera
```

### Comandos en listas

<code>TAB</code>	Los items se pueden doblar, como con los niveles.
<code>M-RET</code>	Inserta un nuevo item en el nivel actual.
<code>M-S-RET</code>	Inserta un item con <i>checkbox</i> . Para cambiar el estado de la <i>checkbox</i> , basta con <code>C-c C-c</code> .
<code>C-C -</code>	Entra en un ciclo para cambiar la forma de enumerar la lista. El ciclo es: ('-', '+', '*', '1.', '1').

Al igual que en los niveles, en las listas se puede intercambiar elementos de arriba y abajo. O subir y bajar la prioridad de los items.

Para los items los '', no son necesarios.

## Tablas

Org nos permite crear tablas en ASCII. Cualquier línea que sin contar espacios en blanco comience con '|' como el primer caracter se considera parte de una tabla. También se usa '|' para separar las columnas.

### Ejemplo

```
| Nombre | Apellido | Cosa   | Ciudad |
|-----+-----+-----+-----|
| Zoe    | Zarazua  | zacate | Zurich  |
| Lucero | Ledesma  | lámina | Lima    |
```

Las tablas se alinean automáticamente cada vez que se presiona `TAB` (la alinea y cambia a la celda de la derecha), `RET` (la alinea y cambia a la celda de abajo), o `C-c C-c` (la alinea y se mantiene en la celda) dentro de la tabla..

### Comandos en Tablas

<code>C-c  </code>	Nos pregunta de qué tamaño queremos la tabla y crea las celdas necesarias.
<code>M-LEFT</code>	Mover la columna actual a la izquierda <sup>2</sup> .
<code>M-S-LEFT</code>	Matar la columna actual.
<code>M-S-RIGHT</code>	Insertar una nueva columna a la izquierda.
<code>M-UP</code>	Mover la fila actual para arriba <sup>3</sup> .
<code>M-S-UP</code>	Matar la fila actual.
<code>M-S-DOWN</code>	Insertar una nueva fila por arriba de la actual.
<code>C-c -</code>	Insertar una línea horizontal por debajo de la fila actual.
<code>C-c RET</code>	Insertar una línea horizontal por debajo de la fila actual y moverse a la siguiente fila.
<code>C-c ^</code>	Ordena la tabla. Crea un menú para ver las opciones de ordenamiento.

<sup>2</sup>`RIGHT` hace lo mismo pero a la derecha

<sup>3</sup>`DOWN` hace lo mismo pero para abajo