

Presentaciones Latex con Emacs

Lenin G. Falconí

2024-04-27

Outline

- 1 Configuración e Instalación en Windows
- 2 Instalación Python
- 3 Comandos básicos de Emacs
- 4 Configuración en Linux
- 5 Configuración e Instalación en MacOS
- 6 Presentaciones con Emacs
- 7 Diapositivas de Ejemplo Arq Computadores

Procedimiento I

Se requiere realizar los siguientes ajustes para que Emacs opere:

- 1 Instalar WSL o un distro de Linux: Se pueden seguir los pasos de la [Guía de Instalación](#)

- 2 Instalar Emacs:

```
sudo apt-get update  
sudo apt install emacs
```

- 3 Instalar L^AT_EX (básica o completa):

```
sudo apt install texlive-full
```

Existen otras opciones para Latex que consumen menos recursos (ver [enlace](#)):

```
sudo apt install texlive-latex-extra
```

- 4 Instalar Python

- 5 Configurar el archivo en `~/.emacs.d/init.el`

Instalación de Python I

Para instalar python se utilizará el entorno de anaconda o mamba. Existen recomendaciones en favor de mamba. La dirección del repositorio de Github se encuentra en el enlace [miniforge](#)

- 1 Abra el subsistema de Linux para Windows: [guía de instalación](#)
- 2 Descargue e instale Mamba: [Descargar ejecutable desde github](#)

```
wget colocar/direccion/archivo_sh_Linux_x86_64  
sudo chmod +x Miniforge3-Linux-x86_64  
./Miniforge3-Linux-x86_64
```

- 3 Siga los pasos que indica el instalador y reinicie el shell cerrando la aplicación wsl o cerrando el terminal si está en Ubuntu
- 4 Cree un entorno con las librerías a utilizar

Instalación de Python II

```
mamba create --name iccd332  
mamba activate iccd332  
mamba install python=3.11  
python --version
```

Trouble Shooting Mamba/MicroMamba

Puede suceder que al ejecutar el comando `mamba activate` se presente el siguiente error

`'mamba' is running as a subprocess and can't modify the parent shell.`
Thus you must initialize your shell before using `activate` and `deactivate`.

To initialize the current bash shell, run:

```
$ eval "$$(mamba shell hook --shell bash)"
```

and then activate or deactivate with:

```
$ mamba activate
```

To automatically initialize all future (bash) shells, run:

```
$ mamba shell init --shell bash --root-prefix=~/.local/share/mamba
```

If your shell was already initialized, reinitialize your shell with:

```
$ mamba shell reinit --shell bash
```

Otherwise, this may be an issue. In the meantime you can run commands. See:

```
$ mamba run --help
```

Supported shells are {bash, zsh, csh, posix, xonsh, cmd.exe, powershell, fish, ...}
critical libmamba Shell not initialized

En este caso se sugiere ejecutar el siguiente comando para arreglar

Paquetes recomendados de Instalación Python

Instalar los siguientes paquetes para disponer del shell avanzado de Ipython y el complemento **rise** para convertir un jupyter-notebook en una presentación tipo reveal.js

```
mamba install ipython  
mamba install jupyter  
pip install jupyterlab-rise
```

Librerías para Exportar Jupyter Notebook a PDF

Para exportar un cuaderno de Jupyter a PDF se requiere que el Sistema Operativo de Linux disponga de la librería Pandoc:

```
sudo apt-get install pandoc
```

También, si la instalación de \LaTeX se hizo con librerías mínimas, es necesario disponer de xelatex

```
sudo apt-get install texlive-xetex
```

Con esto la exportación del cuaderno a PDF se realiza haciendo clic en *File/Save and Export Notebook As/PDF*

Configuración Emacs I

- El comportamiento de Emacs es completamente configurable por el usuario a través del archivo `~/.emacs.d/init.el`
- El archivo `init.el` puede generar instalaciones y activaciones para emacs por lo que se sugiere reiniciar emacs para que las configuraciones surtan efecto
- Abra el archivo `init.el` en Emacs y escriba la siguiente configuración

```
;; Configure Org Babel to load Python
(org-babel-do-load-languages
 'org-babel-load-languages
 '((python . t))) ; This line activates Python
;;
(setq python-shell-interpreter "/path/to/anaconda/envs/myenv/bin/python")
```

Table: Comandos de Emacs

Comando	Acción	Comando	Acción
C-x C-s	guardar	C-x C-c	Salir Emacs
C-n	siguiente renglón	C-p	anterior renglón
C-f	avanza un carácter	C-b	atrás un carácter
M-f	avanza una palabra	M-b	atrás una palabra
C-a	ir a inicio	C-e	ir al fin
C-SPC	selecciona región	C-g	cancelar comando
C-d	Borrar un carácter	M-d	borrar palabra

Configuración en Linux

Python Se realiza directamente la descarga del Miniforge conforme a lo señalado en **Instalación Python**

Emacs Se realiza directamente los comandos en **Configuración e Instalación Windows** en el punto 3.

L^AT_EX Se realiza directamente los comandos en **Configuración e Instalación Windows** en el punto 2.

En Linux **no** es necesario instalar **WSL**

Instalación de Corrector Ortográfico

Para la corrección ortográfica se requiere de la librería aspell.
Esta se puede instalar de la siguiente manera en Linux

```
sudo apt install aspell aspell-en aspell-es
```

Configuración en MacOS (Instalar Brew) I

1 Instalar brew:

- Utilizar el comando de la página de HomeBrew
- El comando identifica la arquitectura del sistema automáticamente
- Seguir las indicaciones en pantalla

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/
```

1 Para que brew sea reconocido por el sistema operativo ejecutar:

```
echo >> /Users/delta/.zprofile  
echo 'eval "$(/usr/local/bin/brew shellenv)"' >> /Users/delta/.zprofile  
eval "$(/usr/local/bin/brew shellenv)"
```

1 Verificar que brew está funcionando en el sistema usando los comandos:

```
which brew  
brew --version
```

Instalar \LaTeX en MacOS

Usando brew proceder a la instalación completa de \LaTeX mediante el comando:

```
brew install --cask mactex
```

Instalar Emacs en MacOS

Utilizar el comando:

```
brew install --cask emacs
```

Se debe asegurar que Emacs detecte la instalación de \LaTeX

Instalar Python en Mamba en MacOS I

- 1 Instalar wget o usar el comando curl para descargar el archivo

```
brew install wget
```

- 1 Ejecutar el siguiente comando disponible en el GitHub de Miniforge:

```
wget "https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-MacOSX-arm64.pkg"
curl -L -O "https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-MacOSX-arm64.pkg"
```

- 1 Verificar la dirección de descarga del archivo
- 2 Usar `chmod +x` seguido del nombre del archivo descargado para convertirlo en ejecutable
- 3 Realice la ejecución del script de instalación
- 4 Siga los pasos en pantalla de igual manera Windows/Linux
- 5 Reinicie el terminal para observar si Mamba está instalado

Configuración de Teclado para Emacs I

Dado que el teclado de MacOS es diferente, es necesario que el archivo de configuración `~/.emacs.d/init.el` permita el uso adecuado de `Command` como `Alt` o `Meta` y desactivar el `Option` a fin de poder obtener los caracteres especiales del teclado.

- 1 Crear o editar el archivo `~/.emacs.d/init.el`
- 2 Insertar en el archivo la configuración siguiente:

```
(setq mac-command-modifier 'meta)
(setq mac-option-modifier 'none)
```
- 3 Guardar el archivo `C-x C-s`
- 4 Evaluar el buffer `M-x eval-buffer` o reiniciar Emacs

Instalar paquetes adicionales

Puede usar brew para instalar componentes adicionales como pandoc

```
brew install pandoc
```

Hacer una presentación en Emacs I

- ❶ Abrir emacs ejecutando 'emacs' desde la línea de comandos
- ❷ Genere un archivo de extensión .org
 - C-x C-f
 - presentacion.org
- ❸ Active el modo org para beamer: M-x org-beamer-mode
- ❹ Coloque el encabezado por defecto y el de beamer
 - C-c C-e # default
 - C-c C-e # beamer **Observar:** el encabezado de beamer sólo se activa si el **modo org-beamer** está habilitado
- ❺ En opciones, cambiar H:1 a H:2 para que los títulos se interpreten como Secciones de la presentación y las Subsecciones como título de la presentación

Hacer una presentación en Emacs II

- ⑥ Utilice el sistema de marcas de Emacs para escribir la presentación
- ⑦ * Sección/Título
- ⑧ ** Título Presentación/Subtítulo
- ⑨ Una palabra encerrada entre ** equivale a '**negrita**'
- ⑩ Una palabra encerrada entre / equivale a '*cursiva*'
- ⑪ En el shell, instale el diccionario de español para la corrección ortográfica

```
sudo apt install aspell aspell-es
```
- ⑫ Puede cambiar de diccionario en el buffer usando M-x ispell-change-dictionary
- ⑬ Puede activar un modo de revisión al vuelo: M-x flyspell-mode

Algunos Hacks I

- Para insertar propiedades: C-c C-x p
- Para insertar código: C-c C-,
- Para insertar código: M-x org-insert-template-structures y escoge la opción s
- Para insertar un tag: C-c C-c
- Para insertar un link: M-x org-insert-link
- Para generar el archivo pdf: org-beamer-export-to-pdf

El Computador

Un computador es un sistema complejo conformado por varios subsistemas electrónicos. Para poder estudiarlo se adopta la perspectiva de entender su organización y su arquitectura. En general un computador puede hacer las siguientes operaciones:

- procesar información
- almacenar información
- comunicarse con periféricos de entrada/salida i.e. E/S

El CPU

La arquitectura principal usada en los computadores es la Von Neumann. El componente principal del sistema es el CPU, o unidad de procesamiento central, que es un micro chip. El mismo está formado de:

- Unidad Aritmética Lógica: ALU
- Unidad de control
- Registros

A more complex slide

This slide illustrates the use of Beamer blocks. The following text, with its own headline, is displayed in a block:

Theorem (Org mode increases productivity)

- *org mode means not having to remember \LaTeX commands.*
- *it is based on ascii text which is inherently portable.*
- *Emacs!*
- $\int e^{-st} f(t) dt$



Two Columns

- this slide consists of two columns
- the first (left) column has no heading and consists of text
- second (right) column has an image and is enclosed in an **example** block

Example (Another block)



Configurar Org-Babel Python

Consiste en editar el archivo `~/.emacs.d/init.el` para que emacs tenga el comportamiento deseado. Para instrucciones referirse al slide [Configuración Emacs](#)
Este slide muestra cómo hacer una referencia a otra sección

Babel

Python code

```
import numpy as np
def greet(name):
    return f"Hello {name}"

return greet("Alice")
```

The output

Hello Alice