

Presentaciones Latex con Emacs

Lenin G. Falconí

2024-04-27

Outline

- 1 ¿Qué es Emacs?
- 2 Objetivos
- 3 Configuración e Instalación en Windows
- 4 Instalación Python
- 5 Comandos básicos de Emacs
- 6 Configuración en Linux
- 7 Configuración e Instalación en MacOS
- 8 Presentaciones con Emacs
- 9 Diapositivas de Ejemplo Arq Computadores

Un entorno de programación extensible y personalizable I

- No es simplemente un editor de texto.
- Es un **intérprete de Lisp** diseñado para la computación interactiva.
- Un entorno de trabajo que evoluciona contigo.
- "El sistema operativo dentro de tu sistema operativo".
- Filosofía: Proveer un entorno autocontenido y altamente adaptable para la manipulación de texto y datos estructurados.

Filosofía: Plataforma vs. Producto I

- **VS Code:** Un producto con decisiones predefinidas por terceros.
- **Emacs:** Una plataforma donde tú escribes el comportamiento del software.
- **Inversión:** Tu configuración evoluciona contigo por décadas; no caduca.
- **Contexto único:** "Everything in Emacs" para reducir el cambio de aplicaciones.

Es un ecosistema que puede ser personalizado a tu estilo de trabajo, con el principio *Everything in Emacs* que reduce la necesidad de cambiar de contexto entre aplicaciones.

El Corazón: Elisp (Emacs Lisp) |

- Casi todo lo que ves en Emacs está escrito en Elisp.
- Puedes evaluar código en tiempo real.
- No necesitas reiniciar el editor para aplicar cambios profundos.
- **Ejemplo:** ¿Quieres que el editor se comporte distinto al guardar un archivo? Escribe una función y conéctala a un "hook".

Conceptos Clave de Operación I

- **Búferes y Ventanas:** La unidad de trabajo es el búfer (archivo, proceso, ayuda).
- **Modalidad sin modos estrictos:** Operación directa complementada con "modos menores".
- **Comandos por teclado:** Uso intensivo de M-x y C-x para ejecutar funciones Elisp.
- **Documentación integrada:** Ayuda instantánea con C-h k (teclas) y C-h f (funciones).

Emacs en el contexto académico y de desarrollo I

- **Redacción técnica:** Modo **Org** e Integración con **LATEX** y BibTeX (RefTeX).
- **Software Libre:** Garantiza libertad de uso, estudio y modificación (GNU GPL).
- **Gestión bibliográfica:** Herramientas como org-ref para investigadores.
- Soporte nativo para decenas de lenguajes de programación.

La Joya de la Corona: Org-mode |

- Mucho más que "tomar notas": un ecosistema de productividad completo.
- Gestión: Tareas (GTD), agendas y calendarios (`org-agenda`).
- Exportación: Genera PDFs, HTML y diapositivas (como esta) desde un solo origen.
- Literate Programming: Documentos que mezclan texto, ecuaciones y código ejecutable.

¿Qué más puede hacer Emacs? |

- *Edición de código:* Completado inteligente (company-mode), verificación de sintaxis (flycheck), navegación (lsp-mode).
- *Gestión de proyectos* (projectile, magit).
- *Composición de documentos* (Org mode, AUCTeX).
- *Correo y noticias* (mu4e, gnus).
- *Agenda y calendario* (org-agenda).
- *Shell integrada* (eshell, term, vterm).
- *Navegación web* (eww).
- *Personalización total:* Desde atajos de teclado hasta la creación de nuevas aplicaciones dentro de Emacs.

Conclusión: ¿Qué es Emacs? |

- Emacs no es solo un editor;
- Su curva de aprendizaje es pronunciada, pero su retorno en productividad y control es enorme.
- **VS Code:** Gratificación instantánea, pero techo de personalización bajo.
- **Emacs:** Frustración inicial, pero techo de personalización infinito.
- Se trata de disfrutar el proceso de "construir tu propia herramienta".
- Si buscas una herramienta que simplemente funcione de salida, quédate en VS Code.

Conclusión: ¿Qué es Emacs? II

- Si buscas una herramienta que sea una **extensión de tu pensamiento** y estás dispuesto a aprender, Emacs es el destino final.

Objetivos con Emacs

- En este curso, nos enfocaremos en usar Emacs para la creación de presentaciones y artículos académicos con \LaTeX , aprovechando el modo Org como interfaz unificada. Esto ahorra el uso de la sintaxis de \LaTeX , reservándola para requerimientos especiales.
- También abordaremos el concepto de *literate programming* y cómo Org permite crear documentos que involucran: texto, código y ecuaciones matemáticas.

Procedimiento de Configuración en Windows 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 LATEX (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
```

Trouble Shooting Mamba/MicroMamba II

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 y reiniciar el shell.

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

Una vez ejecutado reinicie el shell o execute source ~/.bashrc.

Puede reiniciar saliendo de wsl con exit y luego wsl --shutdown, si se abrió sesión de wsl desde un CMD de Windows

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 L^AT_EX 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 archivo init.el puede generar instalaciones y activaciones para emacs por lo que se sugiere reiniciar emacs para que las configuraciones surtan efecto
- El comportamiento de Emacs es completamente configurable por el usuario a través del archivo `~/.emacs.d/init.el`
- El directorio `/.emacs.d/` se genera la primera vez que abre emacs. Debe crear el archivo `init.el` dentro de éste.

Configuración Emacs II

- Para que no exista conflictos en el uso de la configuración revise si existe alternativamente el archivo oculto .emacs en la raíz del sistema operativo. Para esto use el comando ls -lha desde la posición de home. Por ejemplo, debe existir únicamente el directorio .emacs.d:

```
cd  
pwd  
ls -lha | grep emacs
```

- Si existe el archivo .emacs borrarlo con rm .emacs desde la posición de home

Configuración Emacs Archivo y creación de archivo init.el I

- ① Ubíquese en la raíz de Ubuntu cd
- ② Verifique que se encuentra en la raíz de su sistema con el comando pwd. Por ejemplo:

```
cd  
pwd
```

- ③ Abra emacs desde la línea de comandos de Ubuntu(WSL)~emacs & ~
- ④ Verifique la presencia de archivos ocultos con el comando ls -la desde la raíz (i.e. /home/leningfe)
- ⑤ Si existe un archivo .emacs bórrelo usando

```
rm .emacs
```

Configuración Emacs Archivo y creación de archivo init.el II

- 6 Verifique que existe únicamente la carpeta .emacs.d/

```
cd  
pwd  
ls -lha | grep emacs  
  
/home/leningfe  
drwx----- 11 leningfe leningfe 4.0K Feb 19 18:11 .emacs.d  
-rw-r--r--  1 leningfe leningfe 157M Apr  9  2025 emacs-config-backup.tar.  
drwxr-xr-x  2 leningfe leningfe 4.0K Jul  4  2025 emacs-init-files
```

- 7 Abra Emacs con emacs & desde la raíz
- 8 Cree el archivo init.el haciendo C-x C-f
- 9 Escriba el destino en /home/<<User>>/ .emacs.d/init.el.
Sustituya con con su Usuario

Configuración Emacs Archivo y creación de archivo init.el III

- ⑩ Copie el contenido del archivo desde [Github](#). Use C-y para pegar en el archivo creado. Guarde los cambios con C-x C-s.
- ⑪ Personalice el archivo con las características personales suyas y de su máquina.
- ⑫ Puede cerrar y reiniciar Emacs para iniciar la instalación. Si no desea salir, puede desencadenar la instalación haciendo M-x eval-buffer

Comandos básicos de Emacs I

Emacs utiliza las teclas CTRL y ALT, en combinación con otras para desencadenar acciones y comportamientos. Abreviamos CTRL con C y ALT, que se conoce como META, con la letra M.

Se sugiere que revise el Tutorial Integrado en Emacs para familiarizarse con el sistema. Este tutorial está disponible en español ejecutando el comando M-x
help-with-tutorial-spec-language.

Para ejecutar un comando compuesto no es necesario mantener presionada la combinación de teclas. Usted notará en la barra inferior que cuando ha presionado una combinación, Emacs responde indicando el comando y aguardando a que complete si es que el comando requiere de adicionales. Caso contrario se ejecuta.

Comandos básicos de Emacs II

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 el punto
3.

LATEX 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
```

Habilitación de Elpy I

- Elpy es un paquete de Emacs para edición de scripts de Python. Permite completar el código de manera automática, acceder a documentación, etc.
- Su auto configuración inicia si creamos un archivo .py. Al hacerlo pedirá instalar dependencias necesarias. Esta pregunta aparecerá en la parte inferior en la barra de estado de Emacs y debe contestarla afirmativamente con una y.
- Para revisar el estado de la configuración de Elpy ejecute `~M-x`

Habilitación de Elpy II

elpy-config~. Este comando abrirá un buffer en el que indicará el entorno virtual utilizado, las versiones de python y también si hace falta instalar alguna paquetería adicional.

- En el archivo de configuración init.el es importante definir el path a la carpeta de entornos de **Anaconda**. Para ver cuál es la carpeta de los entornos ejecute el comando mamba info --envs:

```
mamba env list
```

Name	Active	Path
base		/home/leningfe/miniforge3
tfmenv	*	/home/leningfe/miniforge3/envs/tfmenv

Habilitación de Elpy III

En el archivo `init.el` debe personalizar con su dirección respectiva agregando por ejemplo

```
(setenv "WORKON_HOME" "/home/leningfe/miniforge3/envs/")
```

Reinicie Emacs y luego ejecute `M-x pyvenv-workon` y seleccione el entorno de **Anaconda** que se usará para trabajar con Python. Si ejecuta de nuevo el diagnóstico de Elpy, notará que ya no está con `None` en la declaración de `~Virtualenv~/`
Instale la paquetería sugerida en el diagnóstico.

Habilitación de Modelos Grandes de Lenguaje o LLM I

- ① Disponer de una cuenta en GitHub. La configuración del archivo `init.el` hace uso por defecto de CoPilot
- ② Instalar `wslview` a fin de que en el primer uso del Chat se pueda habilitar el uso de CoPilot en el dispositivo: `sudo apt install wslu`
- ③ Instalar el paquete para modo de MarkDown. Los archivos de chat con las LLM se guardan por defecto en formato MarkDown. Sin embargo, si existiera problema, se pueden modificar para que se guarden en archivo `.org`. Para esto puede ejecutar:
 - ① `M-x package-install RET`
 - ② busca el paquete `markdown-mode`

Utilización de Modelos Grandes de Lenguaje o LLM |

El paquete que realiza la conexión entre Emacs y los diferentes LLMs es GPTEL. Para iniciar un chat y configurarlo puede ejecutar: M-x gptel-menu. Se abrirá un menú en donde podrá configurar el funcionamiento del sistema, cambiar de LLM y configurar desde donde se envía el prompt y dónde se obtiene la respuesta. Una sesión de chat se puede grabar C-x C-s y se puede restablecer abriendo el archivo y activando M-x gptel-mode.

Configuración en MacOS (Instalar Brew) |

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/
```

② 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)"
```

Configuración en MacOS (Instalar Brew) II

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

```
which brew
```

```
brew --version
```

Instalar L^AT_EXen MacOS

Usando brew proceder a la instalación completa de L^AT_EX 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 L^AT_EX

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/Miniforge_latest_darwin-arm64.sh"
curl -L -O "https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge_latest_darwin-arm64.sh"
```

- 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.

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

```
(setq mac-command-modifier 'meta)
(setq mac-option-modifier 'none)
```

- ➌ Guardar el archivo C-x C-s
- ➍ 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 |

- ➊ Abrir emacs ejecuntando ‘emacs’ desde la línea de comandos
- ➋ Genere un archivo de extensión .org
 - C-x C-f
 - presentacion.org
- ➌ Active el modo org para beamer: M-x org-beamer-mode
- ➍ 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
- ➎ 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 L^AT_EX 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