

Presentaciones Latex con Emacs

Lenin G. Falconí

2024-04-27

Outline

- 1 Configuración e Instalación
- 2 Emacs
- 3 Arquitectura de Computadores ICCD332

Procedimiento I

Se requiere realizar los siguientes ajustes para que Emacs opere:

❶ Instalar Emacs:

```
sudo apt update  
sudo apt install emacs
```

❷ Instalar \LaTeX (básica o completa):

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

❸ Instalar Python

❹ 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.

- 1 Abra el subsistema de Linux para Windows:[guía de instalación](#)

- 2 Instale el paquete bzip2

```
sudo apt-get install bzip2
```

- 3 Descargue e instale Mamba:[Descargar de github](#)

```
wget colocar/direccion/archivo_sh_Linux_x86_64
```

```
sudo chmod +x Miniforge3-Linux-x86_64
```

```
./Miniforge3-Linux-x86_64
```

- 4 Siga los pasos que indica el instalador y reinicie el shell cerrando la aplicación wsl o cerrando el terminal si está en Ubuntu

- 5 Cree un entorno con las librerías a utilizar

```
mamba create --name iccd332
```

```
mamba activate iccd332
```

```
mamba install python
```

```
python --version
```

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")
```

Lista de Commands comunes de Emacs

Poner una lista o una imagen

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: M-x org-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** está habilitado
- ❺ Instale el diccionario de español para la corrección ortográfica

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

Algunos Hacks

- Para insertar propiedades: C-c C-x p
- Para insertar código: C-c C-,
- Para insertar un tag: C-c C-c

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