

LLM Apps

Recommendation: use virtual environments

What are virtual environments?

- Using virtual environments allows us to encapsulate the versions of Python and the modules we use.

Why is it advisable to use them?

- It prevents:
 - Us from "contaminating" ourselves with the version of Python or other globally installed modules.
 - Us from "contaminating" the global environment with our version of Python or the modules we are going to install.

It's not necessary, but it's advisable

- It can save you headaches:
 - Having an old version of Python installed on your computer.
 - Having other projects that use older versions of the modules.

If you're already familiar with virtual environments

- If you are already familiar with Python virtual environments or already have another system installed to manage this, such as Anaconda or Miniconda, you can ignore these tips.

How can we work with virtual environments?

- By installing pyenv.
- And creating a virtual environment for each of our projects.

Caution

- If you are using an Anaconda environment and want to use pydev, deactivate the Anaconda environment first.
 - `conda deactivate`

To install pyenv

- In summary, on MacOS we have to:
 - have homebrew installed
 - `brew —version`
 - have the required dependencies installed
 - `brew install openssl readline sqlite3 xz`
 - install pydev
 - `brew install pyenv`
 - update the configuration file
 - ask ChatGPT for instructions
 - open the `.bashrc` or `.zshrc` file
 - copy the indicated text
 - save and exit
 - restart terminal
 - install the versions of Python we want
 - `pyenv install 3.11.4`
 - restart terminal

Install pyenv-virtualenv

- `brew install pyenv-virtualenv`
- `pyenv virtualenv-init`
- restart terminal

Create a virtual environment for each project

- Create the project directory
- Create a venv with pyenv indicating the version of Python we want
 - `pyenv virtualenv 3.11.4 environmentname`
- Activate the venv with pyenv
 - `pyenv activate environmentname`
- Install the modules we want
 - `pip install langchain`
 - `pip install jupyter`
 - `pip install python-dotenv`

You can use the same virtual environment for the course

- We will use the same versions of Python and the modules.
- Being in a virtual environment, you don't "contaminate" it with other elements on your computer.
- From now on, it's good practice to start a new virtual environment for each project.