

# **Essential software and tools I wish someone had told me about when I started my PhD**

**Ángel Javier Omella**

**University of the Basque Country (UPV/EHU), Bilbao, Spain**

**19<sup>th</sup> – 24<sup>th</sup> January 2025, Chile**

**COURSE: Introduction to coding neural networks and partial differential equations**



Universidad  
del País Vasco

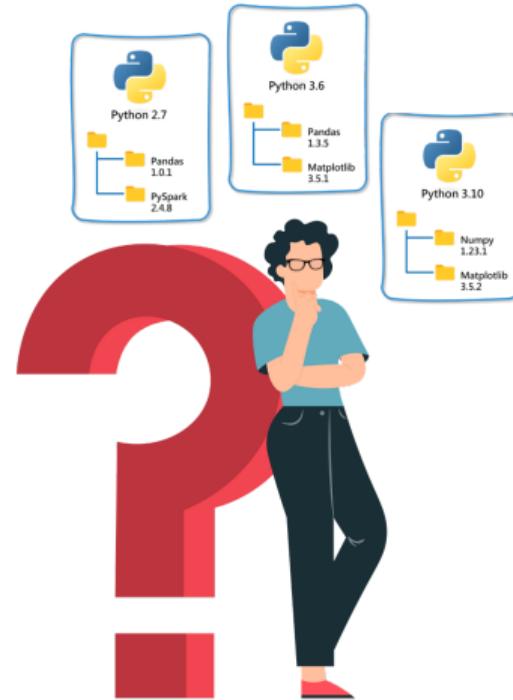
Euskal Herriko  
Unibertsitatea

# Outline

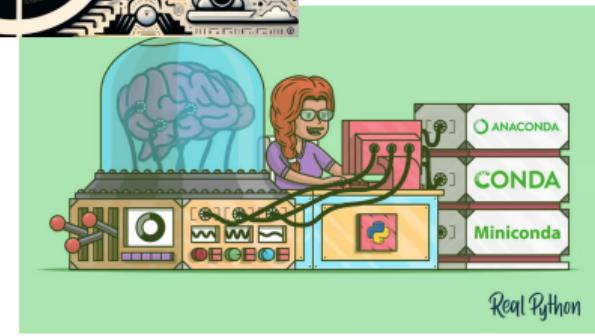
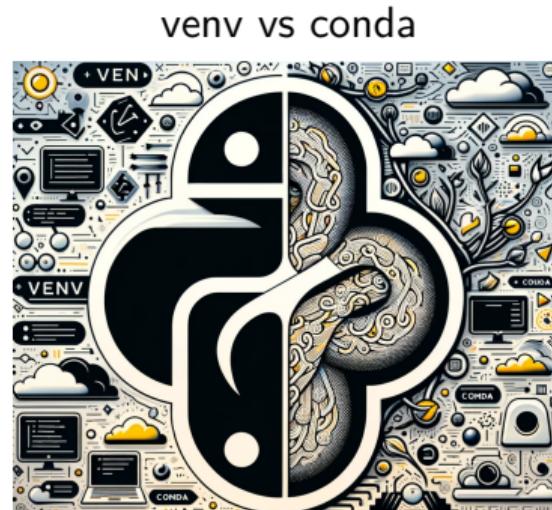
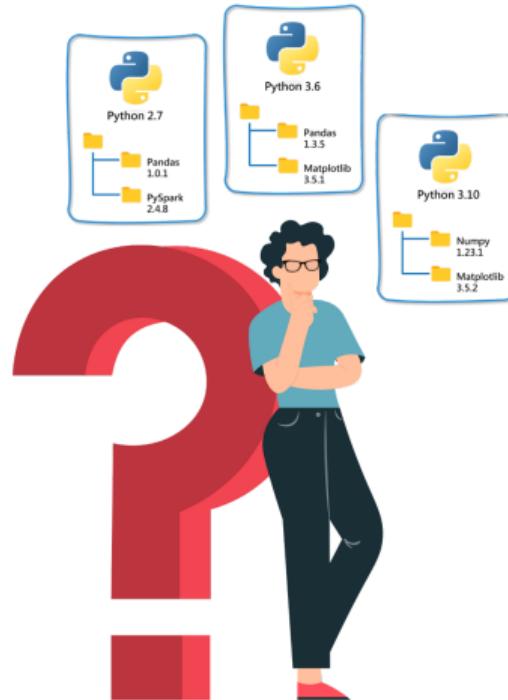
- 1** Environment and Dependency Management
- 2** Version Control
- 3** Development Environments
- 4** Tools for Deep Learning
- 5** Efficient Configuration Management
- 6** File Transfer Tools
- 7** Conclusions and Questions



# Environment and Dependency Management



# Environment and Dependency Management



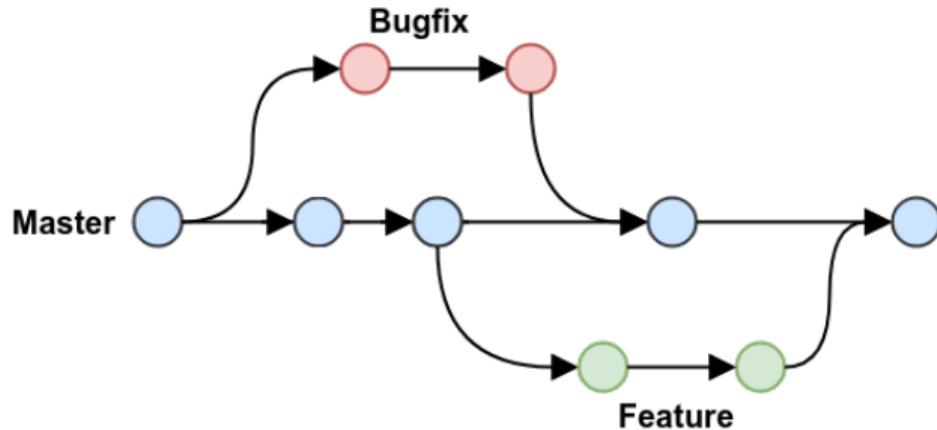
# Version Control



# Version Control



# GitHub for version Control on the cloud



```
git status
```

```
git log -3
```

```
git clone
```

```
git add "file.py"
```

git branch Feature

git pull

```
git commit -m "comment"
```

```
git checkout main  
git merge Feature
```

```
git push
```

# Development Environments



# Development Environments



The screenshot shows a Python development environment with the following details:

- File Explorer:** Shows a directory structure for a project named "differentiable-delaunay-triangulation". The files include:
  - \_\_init\_\_.py
  - DelaunayD.py
  - fix.py
  - losses.py
  - main.py
  - model.py
  - optimizer.py
  - parameters.py
  - segments.py
  - update\_boundary.py
  - vertices.py
  - weights.py
  - jax\_update\_boundary\_extensions.py
  - pytriggle\_examples.py
  - results.py
  - src/
  - triangle\_min.py
  - LICENSE
  - main.ipynb
  - pythainn\_backend\_jax.py
  - README.md
- Code Editor:** Displays Python code for the `fix` module. The code implements a differentiable delaunay triangulation using JAX and Keras. It includes functions for updating boundary vertices, computing output shapes, and calling the function with inputs.
- Terminal:** Shows a Jupyter notebook session with several cells running. One cell contains the command `!jupyter nbconvert --to html`. Another cell shows the output of `keras.ops.abs(a) + 1E-12` for a in `self.x_list`.
- Variable Inspector:** Shows variables and their types in the current scope. Variables include:
  - dt: str
  - fix\_segments\_boundary: Array of int64
  - fix\_vertices: Array of float64
  - Inputs: list
  - l: list

# Tools for Deep Learning: Keras and its Backends



# Keras



# Tools for Deep Learning: Keras and its Backends



# Keras



`@torch.jit.script`



`@jax.jit`



`@tf.function(jit_compile = True)`

# Hydra for Efficient Configuration Management



- Dynamic Configurations: Quickly change parameters in an organized manner
- Experimentation: Ideal for handling multiple experiment runs with different configurations

A simple example of use developed by Tomas Teijeiro:  
<https://github.com/Mathmode/hydra-examples>

# File Transfer Tools

```
% secure copy  
scp -r <user>@<host>:/path/directory /path/local/directory  
  
% sincronize archives  
rsync -a <user>@<host>:/path/directory /path/local/directory
```



A screenshot of the FileZilla graphical user interface. The interface has two main panes: 'Sistema local' (Local System) on the left and 'Sistema remoto' (Remote System) on the right. The local system pane shows a directory structure with files like 'beamforming.pdf', 'GettingStarted.mlx', 'Makefile', 'PULSE\_INVERSION.pdf', 'out\_2.h5', 'out\_3.h5', 'example.h5', 'out\_1.h5', 'example\_input.h5', 'kwave\_input\_data.h5', 'input\_dela\_120\_120\_120\_hex\_cuboi...', 'ioncosciMeeting-100322.pdf', and 'zip-2.1.1.tar.gz'. The remote system pane shows a similar directory structure with files like 'keras', 'app', 'code\_server', 'kwave-main', 'kwave', and 'miniconda3'. The bottom status bar indicates '1 archivo y 2 directorios. Tamaño total: 8.3 MB' (1 file and 2 directories. Total size: 8.3 MB).

# Conclusions

- Conda or venv are useful tools to manage different packages and versions
- You have to use a control version in the cloud:
  - keep your code save
  - collaborate with other researchers
  - share your code with the community
- Use Keras
  - encode to run at the end with XLA compilation
  - use and read the documentation every day
  - read code from others
  - be updated

