# Multi-Agent Systems

Jordi Pascual – jordi.pascual@urv.cat

## Presentation of the labs

MESIIA – Master's Degree in Computer Security Engineering and Artificial Intelligence
MAI - Master's Degree in Artificial Intelligence

# Outline

1. Course labs objective
2. URV Virtual Campus
3. Course timing and evaluation
4. Software requirements

Design and develop a Multi-Agent System using the CrewAI framework

# 2. URV Virtual Campus

- All the material for the course will be available at the **URV Virtual Campus** (https://campusvirtual.urv.cat/)

- In the **Practical part** section, you will find all the materials related to the practical part of the course

# 3. Course timing and evaluation

| | | Lab/Practise | Comments |
|---|---|---|---|
| 1 | 25/09/2024 | Presentation of the lab sessions | |
| 2 | 2/10/2024 | Defining agents and tasks | |
| 3 | 9/10/2024 | Using tools | |
| 4 | 16/10/2024 | Creating custom tools | |
| 5 | 23/10/2024 | Leveraging tasks | |
| 6 | 30/10/2024 | Agent collaboration I | |
| 7 | 6/11/2024 | Task 1 presentations | 10% |
| 8 | 13/11/2024 | Agent collaboration II | |
| 9 | 20/11/2024 | Organising and training complex MAS projects | |
| 10 | 27/11/2024 | Pipelines and routers | |
| 11 | 4/12/2024 | Work session | |
| 12 | 11/12/2024 | Task 2 presentation | 10% |
| 13 | 18/12/2024 | Work session | |
| 14 | 8/01/2025 | Work session | |
| 15 | 15/01/2025 | Practical work presentation | 35% |

# 4. Software requirements

You must install the following software:

- Python environment management: conda / miniconda / mamba / virtualenv / …

- Ollama to execute AI models locally

- Visual Studio Code as the IDE

- Git as the source control system

# 4.1. Python environment

- The use of a Python virtual environment / package management software is highly recommended

- You can choose **any** of them. The slides will show the examples using **conda** / **mamba** (**conda** and **mamba** commands are fully interchangeable)

1. Create a new environment with name **mas** (you can choose any other environment name) with **python 3.12**:

   **conda create -n mas python=3.12**

   Confirm changes: [Y/n]  *Y*

```
Summary:

Install: 25 packages

Total download: 37MB
_____

Confirm changes: [Y/n] Y
```

```
jordi@jordi-Ubuntu:~$ mamba create -n mas python=3.12
```

# 4.1. Python environment

2. Activate the new environment with name **mas** (or your chosen environment name). The environment must be activated each time you re-open the terminal:

   `conda activate mas`

   ```
   jordi@jordi-Ubuntu:~$ mamba activate mas
   (mas) jordi@jordi-Ubuntu:~$ 
   ```

3. Install CrewAI version 0.61.0 with all the tools:
   `pip install 'crewai[tools]==0.61.0'`

4. Install other required libraries:
   `pip install ollama duckduckgo-search transformers torch poetry`

   `conda install -c conda-forge osmnx`

# 4.1. Python environment

4. Install Jupyter Lab. The first labs will be provided as Notebooks. Although VS Code can also open Notebooks, there are some incompatibilities with human loops in CrewAI

```
conda install -c conda-forge jupyterlab
```

```
(mas) jordi@jordi-Ubuntu:~$ mamba install -c conda-forge jupyterlab
```

```
jupyter lab
```

```
(mas) jordi@jordi-Ubuntu:~$ jupyter lab
```
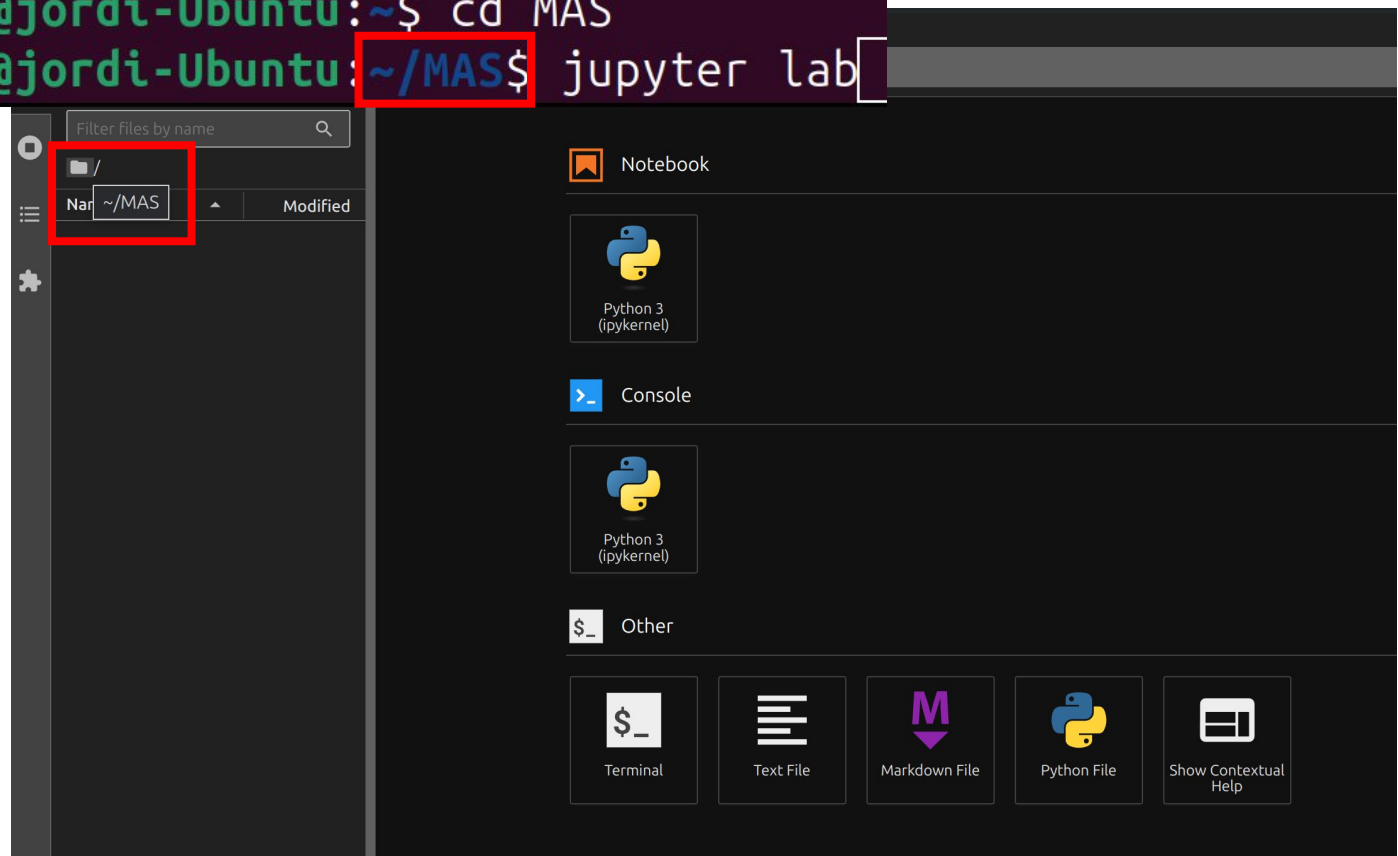
***Ctrl+C*** & ***y*** to close Jupyter Lab

```
Shut down this Jupyter server (y/[n])? y
```

# 4.1. Python environment

Jupyter Lab will open in the current working directory of the terminal

# 4.2. Ollama

- Download the LLM and embedding models. You are free to try different ones, local or API based (OpenAI, Google, etc.). We are going to use the following ones:

  1. llama3.1 8M: *ollama pull llama3.1*

  2. mxbai-embed-large: *ollama pull mxbai-embed-large*

```
jordi@jordi-Ubuntu:~$ ollama pull llama3.1
pulling manifest
pulling 8eeb52dfb3bb... 100%                          4.7 GB
pulling 73b313b5552d... 100%                          1.4 KB
pulling 0ba8f0e314b4... 100%                          12 KB
pulling 56bb8bd477a5... 100%                          96 B
pulling 1a4c3c319823... 100%                          485 B
verifying sha256 digest
writing manifest
removing unused layers
success
```

```
jordi@jordi-Ubuntu:~$ ollama pull mxbai-embed-large
pulling manifest
pulling 819c2adf5ce6... 100%                          669 MB
pulling c71d239df917... 100%                          11 KB
pulling b837481ff855... 100%                          16 B
pulling 38badd946f91... 100%                          408 B
verifying sha256 digest
writing manifest
success
```

# 4.3. Notebooks

Labs 2 - 6 will be provided as Notebooks. You can open them using Jupyter Lab (or VS Code, PyCharm, etc.). Explanations, code and exercises are provided there