# Resumo de Tecnologias e Ferramentas

## 1. Gerenciamento de Ambiente

* Conda (Miniconda): criação e ativação do ambiente pose\_cpu\_env com Python 3.10.
* Conda-Forge / PyTorch Channel: instalação de dependências de sistema (libgcc-ng, libstdcxx-ng), PyTorch CPU-only e ferramentas de build (CMake, Ninja).

## 2. Bibliotecas de Base

* PyTorch (1.13.1+cpu): framework principal de tensor e deep learning.
* MMEngine: infraestrutura de treino/inferência do OpenMMLab.
* MMCV-full (v2.1.0): extensões C++/CUDA para acelerar operações de visão.

## 3. Modelos de Pose

* MMPose (v1.3.2): biblioteca de estimativa de pose 2D/3D.
* HRNet-W48 COCO 256×192: checkpoint pré-treinado para 17 keypoints humanos top-down.
* APIs usadas: init\_model, inference\_topdown, vis\_pose\_result.

## 4. Detecção de Pessoas

* Ultralytics YOLOv8 (n-tiny): detecção rápida de 'person' em CPU.
* Workflow de crop: detectar com YOLOv8 → salvar recortes → alimentar recortes ao MMPose.

## 5. Demos e Scripts

* demo/image\_demo.py: inferência top-down simples sobre uma imagem fixa.
* demo/topdown\_demo\_with\_mmdet.py: pipeline único com RTMDet + MMPose.
* run\_yolo\_mmpose.py: script custom unificando YOLOv8 → MMPose.

## 6. Testes Realizados

* Imagens de rosto (300W/indoor\_020.png): teste inicial, keypoints faciais visíveis.
* Imagem full-body COCO (000000000785.jpg): pipeline completo de detecção → pose gerou esqueleto completo.