

# THEKER's technical challenge

This is a technical challenge for you to show your acquired skills during your professional career.

If you're completing this challenge means we think you could be a great addition to the team. Show us your worth!

Good luck 😊

## Objective

Develop a computer vision application that given an image can do these 3 tasks:

1. **Detect and identify** the objects requested by the user on the image  
*Input:* a list of strings, or "all" for all the objects  
*Output:* the image with the bounding box of the items
2. **Detect and decode the Code128 barcodes** of the requested objects and compute their **normal surface vector**  
*Input:* a list of strings, or "all" for all the objects  
*Output:* the image with the bounding box of the barcodes and the values of each barcode and the 3D arrow of the normal surface vector of each barcode
3. **Relationship between barcodes and objects**  
*Input:* either a name of an object or a barcode value  
*Output:* a barcode value or a name of an object

Examples:

- *Input:* "box". *Output:* the barcode value of the box
- *Input:* barcode value of the shoe. *Output:* "shoe"

We encourage you to create **additional functionalities** that you believe will add value to this application.

## Task Constraints

- You **must not use** ready-made barcode decoders (although you can use them as a benchmark) such as:
  - Pyzbar, Zxing, Dynamsoft, or similar libraries
- For the **detection and location** of the main objects (i.e. mug, bottle, box...) you **must not use pretrained detection models** (YOLO, Faster R-CNN, SSD, etc.) as it must be as generalistic as possible.

## Deliverables

Do not include models, weights or other heavy files. You must deliver **only** the next files in a zip file named "**NAME\_SURNAME\_SOFTWARE.zip**"

1. A **Jupyter Notebook** with **saved outputs after an execution** implementing the full system.
2. Example **input and output images**, showing the prompt and corresponding bounding box drawn on the image
3. A **README or markdown summary** explaining:
  - Your strategies for each part of the challenge
  - Which models, libraries and tools were used

- Limitations and potential improvements
- 4. Optional: a script / requirements.txt for the necessary libraries and models.

## Evaluation Criteria

- **Accuracy, consistency** and **speed** of the whole system
- Robustness with **unseen objects** and other **unexpected scenarios**
- **Clean and maintainable** code