# Computação em Larga Escala

(ano letivo 2024'25)

# Practical Assignment 2

Transitioning from Single-Threaded to Distributed-Memory Applications Using MPI

## Objective

Convert the **weather-stations** and **word-count** exercises, for which you have previously developed single-threaded solutions, into distributed-memory applications using **MPI** (Message Passing Interface). Your solution should be implemented in C++ using MPI (e.g., MPICH), and must support execution across multiple processes. Design your code to run efficiently on a multi-core or multi-node Linux/Unix cluster.

## Requirements

1. **Distributed Task Pool:**
   - Implement a task distribution strategy using MPI.
   - Each MPI process should independently process a portion of the input data.
   - Ensure that all results are gathered correctly at the master process (rank 0).
2. **Weather-Stations Output Sorting:** (Groups of 3 only)
   - Each process processes a subset of the input stations and computes local maxima.
   - Gather all local maxima at the root process.
   - Sort the consolidated results based on maximum temperature using MPI collective or local sort on the root.
3. **Validation and Performance Evaluation:**
   - Ensure output correctness compared to the single-threaded version.
   - Measure and compare execution time between single-threaded and MPI versions.
   - Calculate **speedup** and **efficiency** to evaluate scalability.

## Grading

- Development and validation of a MPI version of the `weather-stations` problem according to specification – 14 points
- Development and validation of a MPI version of the `weather-stations` and `word-count` problems according to specification – 20 points

## Deliverable

You should host your source code repository on **GitHub**. Ensure that your repository includes:

- **All source files** for both single-threaded and MPI implementations.
- A **README.md** file with setup instructions, usage details, and a summary of your approach.
- **Performance analysis** results, including execution time comparisons and speedup calculations.

## Deadline

April 16, at midnight.