# System For Pratical Evaluations of Network Administration Course

Diogo Nunes
Mestrado em Engenharia de Redes e Sistemas Informáticos
Departamento de Ciência de Computadores
2025

**Orientador**
Prof. Rui Prior, Faculdade de Ciências

U. PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas
pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ______/______/________

MSc

# Acknowledgements

Acknowledge ALL the people!

# Resumo

Este tese é sobre alguma coisa

Palavras-chave: física (keywords em português)

# Abstract

This thesis is about something, I guess.

Keywords: Computer Science

# Table of Contents

# List of Figures

# List of Abbreviations

**ANN** Artificial Neural Network. 6

**CS** Computer Science. 1

# 1. Introduction

In today’s digital age the need for qualified Computer Science (CS) professionals is growing. The CS field is vast and has many areas of expertise, one of which is network administration. It is a crucial part of any organization, as it is responsible for the maintenance and management of the organization’s network infrastructure. Proper training for network administrators is crucial for preparing them for real-world situations. One way to provide this training is through practical evaluations, allowing students to apply the knowledge they have acquired in a real-world scenario, helping them to develop the skills they will need in their future careers.

Creating a physical network environment for practical evaluations may be costly and challenging to scale for large student populations. Emulation and virtualization technologies can help to simplify and cost-effectively create practice environments for students. These technologies alone do not address the issue of manually reviewing a network topology’s setup. Manually reviewing each student’s network configuration can be time-consuming and prone to human error, rendering it challenging for their instructors. Automating the evaluation process may substantially alleviate the burden on educators and guarantee uniform and fair assessments.

## 1.1. Aims and Objectives

This dissertation continues the work of a previous student, who carried out research and first steps of development of a system for automated evaluation system for network topologies. The main goal is to design and implement a scalable system capable of automatically evaluating evaluating network topologies that make use of different vendors and device types. The support for different vendors and device types is crucial, as it allows students to practice with a variety of networking equipment, preparing them for the real-world scenarios they will face in their future careers. Automating the evaluation process will help educators dedicate more time to other tasks such as supporting students, and would also provide a more consistent and fair evaluation, eliminating the possibility of human error.

— Talk about the end goal —

The main steps of this project are as follows:

- Study the bases for the system already developed
- Requirements gathering
- Identification of the main problems that need to be solved
- Proposal of solutions for these problems
- System design
- Implementation of a prototype
- Testing with volunteers to validate the system and identify possible limitations.

# 2. Background

This chapters main focus is to provide the reader with the necessary background information to understand the context of this project. The main goal of this project is to provide a system capable of automatically evaluating network topologies by validating configurations and running tests on different devices in the network. Analogue systems exist in the market, primarily focused in programming. These systems receive code from students and subsequently run tests on it against multiples test cases and are already widely deployed in educational environments. Shifting from programming to network topologies appears simple at first glance but comes with a particular set of challenges not present in programming evaluations. Each student will require an individual network topology to work on, which can be addressed by using virtualization platforms. There is also the matter of communicating with the devices in the network, which can be addressed by using network automation tools. Finally there is the matter of combining these technologies to create a system capable of automatically evaluating network topologies.

## 2.1. Programming Evaluation Systems

While not directly related, they are the main inspiration for this project. Programming evaluation systems are widely deployed in universities and other educational institutions. These systems receive code from students and subsequently run tests on it, outputting a score and even being configurable to provide students the first test case that they failed in, guiding students to the solution without handing it out.

These tools typically provide a structured approach to test coding and problem solving skills. They begin by offering a problem statement coupled with an optional image and an example test case, normally in the form of input and expected output. Users can interact with the system by use of an online code editor, where they can write their solution and submit it for evaluation, or by uploading a file with their solution. The system then evaluates the provided solution against multiple pre-defined test cases, and validating the output agaisnt the know-good output, outputting a score based on the number of test cases passed. The system may also be configured to have time and/or memory constraints, to ensure that temporal and spatial complexity are also taken into account.

All of these, serve to provide a thorough evaluation of the student's solution, which can help guide a student to better their coding and problem solving skills.

In the context of the Department of Computer Science (DCC), Mooshak and Codex are commonly deployed to be used in the context of classes and even exams and programming contests.

## 2.2. ProxmoVE

## 2.3. Nornir

## 2.4. Python?

— Main technologies used to talk about — Python Nornir GNS3 ProxmoxVE Flask Requests -> celery -> HTTPX WSGI -> Gunicorn Linux NGINX? Gunicorn?

# References

# Appendix Title Here

Write your Appendix content here.