TÉCNICO LISBOA

# NetVM4LMS-Network Topologies for LMS

**Diogo Humberto da Silva Nunes**

Thesis to obtain the Master of Science Degree in

## Eletrical and Computer Engineering

Supervisor: Prof.João Nuno de Oliveira e Silva

### Examination Committee

Chairperson: João Luís da Costa Campos Gonçalves Sobrinho
Supervisor: Prof.João Nuno de Oliveira e Silva
Member of the Committee: Fernando Mira da Silva

**June 2023**

Declaration

I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

# Agradecimentos

Esta tese representa a conclusão de um período de aprendizagem e sacrifício, onde não só ultrapassei muitos obstáculos mas acima de tudo cresci como pessoa, como estudante e como futuro engenheiro. Todos os que me acompanharam nesta jornada foram importantes à sua maneira.

Em primeiro lugar, agradecer ao meu orientador, Prof. João Nuno de Oliveira e Silva, pelo acompanhamento e disponibilidade incansável que demonstrou ao longo do desenvolvimento desta tese.

Aos meus colegas, principalmente aqueles que estiveram comigo desde o dia que entrei no Instituto Superior Técnico pela primeira vez e que viriam a ser companheiros de aulas, de projetos, de laboratórios e de vida, agradecer pelo suporte, amizade e companheirismo.

À minha família, pelos conselhos e por terem sempre uma palavra de força e coragem.

Aos meus pais, Humberto e Maria, e à minha irmã, Susana, por todo o sacrifício, apoio e suporte incondicional. Esta conquista não é só minha, mas deles também.

Obrigado Instituto Superior Técnico por me formar.

Obrigado Lisboa por me acolher.

Obrigado a todos.

# Abstract

During the lockdown period, due to the COVID-19 pandemic, it was noticed that there were no efficient alternatives to replace the practical component carried in physical labs or to transmit concepts and perform tasks remotely. Also, most of these physical labs used virtual machines that consume a lot of resources from the students' computers and require a complex process of configuration and distribution.

To solve these problems the NetVM4LMs system was created, allowing the creation of virtual labs easily distributed through an LMS such as Moodle. This system allows the teacher to create and configure networks and subnets and launch virtual machines on them, therefore defining an execution environment that will be accessed by students to perform the exercises proposed in that lab. This execution environment is managed by a cloud infrastructure, OpenStack, allowing students to access the virtual machines through the Internet and solve the lab anytime, anywhere.

This system is therefore an effective alternative to physical laboratories since it can simulate a wide range of network topologies, while also bringing the advantages of reducing the computational resources needed by students to perform the laboratory and facilitating teachers' tasks of distribution and configuration.

# Keywords

Virtual Laboratories; LMS; OpenStack; Virtual Machines; Networks

# Resumo

Durante a época de confinamento, derivado à pandemia de COVID-19, notou-se que não existia uma alternativa eficiente para substituir a componente prática realizada em laboratórios físicos nem para transmitir conceitos e realizar tarefas de forma remota. Para além disso, a maioria desses laboratórios físicos utilizam máquinas virtuais que consomem bastantes recursos aos computadores dos alunos e necessitam de um processo complexo de configuração e distribuição.

Para resolver estes problemas foi criado o sistema NetVM4LMs que permite a criação de laboratórios virtuais facilmente distribuídos através de um LMS como o Moodle. Este sistema permite ao docente criar e configurar redes e sub-redes e lançar máquinas virtuais nas mesmas, definindo assim um ambiente de execução que será acedido pelos alunos para realizarem os exercícios propostos nesse laboratório. Esse ambiente de execução é gerido por uma infraestrutura cloud, o OpenStack, permitindo assim aos alunos acederem às maquinas virtuais através da Internet e resolverem o laboratório a qualquer altura e em qualquer lugar.

Este sistema é assim uma alternativa eficaz aos laboratórios físicos uma vez que consegue simular um vasto conjunto de topologias de rede trazendo ainda a vantagem de reduzir os recursos computacionais necessários por parte dos alunos para a realização do laboratório e facilitando aos professores as tarefas de distribuição e configuração do mesmo.

# Palavras Chave

Laboratórios Virtuais; LMS; OpenStack; Máquinas Virtuais; Redes

# Contents

# List of Figures

# List of Tables

# Acronyms

- **API** - Application Programming Interface
- **DNS** - Domain Name Service
- **LTI** - Learning Tool Interoperability
- **LMS** - Learning Management System
- **QoS** - Quality of Service
- **SFTP** - Secure File Transfer Protocol
- **SSH** - Secure Shell
- **VM** - Virtual Machine
- **VMM** - Virtual Machine Manager
- **VPN** - Virtual Private Network

# 1

# Introduction

## Contents

## 1.1 Overview

The use of technology to aid in the learning process has become increasingly common and important in the educational landscape. Over time, technological evolution has brought several developments in the area of teaching, making it simpler and more interactive. One of the most widely used tools in today's education and one that greatly facilitates organization and communication in courses are Learning Management Systems (LMS), such as Moodle, Open edX, or Canvas. These systems have been used as a digital platform allowing teachers to provide content and activities that are easily accessed by students anywhere.

Another technology widely used in teaching, more specifically in engineering, are virtual machines. A virtual machine allows the creation of simulated execution environments that can be used to perform tests and simulations in a controlled environment. These machines are able to run different operating systems and applications, without needing to have access to specific hardware, which can be very useful in a teaching environment where the availability of physical resources may be limited. These virtual machines run through a virtual machine management software, such as VMWare or VirtualBox, which allow you to create, configure and define the resources of these machines. In addition, an important advantage in its use is the ease of distribution of pre-configured environments, which reduces the time spent in configuring development environments, allowing students to concentrate on learning more advanced concepts.

There are some areas where virtual machines have proven to be very useful, for instance in the simulation of computer networks in controlled virtual environments, since they allow the creation of different network topologies, with different network devices, such as routers, switches and firewalls, allowing students to experiment with different network configurations and their effects on network performance and security. In addition, virtual machines also allow the simulation of interactions between different machines on a network, allowing the test of different interaction scenarios between systems, such as communication between servers and clients, file sharing, remote access, among others.

Virtual machines can run on a physical computer, consuming its own resources, or on a cloud infrastructure. The use of cloud infrastructures has become increasingly important in recent years, as it offers a set of advantages over traditional infrastructures, such as remote access, rapid provisioning of resources, scalability, among others. An example of cloud infrastructure is OpenStack, an open-source platform with a modular architecture, which consists of a group of services with different functions, making it highly flexible and scalable according to the needs of developers and which allows organizations create public and private clouds that support a wide variety of virtualization technologies.

In the various curricular units that make up the Electrical and Computer Engineering courses, there are several that use virtual machines and those that require various types of networks to illustrate and simulate the concepts and protocols given in the course programs, namely in the areas of Computer Networks, Distributed Systems, Security and Cryptography, among others. The virtual environments created must be secured and controlled for simulations and testing, allowing students to practice and develop their skills without the risk of damaging real systems.

## 1.2 Problem

The emergence of the COVID-19 epidemic, showed the weaknesses of current teaching methods when it became remote. The difficulty to access specific resources and the lack of communication channels and efficient ways to teach more complex and pratical concepts proved to be an obstacle when it comes to teaching topics related with networking or cryptography. In physical laboratories, some curricular units need to setup physical network environments and machines that both students and teachers might not have access when placed in a remote environment, not allowing them to perform the necessary simulations to consolidate the knowledge acquired in the curricular units. Also, that can be costly and time-consuming, particularly for smaller institutions with limited budgets and IT resources.

The virtual machines, that sometimes are used to simulate nodes across the networks, may use computational resources that not all user computers can support, compromising the learning of the students in question. In addition, making them available to students can be quite complex, and it may be necessary to create and configure the virtual machine from scratch, or, if the teacher provides an already configured image, to download it, which may take a long time.

## 1.3 Proposed Solution

The solution proposed by this work consists in developing a system that allows the teachers to define an environment to simulate a physical network laboratory. With this system, it will be possible to create networks with any topology so that the student can have an experience as similar as possible to a physical laboratory with physical networks, allowing them to practise with real-world scenarios and develop pratical skills and knowledge that will be relevant to their future careers. In those networks, virtual machines will be deployed and run in a cloud infrastructure to alleviate the resources consumed by the execution of the virtual machines in the students' computers and to facilitate the creation, configuration and distribution of each instance.

These environments should be easily shared in any LMS, allowing the authentication of each user inside the system, but also to allow the creation and grading of activities or evaluations based on that environment. The system can provide a more flexible and personalized learning experience by allowing students to access virtual machines and network environments at any time and from any location, as long as they have an internet connection. This can help to accommodate different learning styles and schedules, as well as provide opportunities for remote learning and distance education.

With the development of this project we therefore intend that:

- Faculty management and provision of virtual machines is made easier.
- Teachers are able to create environments with any network topology consisting of multiple fully customized virtual machines.
- It is possible to integrate these virtual machines with an LMS to perform exercises that were previously limited to physical labs.

## 1.4 Thesis Outline

This thesis is divided in 5 more chapters following this one. The next chapter, is a background overview, where the resources and practices currently used in network laboratories will be presented, give some examples of systems that are used in colleges across the world to implement virtual labs and introduce two technologies that were used in the developed system. Then in Chapter 3, will be presented the requeriments, functionalities and plataforms that constitute the system. In Chapter 4, will inform about the stack used to develop the system and what system roles and routes exists. The fifth chapter will be a guide for the execution of the system for both students and teachers. The sixth and last chapter, is a conclusion and will be discused future improvements for this system.

# 2

# Background

## Contents

In this section it will be discussed the ways that networking related concepts are taught in engineering courses and what technologies can support it. The first two section will focus on how that teaching is made in the current days, what tecnologies are used the simulate the network components and some emulators that are used in colleges. On the last two chapters will be discussed some tecnologies that can bring advantages and new funcionalities, improving the quality and efficiency of teaching and learning network related themes.

## 2.1 Network Laboratories

Teaching network related concepts is a very challenging task, because students have to acquire both theoretical knowledge, but also the pratical skills to configure environments that can be similar to real world scenarios. [1] To make this happen, there are some challenges to teaching and learning in an effective way:

1. Computer Networking concepts and protocols are very abstract to many students.

2. Resource provision for hands-on practical experience in Computer Networking is problematic.

There are two main components on teaching network concepts and that need to be setup or simulated on pratical laboratories, the network itself and all the required components and configurations, and the instances that are running on those networks.

Regarding instances, one of the most asked requirement is running Linux, causing a problem because the majority of the students have computers running on Window or MacOS. To overcome that, students have to install a VMM and create the needed virtual machines to do the assignment, usually following a group of instructions given by the teacher. The reason behind Linux use, is because its a very light operating system, most of Linux distributions require only around 128MB of RAM and disk space and the process of installation is much easier and require lesser user input, but the main one it is because gives great support for networking, providing a wide range of command-line tools, such as ssh or telnet, and the creation of client-server systems can be easily setup. [2]

The network where those instances will be launched can be setuped using physical hardware or using simulation softwares. In the firsts laboratories sessions, where the concepts to be transmited are still introdutory and basic, the teachers tend to use physical hardware, such as routers, switches, connections cables, among other elements, connected to specific computers also in the laboratory. These laboratories are conducted on campus and have a certain time limit, that sometimes is spent on inicial configurations of the hardware and software, and if the students did not finished the assignment, they need to come back on the next available turn, wasting more time in configurations and compromising the accomplish of the proposed tasks. [3]

To illustrate more advanced concepts, save time in configurations of hardware devices and to allow the students to conduct the lab at any time and place, teachers use software to simulate the network. These softwares are usually free and easly downloaded and allow the simulation of a wide range of network topologies, devices and tecnologies. Some examples of softwares used in teaching to simulate networks are:

- **GNS3**: Graphical network simulator that allows emulation of complex networks, with a graphical user interface that enables users to design their network topology, producing a specification text file for each network device. These network devices can interact with real world networks expading the entire topology. It has the capability of capturing packets usign Wireshark to monitor traffic of the network. [4]
- **OMNeT++**: Open source, modular and component-based event simulator tool used, in research and educational environments to simulate any system composed of devices interacting with each other. It offers a powerfull graphic interface, and tracing and debugging are easier that other simulators. [5] [6]
- **ns-3**: Project still under heavy development, that supports simulation and emulation using sockets and generates packet trace files that can be analyzed with tools like Wireshark to take out conclusions about the network traffic. [5] [6]

## 2.2 Virtualization of Network Laboratories

In all engineering courses, laboratory activities play a fundamental role in the students' learning process since they allow the consolidation of the concepts given in the theoretical components and the development of problem-solving skills in contexts close to the real ones. In areas such as computer networks, the student needs different types of hardware and software, such as computers, routers, switches, firewalls, etc, in order to be able to practice everything that is taught in the theory classes. While physical labs are ready to meet the requirements for this, more and more virtual labs are beginning to appear, which, using virtualization tools, provide the same capabilities. [7]

Virtual labs have thus appeared as a way to make labs more interactive, interesting and attractive to students, increasing their interest and perception and consequently increasing the effectiveness in their learning. In addition, there is the advantage of accessibility, since virtually they are available at any time and on any computer so that the student can practice at will, and affordability, since a physical lab is far more expensive than a virtual lab. [7] [8]

Kumamoto University in Japan developed NVLab [7], a web-based lab focused on computer networks that allowed students to practice network design and configuration, using virtualization to create the various virtual machines in the network and VNC (Virtual Network Computing) technology which allows users to remotely access resources from a range of devices in order to host and test virtual machines. This lab consisted of two tools: the Designer, which allowed you to design a network and send this information via an XML document to the server; and the Builder, which created the virtual machines corresponding to the network present in the sent XML document. With this, the students connected to the VNC module through the browser, and through an application that allowed them to control the different virtual machines in the network, configured and tested the network created. Finally, a questionnaire was given to the students who tested this system, obtaining very satisfactory results.

At the Open University of Catalonia, a system called PlanetLab@UOC [9] was developed to support the students' tasks in courses on distributed systems. This system used an existing platform, PlanetLab, which was a set of computers available to simulate and test programs, and its main functionality was to allow students to obtain real-time feedback on the developed tasks. The system's architecture was divided into three components: the student's computer, where the work developed by the student was being executed based on a template provided by the teacher; the PlanetLab network computer, where the teacher's implementation was being executed; and the faculty server, responsible for controlling the execution and evaluation process, making the comparison between the student's and the teacher's implementation, and running an application that managed all the executions done by each student.

A student at the Instituto Superior Técnico has developed the VM4LMS system [10], on which this will be based, which integrates virtual machines into an LMS to support lab classes. This system allows teachers to publish virtual machines previously configured to be run by students locally or in the cloud. Students access the lab through the course page in the LMS. This system has the ability to create, delete, run and terminate virtual machines, and it is also possible to extract files selected by teachers from the respective student machines.

## 2.3 E-Learning Support Tools

E-learning, or eletronic learning, tools are digital resources that help students and teachers access or share importante concepts or activities through any eletronic device. The implementation of e-learning technologies has as main advantage to education the flexibility that allows students to learn and practice concepts from anywhere they want and at any time, but also to provide a easy way for teachers to manage courses, share assignments or any other relevant material and to facilitate the communication between them and the students. [11]

### 2.3.1 LMS

A LMS [12] is an application that provides all the necessary tools to teachers for the teaching process. Its use thus allows the learning process by students to be done remotely. The main objective is to provide students with learning mechanisms, such as documents, programs or videos, so that they can carry out the various school activities.

In addition, an LMS should provide functionalities to make the teaching process more interactive, through the creation of forums or chats between students and/or teachers and the exchange of feedback between the two parties, and customizable, allowing teachers to monitor the progress of the various students through their work and thus adapt to the various difficulties of each one. [12] [13]

There are several LMS but we will focus on the following 4: Moodle, Canvas, Sakai and Open edX, all of them open-source.

**Moodle** was developed in PHP, which stands out for being very customizable, flexible and scalable allowing the creation of plugins, the integration of external applications and a high number of users that can have various roles and permissions. It has a notification system that sends alerts about delivery deadlines and new tasks or posts, a file management system that allows you to extract or upload files in clouds, among other features. [13]

Canvas can be used on different devices such as computers, cell phones and tablets, and even has two different applications for teachers and students. Focused on graphics and a simple interface, it has plenty of personalized tracking tools, making it possible to see the progress of a particular student and provide a leveled learning method. [13] [14]

**Sakai** was developed in Java and is oriented towards the academic sector with the goal of facilitating group projects and tasks. It has several functionalities such as chat, collaborative learning and document distribution tools, statistics, among others. [13] [15]

Finally, **Open edX** consists of two platforms: the LMS, where students can access and complete their courses; and the Studio, which allows faculty to create and manage their courses. Open edX enables active learning where the student interacts directly with the teacher. [16]

Regarding the creation of exercises and activities for assessment or to consolidate knowledge, the LMS presented have almost the same solutions. The most common are quizzes, with various types of questions, mainly multiple choice, true or false and open answer, but in some cases matching or fill-in-the-blanks questions are also possible, which automatically check the answers given, the creation of assessments with a deadline and delivery through file submission, discussion forums and chats.

### 2.3.2 LTI

Since the LMS is a mechanism to complement learning and teaching, it should be possible to integrate any external tool that is convenient for teachers and beneficial for students. For this to be possible, some LMS developed their own extensions that allowed users to navigate these external tools through hyperlinks, exchanging information between the two through communication protocols. The LTI protocol appeared as a way to integrate any external tool into any LMS. [17] It has as objectives:

- Provide a simple model for launching external tools in the LMS, consisting only of a URL, a key and a secret.
- Define a protocol for running external applications on the LMS, supporting authentication and preserving the learning context and its user roles.

The exchange of information in the LTI protocol starts when a teacher or the LMS administrator has access to an external tool, which provides its URL, key and secret. If he is an administrator, he can use that information to insert the tool into the LMS. From this point on, teachers can add the tool to their pages. When students select the tool, the LMS uses the URL, key and secret to launch the student from a frame or a new web page. The external tool receives an execution request that includes the user's identity, course, role, key and signature. This information is sent using an HTTP form generated by the user's browser and automatically submitted to the external tool using JavaScript. The information in the HTTP form is signed using the OAuth protocol. [18]

The following elements are sent from the consumer of the tool, i.e. the LMS, to the tool:

- **resource_link_id**: Unique identifier for each Tool Consumer.
- **resource_link_title**: Title of the tool. Text that appears in the link.
- **user_id**: Identifies the user.
- **user_image**: User avatar.
- **roles**: List of roles (student, teacher, etc).
- **lis _person**: Set of user related information such as name, email, etc.
- **context_id**: Identifier for the context containing the executed link.
- **context_title e context_label**: Context Title.
- **oauth_consumer_key e oauth_signature**: Used to sign and protect the information sent.

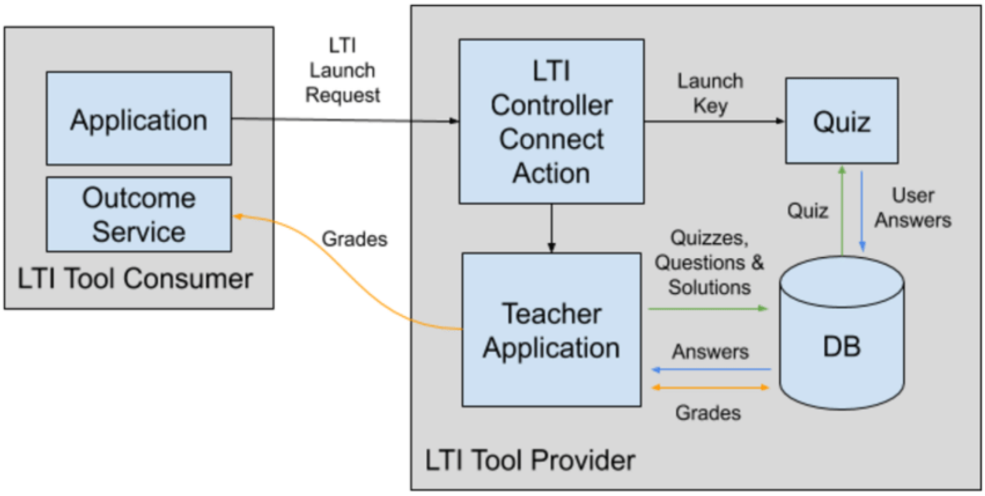


**Figure 2.1:** Example of how the LTI protocol works

## 2.4 Virtualization

Virtualization [19] is a technique that, by sharing the resources of a hardware, allows several virtual machines to run on just one machine. This virtualization can be done locally on a computer or in the cloud. In local virtualization, a virtual machine uses the resources of the computer where it is being executed, such as hardware, memory, processor, among others. The connection between the virtual machine and the physical computer is the responsibility of the Virtual Machine Manager (VMM), a software that allows a single physical computer to support multiple virtual machines. In cloud virtualization, this is accomplished through open-source software that provides an infrastructure to manage the various instances in a cloud environment.

When it comes to the execution environment, there are several types of virtualization, including desktop virtualization and server virtualization.

- **Desktop Virtualization**: It allows the creation and storage of multiple desktop instances of a user on a single host. All this is possible using a VMM that manages and allows the virtual instances to use the server's computing resources. The VMM creates the VMs that simulate the user's desktop environment, which may have different operating systems, applications and settings. [20] [21] Some examples of software that implements desktop virtualization are:
  - VMWare Workstation: VMWare's desktop virtualization tool, used to install and run multiple instances with multiple operating systems on a single computer. [22]
  - VirtualBox: Software developed by Oracle that provides the ability to create a large number of VMs and is compatible with a large number of both host and guest operating systems. [22] [23]

- **Server Virtualization**:Allows multiple virtual machines to run on a physical server, sharing the resources of that physical server across multiple independent virtual servers. The virtual machines' resources, such as CPU, memory, storage and networking, are provided through the VMM of the physical server. [20] [21] [24] Each virtual machine is independent of the others and can have different operating systems.
  - VMWare vSphere: VMWare's product for server virtualization. It consists of a VMM, ESXi, responsible for the computational resources of the multiple virtual machines, a management tool, vCenter Server, and other VMWare tools that provide various functionalities to the system. [25]
  - OpenStack: IaaS platform designed to build cloud infrastructures using Python and Unix Shell. It is scalable and has a distributed architecture that makes it highly flexible. This architecture is divided into several components in charge of different functionalities. [23] [26]
  - OpenNebula: Open-source IaaS project, developed in C++, Ruby and Java. It is very flexible, being mainly used as a private cloud, VM and network and storage manager. It has a very flexible centralized architecture, which includes three levels: Drivers, which communicates directly with the operating system, is responsible for creating and closing the virtual machines and monitors the operational state of the physical and virtual machines; Core, which is responsible for managing the VMs, including network configuration and storage management; and Tools, which provides some interfaces such as CLI, browser and libvirt API to communicate with users so they can manage their VMs. [23] [26]

### 2.4.1 Network Virtualization

Another type of virtualization is network virtualization. Network virtualization uses software to perform networking functionality that was previously performed by hardware. When using this type of virtualization, the physical network is only used for packet forwarding, while all network management is done using virtual or software-based switches. [24] In addition, it helps optimize the network in terms of data transfer speed, flexibility, throughput, and scalability, while also making it easier for the administrator to allocate and distribute resources to ensure stable network performance. [20]

As a way to assist in network virtualization, a new technology called Software Defined Networking (SDN) appeared, which allows separating the control plane, which controls the operation of network switches and routers, from the information plane, which controls data forwarding through the physical network, thus allowing the control plane to be directly programmed and centrally managed. The SDN architecture is based on the OpenFlow protocol, which is responsible for making the separation between the control plane and the information plane and for allowing an external controller to decide how the data forwarding through the switch is done. [27] [28] Some of these controllers are:

- **OpenVSwitch**: VMWare's open-source software that makes it possible to create virtual switches and configure remote controllers. [27]
- **Mininet**: Open-source software that emulates OpenFlow devices and SDN controllers, thus allowing the simulation of several SDN networks on a single machine. It offers a very simple to use, scalable and customizable service and is therefore very useful to support teaching and research. [27] [29]

Virtual machines must be able to connect to both physical and virtual networks, and to do this, VMMs such as VirtualBox and VMWare make use of various virtual network adapters that can be configured on their software interfaces. These network adapters are configured to operate in different network modes:

- **NAT**: With this mode, the physical machine and the virtual machine share the same identity, thus masking the virtual machine's network activity. This virtual machine can access external networks. [30] [31]
- **Bridged Networking**: In this mode, the virtual machine connects to a network using the physical machine's Ethernet adapter, thus allowing it to communicate with the physical machine, with other computers connected to the physical network, and with external networks, including the Internet. This mode can be used to run servers on the virtual machine that must be accessed from the physical local network. When using this mode, the virtual machine needs to have its own identity that can be generated via DHCP or by configuring the IP address and other information in the machine's operating system settings. [30] [31]
- **Host-Only Networking**: It allows the various virtual machines virtualized by the same computer, and which are connected to the host-only adapter, to communicate with each other, thus obtaining an isolated virtual network. [30] [31]

Regarding software that implements server virtualization, network virtualization involves services that implement network creation and management functions. In the case of OpenStack, there is the service Neutron, [32] the component of its architecture that implements network services using an API to create and manage network components, such as networks, routers, ports, among others. OpenStack also offers Firewall, QoS, DNS services, among others, which make the virtual network created more similar to a physical one. In the case of VMWare vSphere, there are network services such as vSphere Standard Switches, which is responsible for managing network traffic, providing network connectivity to hosts and virtual machines, and vSphere Distributed Switches that allows you to manage and monitor the network configurations of all hosts associated with a switch. [33]

# 3

# System Overview

## Contents

## 3.1 Surveys

A questionnaire was given to some IST teachers in order to collect information about several important topics for the development of this system, namely, the use of virtual machines in the curricular units, the knowledge and use of LMS and OpenStack and the degree of interest and usefulness in several functionalities that the system could implement.

### 3.1.1 Population

Twelve faculty members from different IST courses and areas answered this questionnaire in order to receive varied opinions and to understand what the needs are for each area in question.

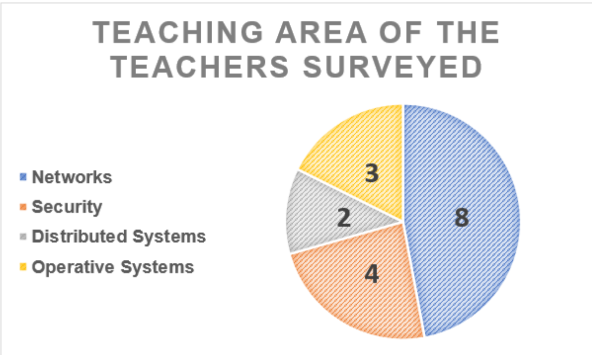


**Figure 3.1:** Distribution of teachers by teaching area

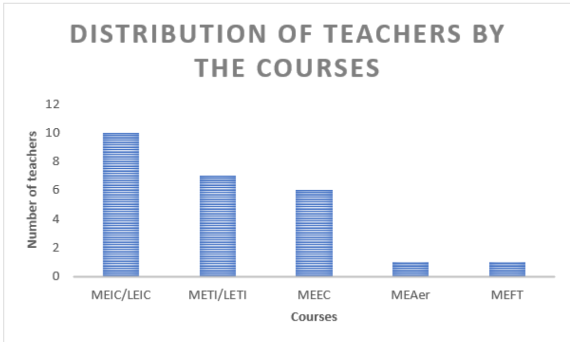


**Figure 3.2:** Distribution of teachers by the courses they teach

### 3.1.2 Questionnaire

The questionnaire consisted of six parts:

- The first part was the presentation, where the faculty members provided information about the courses and chairs they teach.
- In the second part we tried to find out whether the professors used virtual machines in the subjects they taught, and if so, for what purpose and with what configurations.
- The third part asked about knowledge and use of LMS.
- The fourth part asked about the knowledge and use of the OpenStack Cloud infrastructure.
- In the fifth part, the opinion about providing virtual machines in an LMS to support teaching was sought.

- In the sixth and last part, we tried to find out what functionalities the teachers would like to see implemented in the system to be developed.

The questionnaire consisted mainly of multiple choice and Likert scale questions (from 1-Very uninteresting to 5-Very interesting), and also had open response questions for the teachers to provide their opinions more fully and freely.

### 3.1.3 Results

In this section a summary is made of the answers given by the teachers in the different sections of the teacher survey.

In the second part, only 7 of the 12 professors provide virtual machines in the curricular units they teach. The virtual machines provided are all instantiated in VM managers and used mainly to support the laboratory classes and to perform exercises. Most of the virtual machines still need additional configurations to be performed by the students, mainly in terms of networking.

Regarding the LMS, the one that stood out the most was Moodle, with all respondents stating that they know the system and have worked with it as a teacher to provide content and to conduct summative assessments. Regarding OpenStack, most of the teachers (58%) know the cloud infrastructure, but only two have used it.

In the fifth part, although the use of VMs is not unanimous among the courses taught by the teachers, all gave positive responses regarding the integration of virtual machines in the LMS to perform exercises and demonstrate theoretical concepts, with the average response being around 4.5.

In the sixth part, all the mentioned functionalities, like Quality of Service, DNS and Firewall, presented average response values around 4.

### 3.1.4 Discussion

Through the questionnaire it was possible to understand the different uses of the virtual machines provided by the teachers and understand the degree of initial configuration of them. The machines provided are instantiated in VM managers, such as VMWare or VirtualBox, and most of the time students have to perform several configurations in order to prepare them to perform the tasks requested by the teachers. Focusing on network configurations, it is sometimes necessary to configure the network in the operating system and the network routers, that is, to define the network topology in which the virtual machine is inserted. This topology can be a single local network or several local networks connected by routers.

It was also possible to understand what kind of functionalities the teachers would like to see implemented in the system. Since the teachers were mainly from the network and security areas it was expected that the functionalities that would stand out the most were related to making available several machines on the same network and configuring the machines' firewalls. A functionality that some teachers mentioned that they would like to see implemented was a snapshot storage system, that is, images of the current state of a machine.

Regarding knowledge and use of LMS, all teachers know and have worked with Moodle in their curricular units, the opposite when it comes to OpenStack, where most teachers admit knowing the service, but very few have used it. However, when it comes to the fusion of these two tools, the responses were positive, since the teachers found the idea of making available in the LMS virtual machines already configured to perform exercises and illustrate theoretical concepts quite interesting.

In the open response questions, where teachers were able to respond more freely to the advantages and utilities of running virtual machines in the cloud and integrating them into the LMS, we obtained very useful answers regarding the strengths of the solution presented. One of the problems most mentioned and that would be overcome by using virtual machines in cloud infrastructures was the fact that sometimes students have weak computers and are therefore dependent on them to perform some laboratory exercises. With regard to the uses of virtual machines in cloud infrastructures in the teachers' curricular units, the answers were quite similar. Since most of the teachers teach courses in the areas of networks and security, the most common uses are in activities related to networks, namely network simulation and configuration and illustration of security concepts. In the area of distributed systems, the teachers also mentioned that they could use virtual machines to implement client/server and distributed applications.

## 3.2 Requirements and Functionalities

After analyzing the answers provided by the teachers surveyed, the system requirements and the functionalities to be implemented were defined. It was unanimous that the most popular LMS and the one that all surveyed have worked with is Moodle, so having a system that is optimized for this LMS is fundamental and quite beneficial for teachers. Regarding the cloud infrastructure, which will serve as a platform for the creation of networks and virtual machines, and also handles their management, most teachers do not know or have worked with OpenStack, so providing the system with interfaces that allow teachers to create virtual machines and networks according to their needs in an easy and intuitive way is very important.

The following requirements are necessary to make the system similar, but easier to configure and distribute, than a physical laboratory:

1. Allow the teachers to choose the resources (RAM, Disk, CPU) and image for each virtual machine

2. Allow the teachers to choose the network topology in which the virtual machines will be deployed

3. Launch virtual machines in the created network topology

4. Make available virtual machines running in a cloud infraestrusture

5. Allow the teachers to access and analyze student's virtual machines after the conclusion of the assignment

6. Allow the students to have access to the environment through the LMS

All of these requirements are intended to allow the teacher to create an execution environment for the chosen virtual machines that are launched on the networks also created by him. This execution environment will be made available through an LMS in conjunction with the exercises that the students will perform on it. Each student is independent within the system, and will have their own set of virtual machines from which files are extracted or snapshots taken.

Briefly, the main functionalities and objectives of the system will be

- Enable the provision of virtual machines and network-based exercises directly in an LMS.
- Make it easy for the teacher to distribute virtual machines to students, and then collect the resolutions of the proposed exercises.
- Allow access to a wide range of network configurations to perform exercises in a context close to that of a physical lab with a physical network.

## 3.3 Architecture

The system architecture is composed by several components with different functionalities and with interactions between them. The teacher has his own dashboard where he will perform the various configurations to prepare the environment where the assignments will be done by the students. His application consists of a VM manager, where the teacher creates templates for the instances that will be launched on the network, a network manager that allows the creation of networks and a task manager, responsible for defining the activities, the files to be collected from the virtual machines and the instances templates and networks that will be used on the environment.

This application interacts with a set of repositories where different tasks, VMs and network configurations are stored. In addition, there is also a user repository where information regarding each user of the system will be saved. Once the network and VMs settings have been chosen, the VM execution environment is defined, that is, the environment with the settings made by the teacher, where the proposed exercises will be executed and which will be made available through an URL on the Moodle course page. The student interacts with the task repository, with the VM execution environment, where he will have access to the consoles URLs for his virtual machines and performs the proposed exercises.

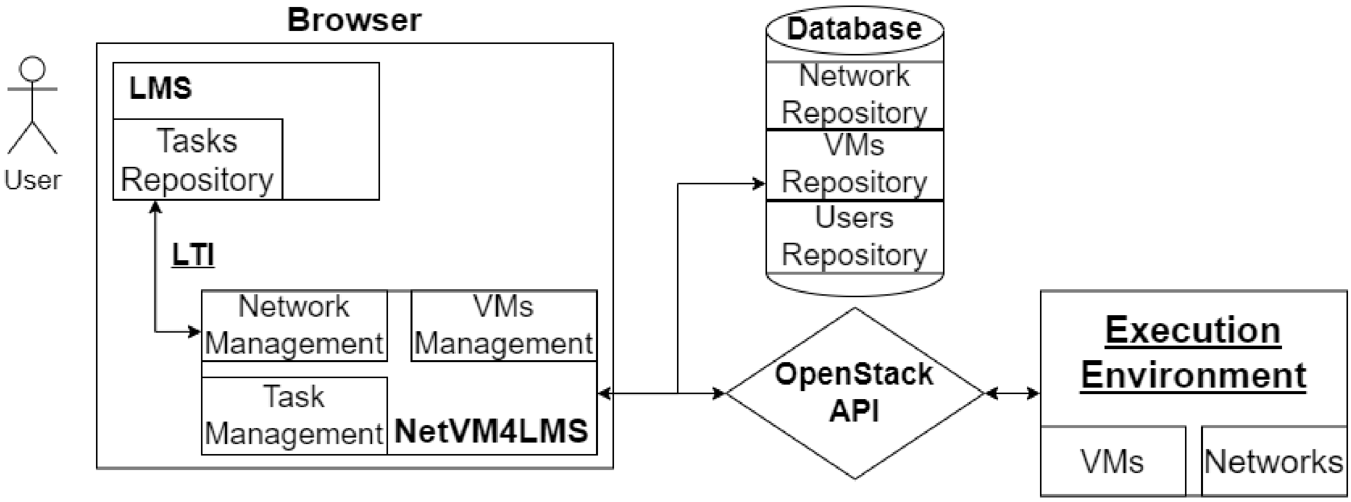


**Figure 3.3:** System Architecture Diagram

Based on the surveys we were able to define which platforms we would use as LMS and as cloud infrastructure. In terms of LMS it was unanimous, since all teachers knew and had already worked with Moodle and therefore its integration into the system was beneficial. With regard to students, and since it is in the LMS that they access the exercises proposed by teachers, I believe that most know or have worked with Moodle, and therefore will not have any difficulty in navigating the platform.

In terms of cloud infrastructure, the platform chosen was OpenStack, since it brings together all the desired functionalities for network configurations and VM management.

The implementation of the system relies on three different platforms that will play different roles in the overall project:

- **OpenStack**: Responsible for managing the VMs and the network where they will be launched and generating the execution environment where the proposed exercises will be performed for the students.
- **Moodle**: Platform accessed by the student and where will be made available the environments and assignments proposed by the teacher, managing the exercises and users.
- **NetVM4LMS**: System that was developed and that will make the connection between OpenStack and Moodle. Communication with OpenStack will be done through its API, which allows the definition of networks, creation of virtual machine images, among others, and communication with Moodle will be done through the LTI protocol.

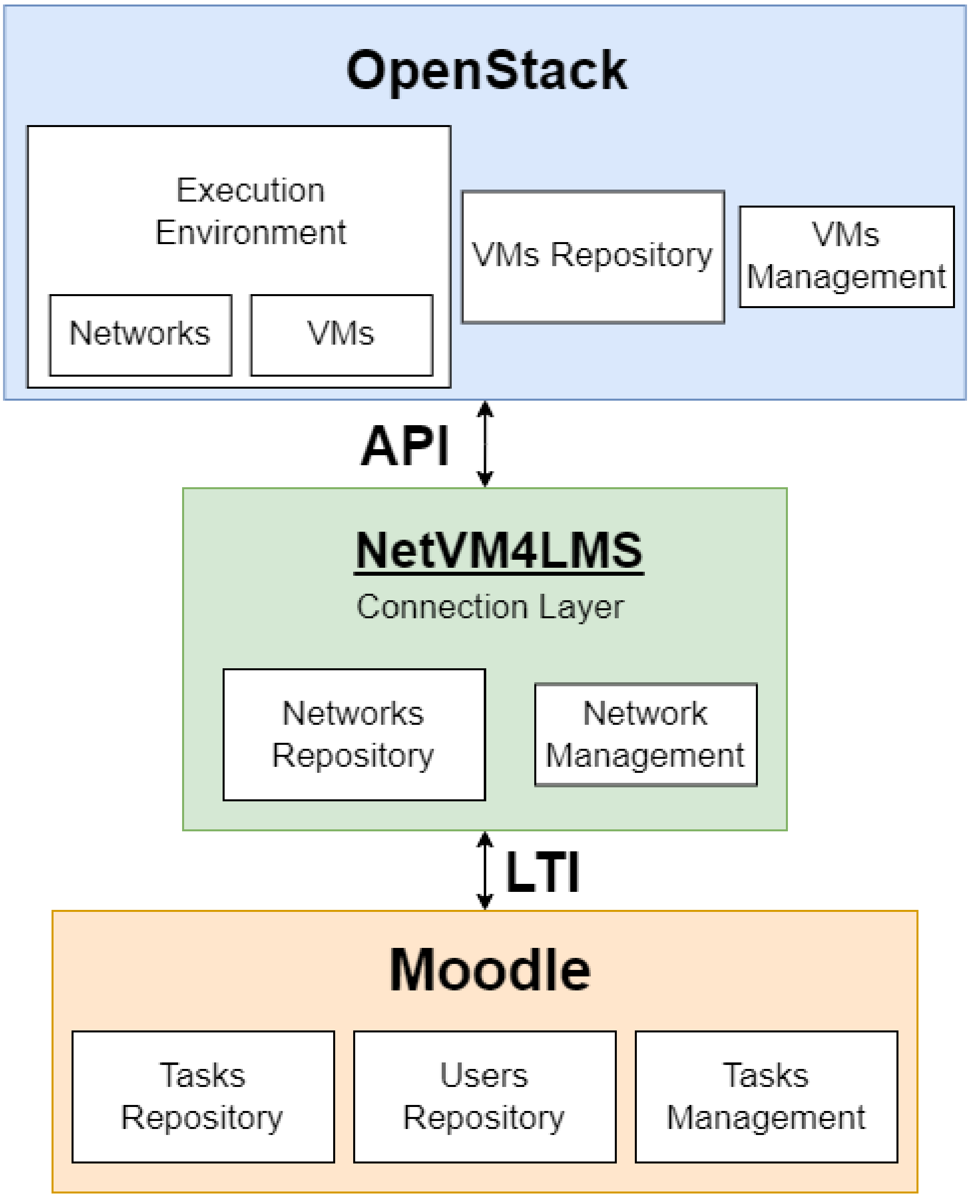


**Figure 3.4:** System Implementation Diagram

Using the LTI protocol to connect to Moodle not only simplifies the process of integrating external tools, but also provides a service for authentication and secure exchange of information.

OpenStack, through its API that facilitates its integration in the system, provides several network functionalities and services. It allows the creation of networks with any network topology, a local network or several connected networks, with or without internet connection.

## 3.4 Database Model

To support this application it was created a database to save all the necessary information for the well work of the system. This database is divided into several tables that saves information regarding an element and all the different connections between tables:

- **user**: Saves information about the users of the system. Each user has a role associated (professor, user or administrator), that is associated with the roles table by a role_id attribute.
- **roles**: Stores information related with the different roles of the system and its permissions.

Each user of the system is associated with 3 other tables:

- **instance**: Used to store information about each virtual machine of the system, like flavour and image of the instance, the url of the console, a boolean that controls the access to internet and the id inside Openstack system, used to make reference to it on the API requests made.
- **snapshot**: Save informations about all the snapshots taken by the students of the instances that were used to make the assignments.
- **user_course**: Since each user can be registered in many courses, it was created a many-to-many relationship between this two tables, being the user_course table the join table in this relationship.

Inside this system, each course has its own instances, networks and environments, and so, these are not shared among different courses.

- **env**: Table that saves the informations related with the environment created by the teacher, like name and description, the number of instances that each student will have access and a text attribute that is a JSON object with the name, instructions, template and the path of the files to be retrieved from each of those instances.
- **network**: Saves the name, the identifier from Openstack and boolean variable that controls if the network was launched or not. Each network can have many **subnetworks** that also have a table in the database.

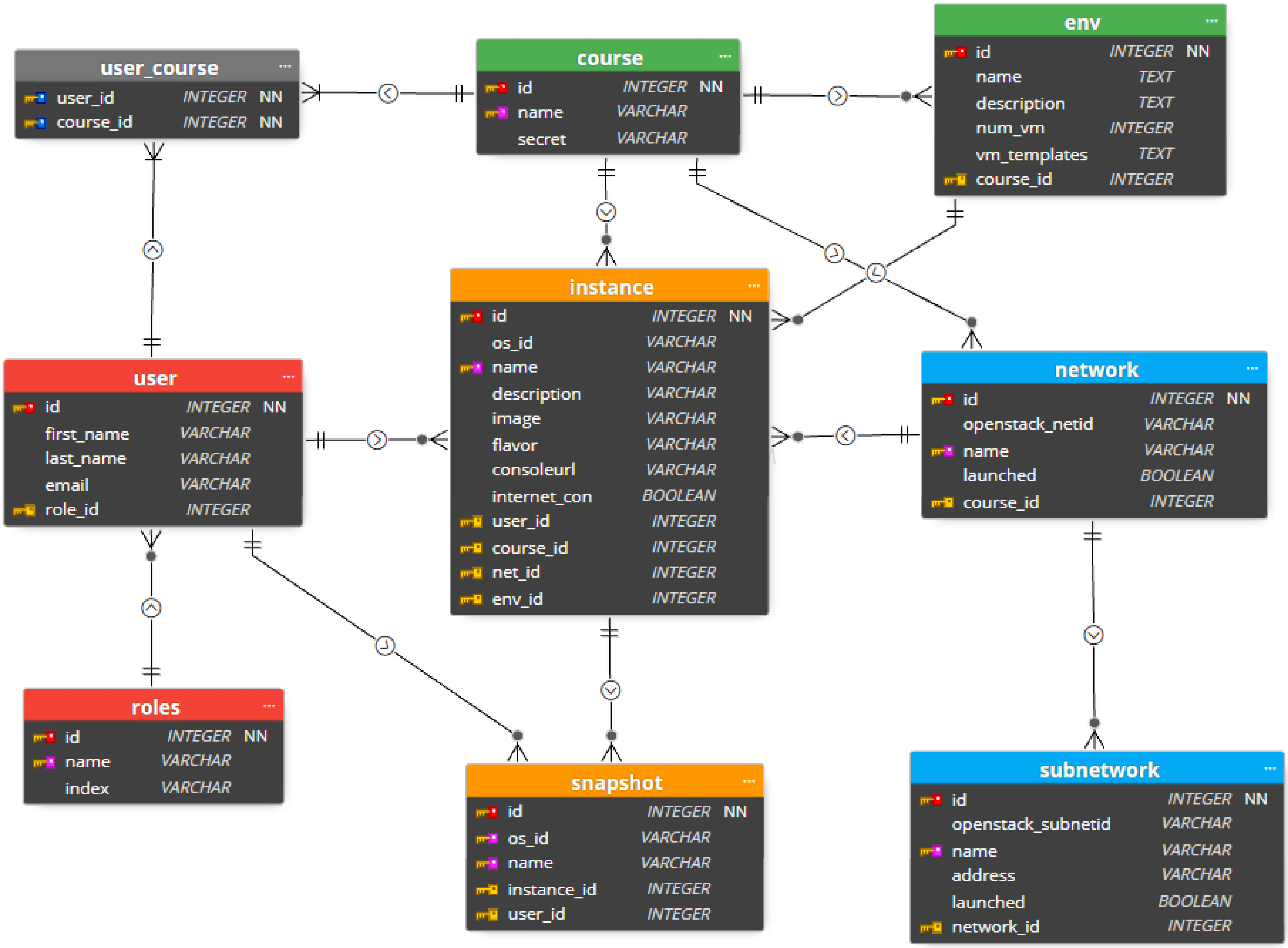


**Figure 3.5:** Database model diagram

# 4

# Implementation

## Contents

## 4.1 Tecnological Stack

In this section it will be presented the technologies and frameworks that were used on the implemention this system.

### 4.1.1 OpenStack

As said previously, OpenStack was used as cloud infrastructure to manage the networks and the instances created by the teachers and used by the students. This infrastructure is an open-source software, extended as an infrastructure-as-a-service (IaaS), that controls large pools of compute, storage and networking resources managed throughout a dashboard or using the available API. OpenStack is divided into services, each one providing specific resources to the users. [3] [35] For this project, were used three main services APIs:

- **Nova**: Main computing service of OpenStack that supports the creation of virtual machines, giving complete control of the computing resources and allowing the instances to run on OpenStack's computing environment, reducing the time required to obtain and boot new instances. Provides an API that allows the developers to create multiple servers and consoles of them, and to perform actions on those servers, like create image, associate floating IPs or add security groups. This service is aided by another service, *Glance*, that provides images to the compute instances launched. [36] [37]

- **Neutron**: Component that provides the software-defined networking services for OpenStack, allowing the creation of network topologies and the configuration of advanced network policies. With the provided API endpoints it is possible to create networks and add subnetworks to them and to create routers to connect those networks, but also offers Firewall-as-a-Service to deploy firewall groups to protect the networks and the possibility to manage Quality of Service rules, like bandwidth limit or minimum packet rate, among other services. [38]

- **Keystone**: Service that provides API client authentication and service discovery. This service generate tokens that allow users to have access to the different services APIs and allows the creation of users, groups (collections of users) and domains (collections of groups and projects). [39]

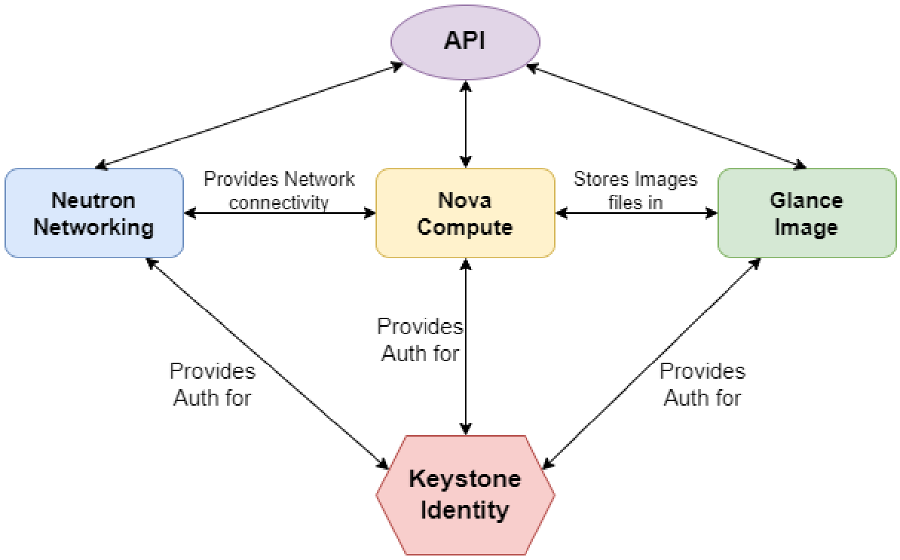


**Figure 4.1:** Openstack services relations

### 4.1.2 Python

The language used to develop NetVM4LMS was Python, a language with unique features and one of the best choices for scientists and engineers to code scientific applications. [40] [41] It has a clean syntax that make the code easy to understand and a large number of libraries that allow the creation of very complex programs. Two examples of libraries that helped implement some necessary features to the system were:

- **PyLTI**: Python implementation of the LTI protocol, which allows the integration of the web application with any LMS that supports LTI by providing an API to exchange messages directly. [42]
- **Paramiko**: This library allow the connection between the web application and the instances running in the cloud infraestructure using the SSH protocol and the retrieval of any file from that same instance through SFTP. [43]

### 4.1.3 Flask

Flask is a collection of libraries and modules that provides a simple way for developers to create a web application. It uses Werkzeg WSGI toolkit to implement requests and response objects and jinja2 as template engine which allows the use of specific data to render dynamic pages. It stands out for beign very explicit increasing code readability, scalable, meaning that the created applications have potential to grow quickly, and very flexible, allowing the projects to explore very different functionalities without compromising the main structure of the web application. [44] [45]

To complement the use of Flask framework some others libraries were used. One example is Flask-Login, that provides user session management, handling the processes of loggin in and out, saving the users' session across the navigation throughout the web application and allowing the restriction to some routes to only authenticated users. [46] Other library used was Flask-WTF, which is a Flask extension that allow the use of WTForms library to render web forms with different types of inputs, used to collect the inputs from teachers when creating the networks, instances and environments. [47]

### 4.1.4 SQLite/SQL Alchemy

As mentioned in section 3.4, it was created a database to save and retrieve data during the execution of the system. To create that database was used SQLite. SQLite is an open source embedded database written in C, that stands out for being very lightweight and easy to implement, beign the most widely deployed database in the world. It offers a relational database management system, that usually requires a separate server to operate, but in this case, it is able to access and manipulate the database without need of a separate server, storing the whole database in a single file. [48] [49]

To help access and create this database, was necessary a library to allow queries to the database and to create all the necessary tables and relations between them. To do that, was used SQLAlchemy, which is a Python library that is usually used as an Object Relational Mapper (ORM) using classes to create the database tables. It supports the most common data types, four types of relationships (One To Many, One To One, Many To One and Many To Many) and querying data is very simple, providing a group of functions through a Query API. [50]

## 4.2 User Management

The user management starts when Moodle sends a launch request to the system through a specific route, with some informations regarding the user that launched the tool. Among those informations is a list of the roles that the user has on Moodle, that is converted to one of the roles of the system, and after that, using other informations from that request, it is created an entry in the database for that user and is made the login in the system using Flask-Login.

There are two main system roles, each one of them with the respectives views and routes that allowing them to perform the actions according to their purposes. The roles of the system are:

- **User**: Represents the student and has as main objective to access the URL published in the course Moodle's page to get access to the environment created by teacher to make the assignment. Each user will be handled individually by the system and will have its own instances and snapshots.

- **Professor**: Responsible for define the different environments for the proposed assignments. Is allowed to create the networks, subnetworks and instances that will be used to define the environments for the virtual machines that students will have access to make the given assignments. It will have access to the virtual machines and extracted files of each student.

Now that all the users that access the system are identified and logged in, they need to have access to the OpenStack API to make the necessary requests. As said in section 4.1.1, the service Keystone is responsible for providing authentication tokens to allow users to have access to the services APIs, and so a request is sent to a specific endpoint, providing information regarding the domain and the respective username and password, and the response brings in the header a token that will be sent in every request made to services' APIs to prove that the user is authenticated.

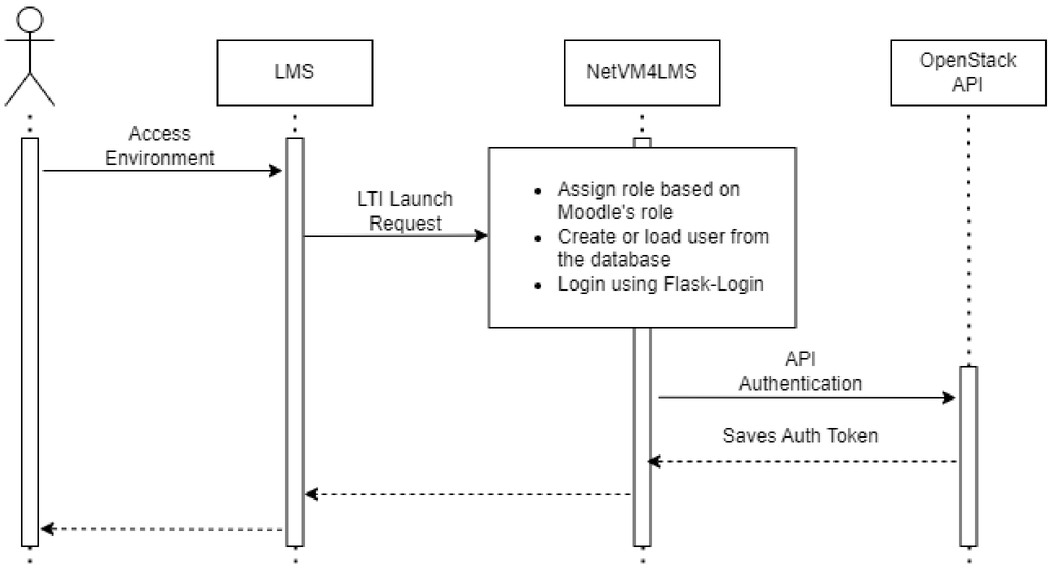


**Figure 4.2:** User authentication process

From the point of view of OpenStack, is not made any type of user management. The only management that can be done is a course management. By using Keystone service it is possible to define projects that can represent each course. Each project would have its own networks and instances, and the API authentication and requests will be handled individually in each project.

## 4.3 Routes

The system has different routes depending on role of the user and each route will either render a page through a custom template or make a request to OpenStack.

### 4.3.1 Launch Route

The launch route is the one that makes the connection between NetVM4LMS and the LMS, and allows the authentication inside the system. This route is published in the Moodle's course page that will allow the students to have access to the environment proposed for the assignment and will allow the professor to have access to the respective dashboard where all the configurations and definitions are made.

The route structure is **/launch/**<**int:task_id**> where task_id represents the identifier for the environment that will be launched. When the student or the teacher access that URL, Moodle will send a POST request with information regarding the user and the course from which the url was accessed. With that information, the system will check if the course and user already exists in the database, and if not, will created them and load the course into a environment variable and log in the user using Flask-Login.

After that, a request is sent to OpenStack Keystone service that generate tokens in exchange for authentication credentials. The response brings in the header a token that represents the authenticated identity and will be sent in the header of all the requests made for the services APIs.

At last, and if the user is a student, a request will be sent to the Nova service to generate new console URLs for the instances assigned to that student, overcoming the console URLs expiration and allowing students to always have access to their instances when accessing the proposed environments.

### 4.3.2 Professor Routes

In the following section will be presented the routes that the professor role has access. Each route has a specific objetive to either render a page to show information, an interface to collect input and create an element in the system or to perform a specific request to OpenStack. Will also be presented a diagram that represents the sequence of interactions between elements that happens during the professor execution.

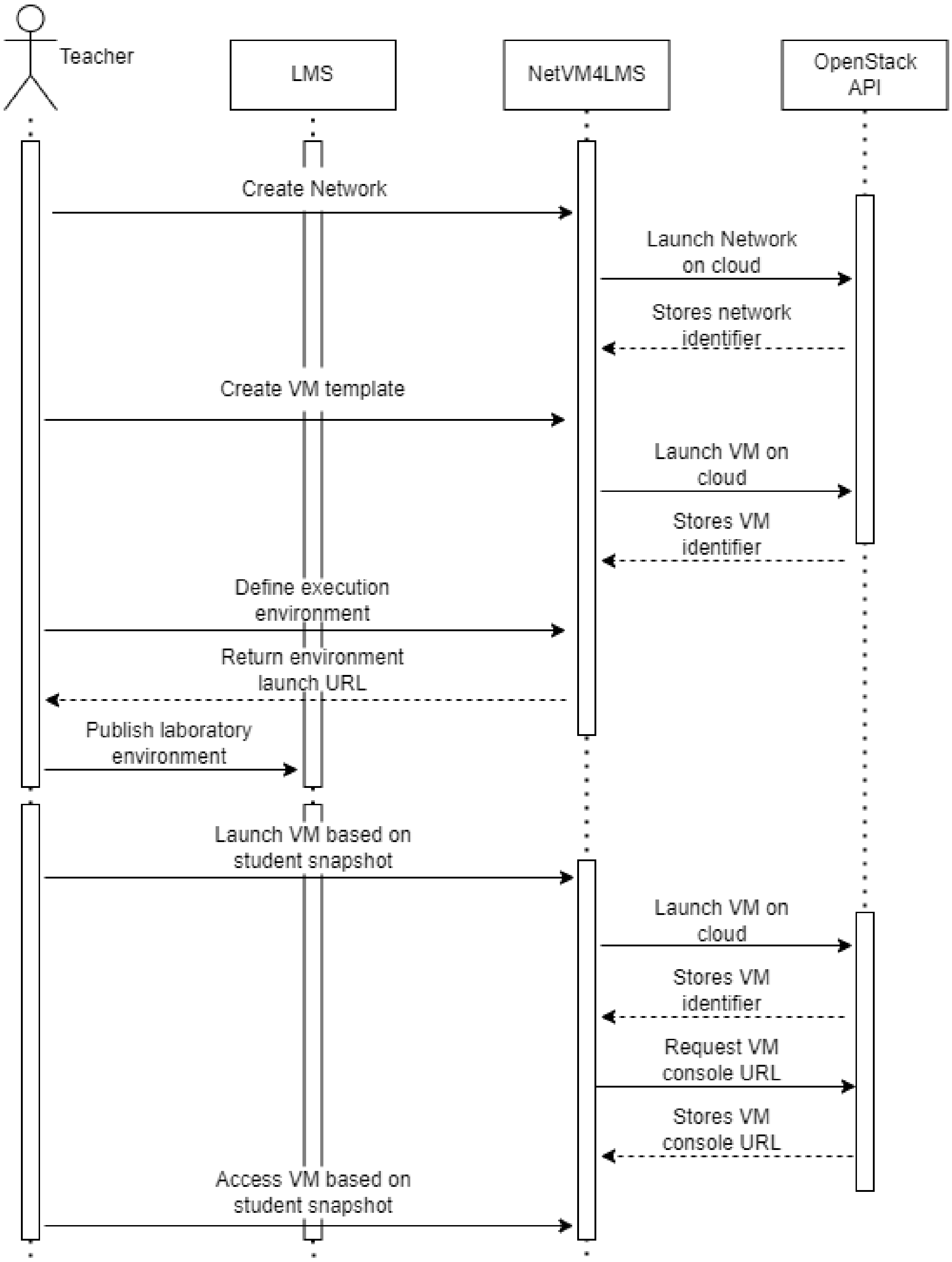


**Figure 4.3:** UML Sequence Diagram for teacher implementation

**Table 4.1:** Professor Routes

| Endpoint | Description |
| --- | --- |
| /prof/ | Professor dashboard where the networks, instances and tasks are created. |
| /prof/create_network | Interface to create a network. |
| /prof/network/<int:net_id>/info | Displays all the information related with the network identified by net_id. |
| /prof/network/<int:net_id>/delete | Delete the network identified by net_id. |
| /prof/network/<int:net_id>/addsubnet | Add a subnet to the network identified by net_id. |
| /prof/network/<int:net_id>/launch | Launch the network identified by net_id in the cloud infrastructure. |
| /prof/launch_instance | Launch the selected instances in the cloud infrastructure. |
| /prof/delete_instances | Delete the selected instances from the cloud infrastructure and from the database. |
| /prof/create_task | Interface to create a new task environment. |
| /prof/user/<int:user_id>/info | Shows the instances assigned to the user identified by user_id. |
| /prof/user/<int:snapshot_id>/launch | Launch an instance based on the snapshot taken by the student. |
| /prof/user/<int:snapshot_id>/open | Open the instance console for the instance created from the snapshot identified by snapshot_id. |

### 4.3.3 User Routes

In the section is presented the sequence of implemention for the student, and after that the routes that each one has access.

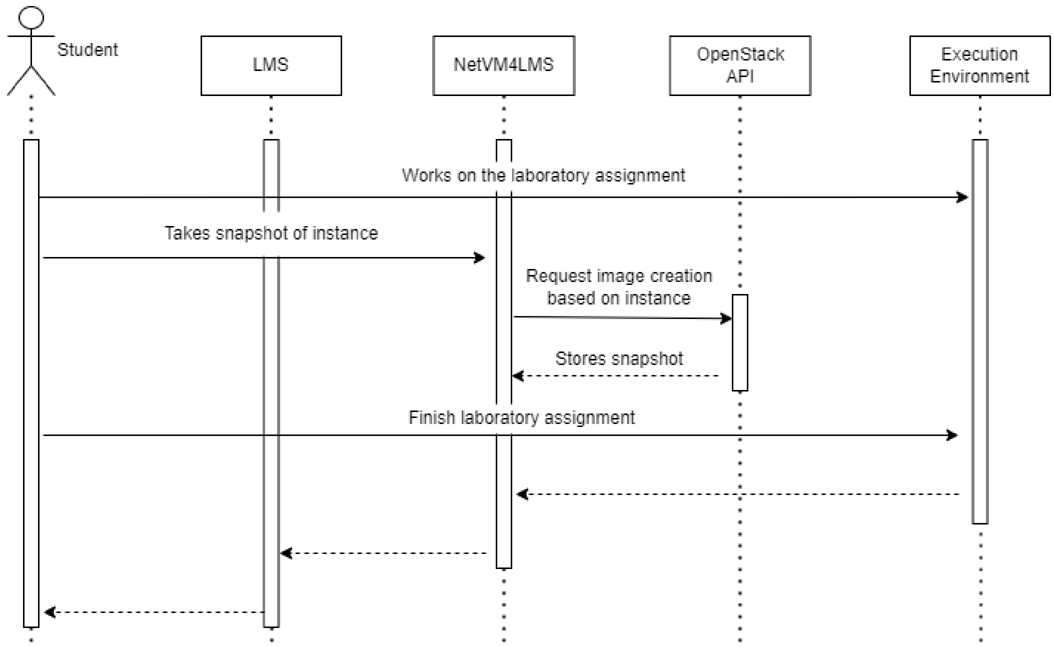


**Figure 4.4:** UML Sequence Diagram for student implementation

**Table 4.2:** Student Routes

| **Endpoint** | **Description** |
|---|---|
| /user/loading_env/<int:task_id> | Makes the load of the environment creating all the necessary instances and assign them to the student. |
| /user/env/<int:task_id> | Displays the environment description with the links to the instances console URL's and the use instructions for each of them. |
| /user/create_snapshot/<instance_id> | Allows the student to create a snapshot of the instance identified by instance_id. |

## 4.4 File Retrieval from Instances

Even though the file retrieval from students instances is not fully implemented in the system, it was tested and proved to be an important functionality to eventually implement. As said in section 4.1.2, Paramiko library was used to make the SSH connection between NetVM4LMS and the instances and after that, using the SFTP protocol, get the file from the instance.

To make the SSH connection the system needs to have access to the IP of the machine and the respective username and password. Depending on the image chosen by the teacher, the instance has a default username and password, but these informations are overwritten by the system, that will define the same password for all the instances created, making it more easy for teachers and student to log in. Also, in Openstack it is possible to define security rules for the instances to block traffic coming from a specific IP range or using a specific protocol, so in order to the system be able to get the file, it is important that the instances are not blocking SSH connections from the IP where NetVM4LMS is running.

After established the connection between the system and the instance from which the file will be retrieved, it is possible to download the file to the teacher computer, or to read and parse the file and save the important informations on the database making a relation with the student that worked on that instance.

# 5

# Execution

**Contents**

In the following section will be presented the some examples of executions of the system, i.e, types of exercises that can be implemented or concepts that can be demonstrated, and, for both teachers and students, a step-by-step guide on how to use it.

## 5.1 LMS Configuration

For teachers to have access to their own dashboard and start preparing the laboratory assignments and environments, they need to have access to the tool itself. To accomplish that they need to have a course page and NetVM4LMS installed in Moodle and that only the administrator, that is responsible for managing the Moodle site, upgrade, install plugins, etc, and have access to all the courses, can do.

The creation of a course is a simple task that any administrator can do and does not affect the way NetVM4LMS will work. To install NetVM4LMS, the administrator needs to access the Site Administration page, access the plugin menu and configure a new external tool manually, where some variables have to be defined:

- **Tool name**: It is the name that will be visible for the teachers when adding the external tool to their course pages. In this case, should be NetVM4LMS.
- **Tool URL**: Used to match tool URLs with the correct tool configuration, and given by the tool provider.
- **LTI Version**: This tool was made for the version LTI 1.0/1.1 and so thats the version that should be choosen.
- **Consumer Key and Shared secret**: It is defined in the application code so should be provided by the tool provider.
- **Default launch container**: Defines the mode that the tool is displayed when launched from the course page. NetVM4LMS was developed to be launched in a new browser window, so thats should be the option selected.

After the administrator finish the creation of the course and the installation of NetVM4LMS, the teacher only need to add the tool to the course page. To do that they need to access the course page to which they want to add the tool, turn on the *Edit Mode*, and add an external tool to the section they wish. They have to provide an activity name for the tool and the tool's launch URL, the rest was preconfigured by the administrator. For the first integration and for the teachers prepare the laboratories before students have access to them, is recommended that they hide it from them, like shown in the figure below.

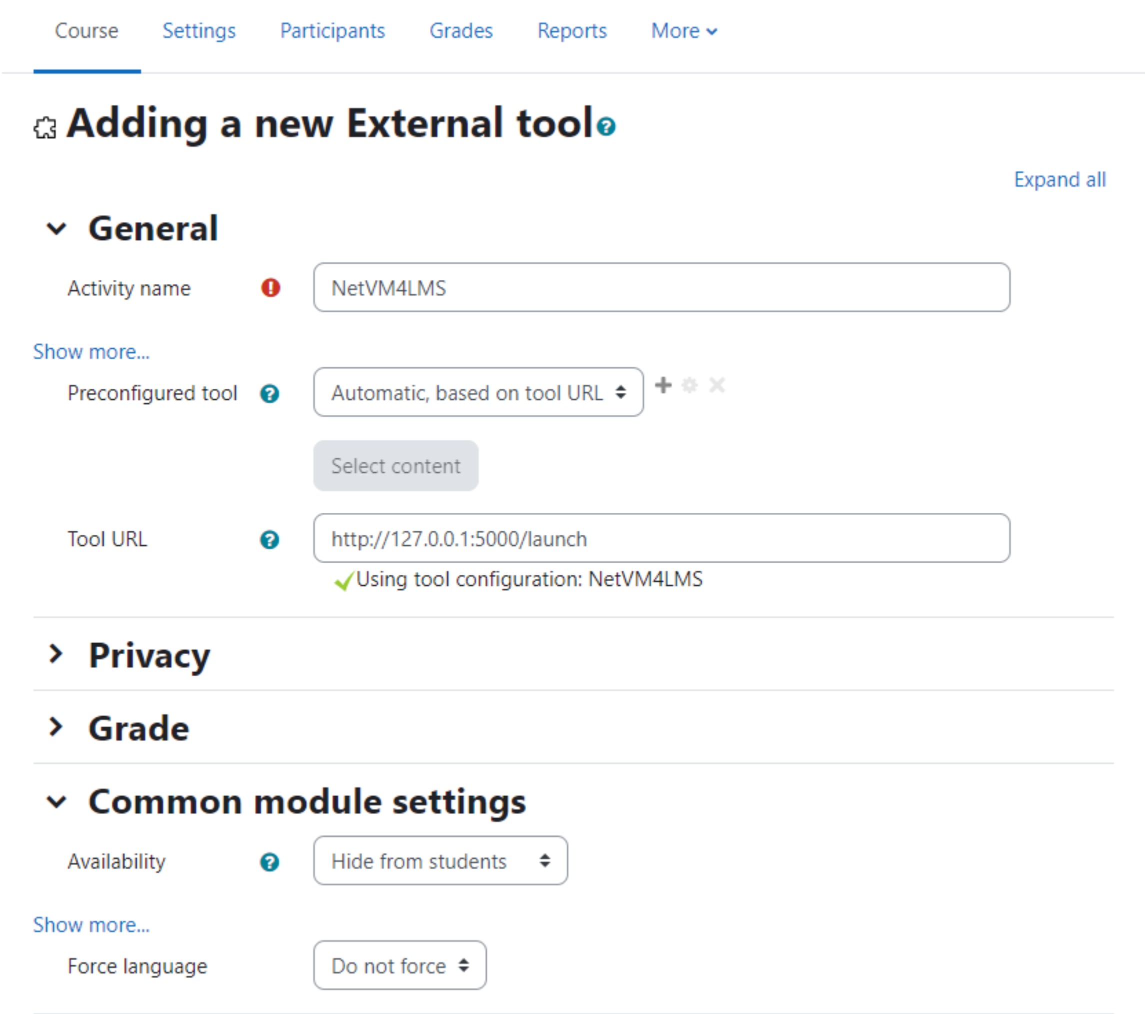


**Figure 5.1:** Teacher's view for adding the system to a course page

## 5.2 Teacher Execution

In this section will be given a guide step-by-step of how a teacher can create a simple network, a virtual machine template, use them to create a environment where a laboratory will be conducted, make that environment available on Moodle and, after a student finishes the laboratory, how to access each of the students virtual machines to check the work done.

1. Access the dashboard.

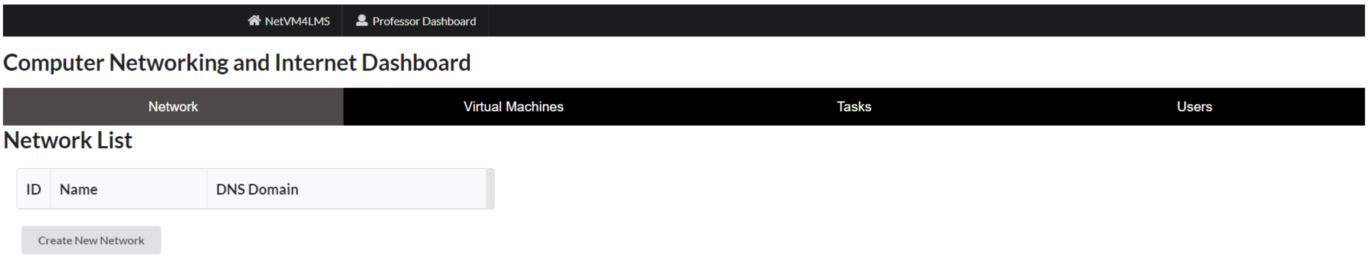


**Figure 5.2:** Teacher Dashboard page

2. Access the Network tab and create a new network and subnetwork.

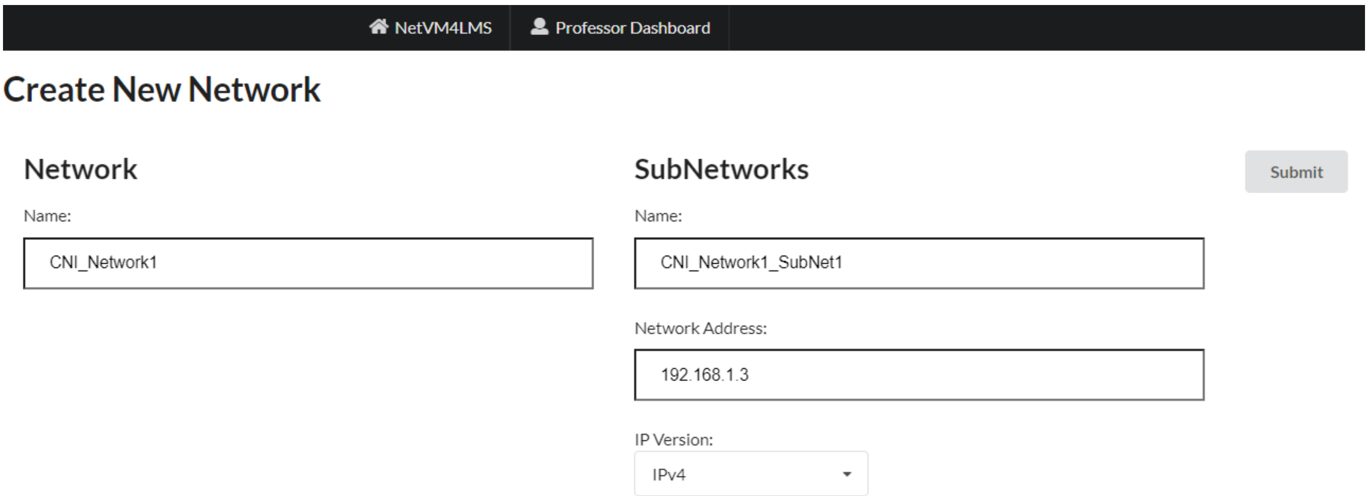


**Figure 5.3:** Interface to create a network

3. Launch the network created, by clicking on the row of the desired network on the network tab from the dashboard, and then clicking on the *Launch Network* to launch the network on OpenStack.

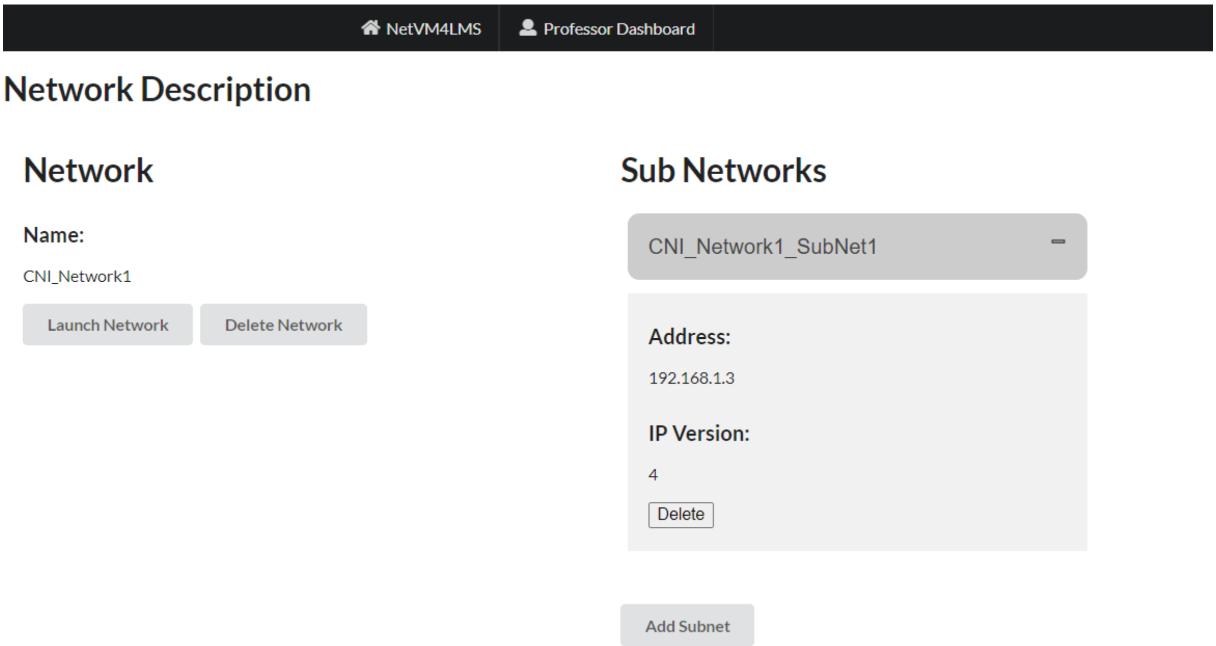


**Figure 5.4:** System Architecture Diagram

4. Go to the *Virtual Machines* tab in the dashboard and create a virtual machine template by defining the flavor, image, internet connection and network where the instance will be deployed.

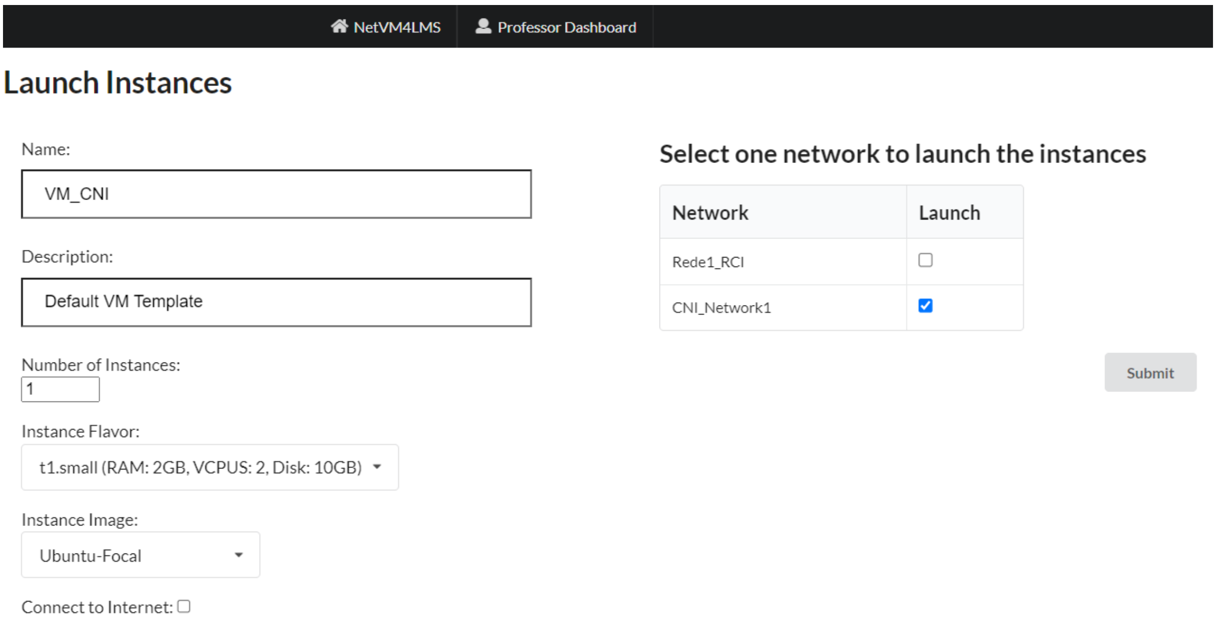


**Figure 5.5:** Interface to create virtual machines

5. Go to the *Tasks* tab and create a new environment defining the name and description, number of instances per student and for each of them, the name, use instructions, path of the files to be retrieved and template.

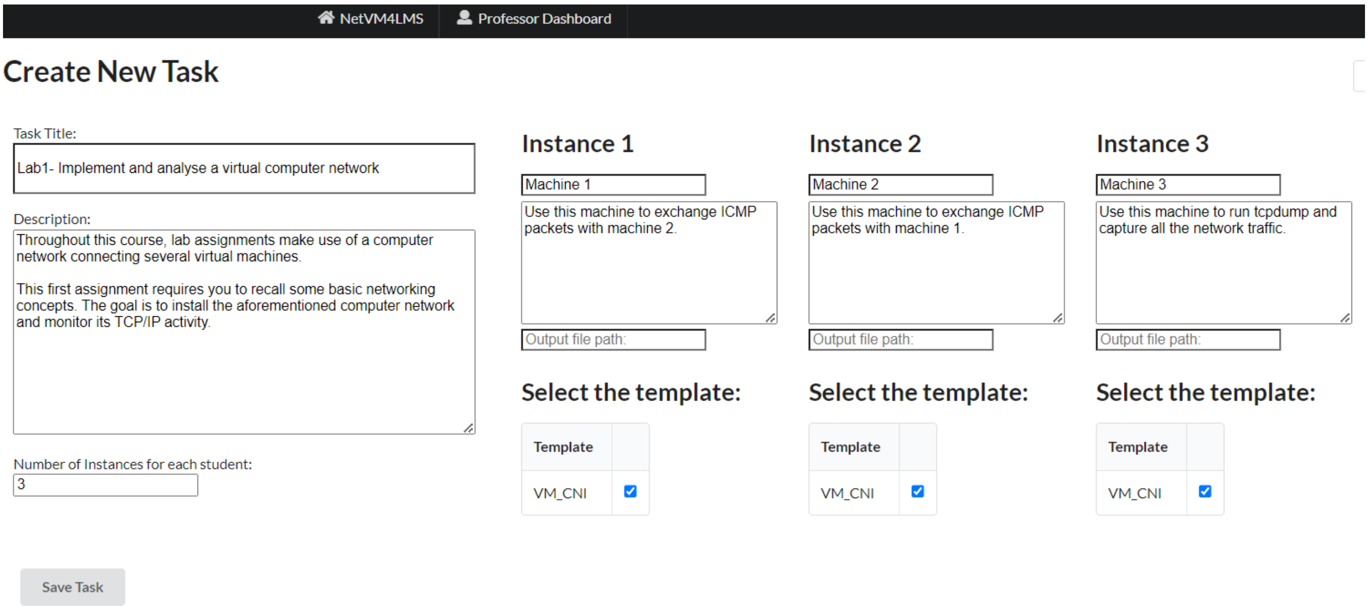


**Figure 5.6:** Interface to create environment

6. Publish environment on the Moodle by using the given launch URL for that environment.

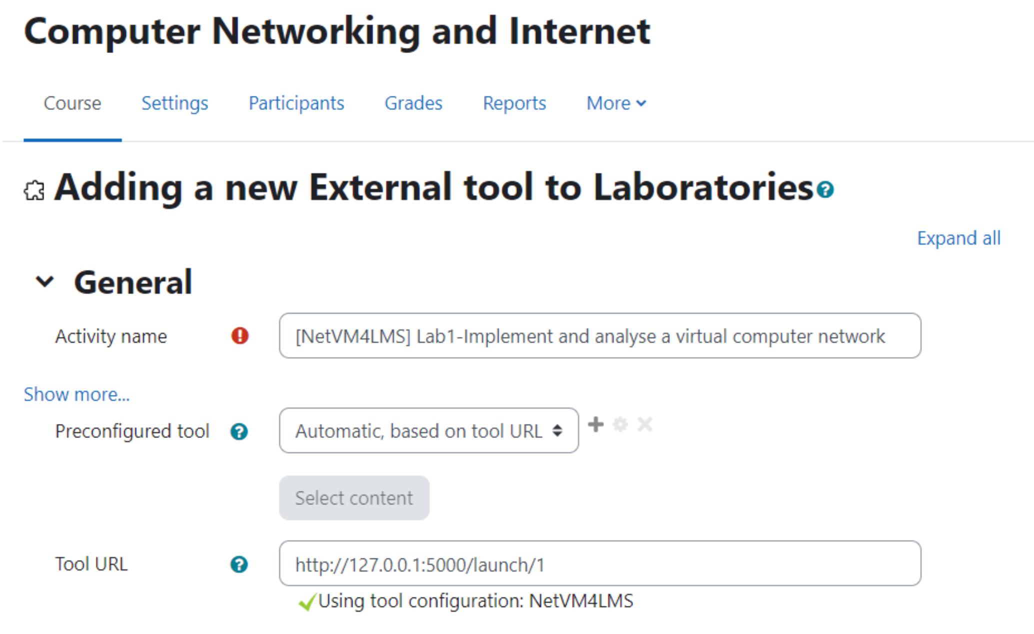


**Figure 5.7:** Moodle's interface to launch external tool

7. Access students instances by selecting the student in the *Users* tab and then choosing which instance want to access, launch a virtual machine using the snapshot taken by the student by clicking on the button *Launch* and then access the console by clicking on the button *Open*.

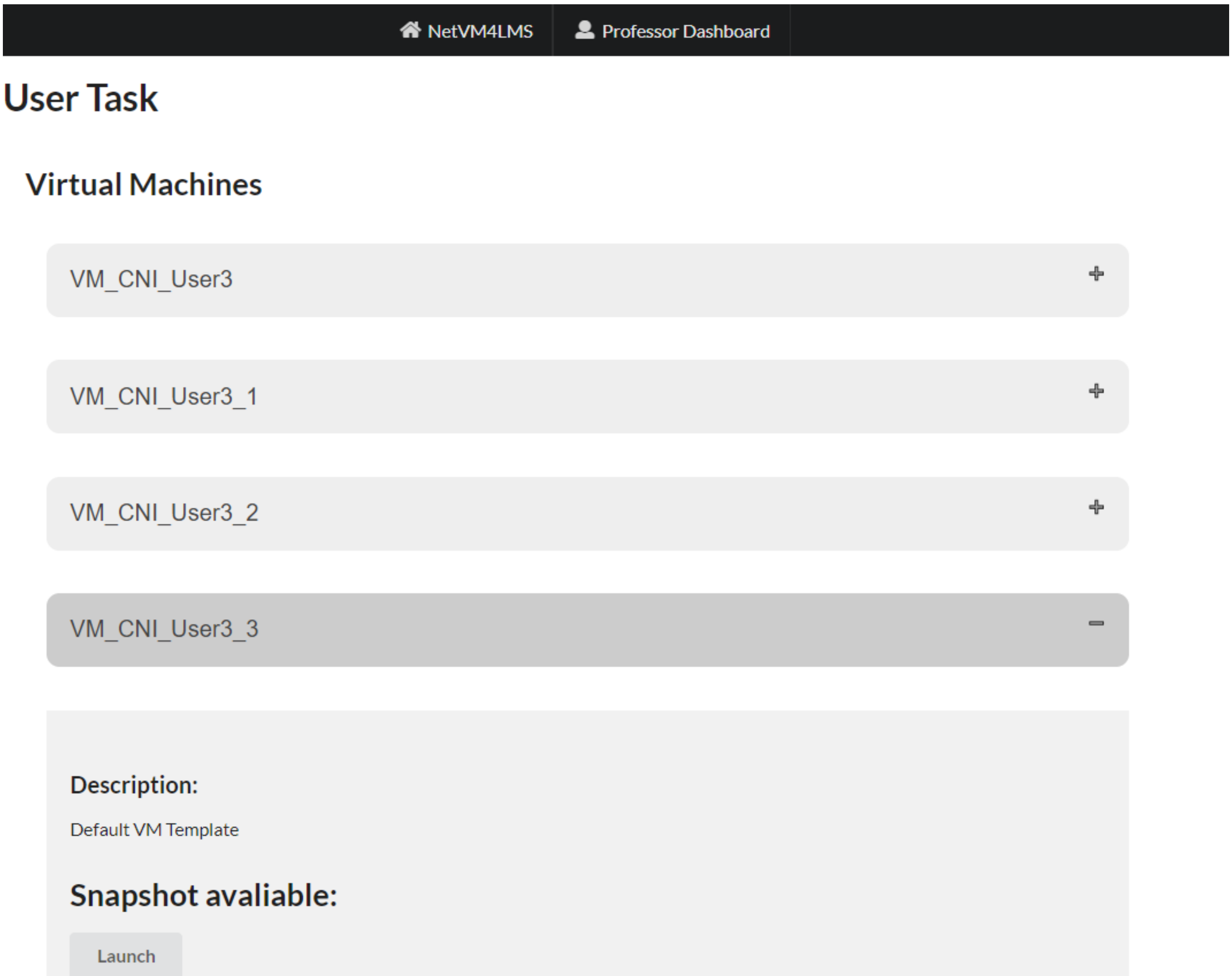


**Figure 5.8:** Page to access students' instances

## 5.3 Student Execution

The student's execution is quite simple when compared with the teacher. It only needs to access the Moodle's course page and click on the link to access the laboratory environment to have access to it. In that page, the student will have access to the laboratory instructions and for each instance its own instructions, the path of the files that will be retrieved and a button to take a snapshot of the instance, saving the current state.

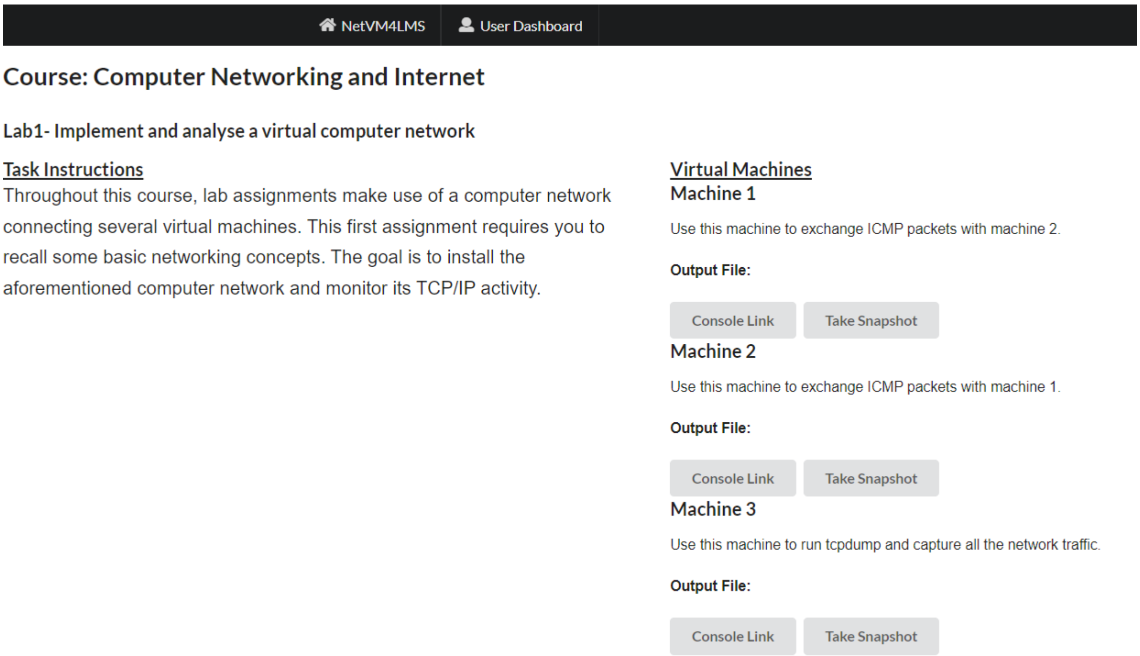


**Figure 5.9:** Student's view of the environment

For each instance there is also a button that will open the respective console on a new window, and that will be used by the student to solve the laboratory assignment by following the instructions give for that machine.

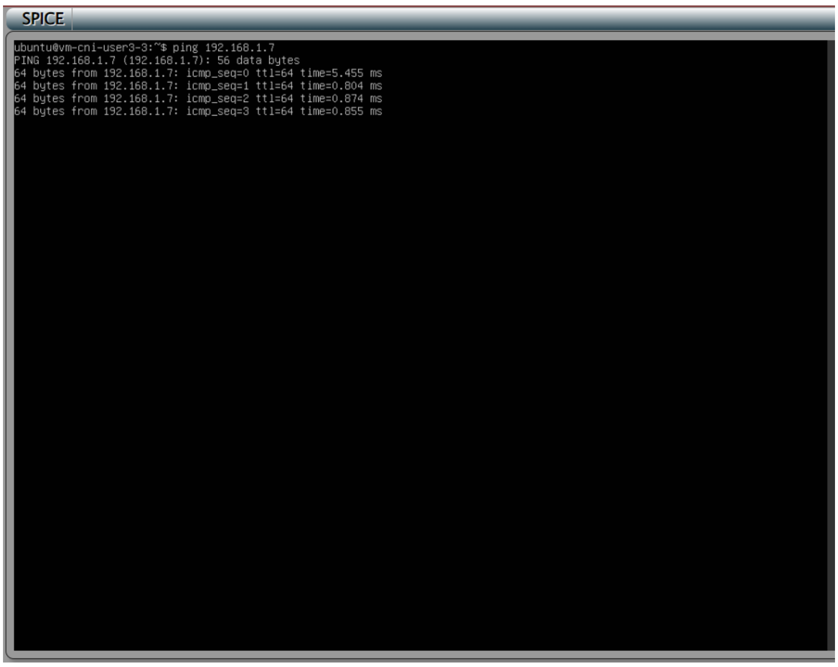


**Figure 5.10:** Instance's console

## 5.4 Execution Examples

This system offers the configuration of the main elements used in network laboratories, networks and instances. With these two elements there are a lot of scenarios that can be implemented to illustrate concepts that easier for students to understand when using a pratical approach, and, since those instance are running Linux, students will also have access to a wide variety of commands and packages. Some examples of concepts that are easily configured a demonstrated using NetVM4LMS are:

- **Firewall**: Setting up a firewall in one instance, by defining rules to filter traffic from specific IPs or to reject specific protocols like ICMP or telnet connections. like telnet connections, to demonstrate how importante it is to protect a network.
- **Protocol Vulnerabilities**: By defining one malicious instance it is possible to demonstrate simple exploits in protocols. One example is explore TCP vulnerabilities, like RST hijacking, where the malicious instance closes a connection between two other machines.
- **Routing**: Using two networks with several instances and an instance that manages the redirect of packets from one network to another, by defining routing rules and configure NAT.

## 5.5 Previous Work Comparizion

As said in section 2.2, this work was based on a previous created system, VM4LMS. This system allowed teachers to create virtual laboratories based on virtual machines, that could be made available via cloud or on the students' computers, and publish them on a LMS using LTI protocol. They were also capable of defining a deadline for the laboratory and also to fully configurate the instances and define the files to retrieve from them.

The main upgrade on VM4LMS that led to the creation of NetVM4LMS was the introduction of networks on the system. On VM4LMS the main focus was on the virtual machine itself, allowing the teacher to fully configure and make them avaliable with very low network configurations made, while the presented system allows the creation of networks and subnetworks and also allows teachers to choose in which networks to launch the created virtual machines.

Another improvements were also made to facilitate the teachers job in checking the results and the work done by the students. While VM4LMS only allowed teachers to have access to some files that were extracted from the virtual machines that the students used to do the laboratory assignment, with NetVM4LMS they have access to a copy of the students instances, allowing them to check the configurations made and if the tasks were successfully concluded.

# 6

# Conclusion and Future Work

**Contents**

## 6.1 Conclusion

Being the two main elements of a network simulation the network itself and the instances that are running on it, and because the configuration, execution and distribution of these elements can consume a lot of time and resources for both teachers and students, develop a system that could facilitate all these processes would be quite beneficial in engineer education.

As result of this thesis, a system called NetVM4LMS was createad which allows students to have an experience similar to a real physical network laboratory by giving them access to an environment with virtual machines running in a defined network topology, managed by a cloud infrastructure that allows them to access that environment from anywhere and at anytime, without consuming resources from their computers. This system is an alternative for teachers to use in pratical classes making easier for them to configure and distribute the virtual machines and to access the resolutions of the students to the proposed laboratory assignment, but also eliminates the need for specific physical hardware, signficantly reducing the costs associated with setting up traditional networks environments in the labs. The integration of the tool with Moodle gives a channel to share the environments, but also provides a way for users to be authenticated inside the system.

In conclusion, the findings of this work demonstrate the significance and benefits of using virtualization technologies, such as the OpenStack, in educational settings. By simulating network environments and enabling interactions between different machines, students are able to develop practical skills and knowledge that directly relate to their future careers, and teachers have a tool that allows them to be flexible and creative when setting up laboratory assignments to demonstrate the theoretical concepts that they teach in classes. This thesis has contributed to computer engineering education by presenting a system that integrates virtual machines and networks running in a cloud infrastructure into the teaching and learning process. This system has the potential to improve practical learning experiences, optimize resource utilization, and prepare students for the challenges of the modern technological landscape.

## 6.2 Future Work

The system NetVM4LMS is now capable of make simple network simulations and is becoming quite close of a physical laboratory. But, in order to increase functionality making it even more capable of simulate real world scenarios and to facilitate the job of the teachers, there are some areas that can be improved.

To increase the variaty of network topologies, would be important to implement the creation of routers. The API from Openstack service, Neutron, allows the creation and configuration of routers in a similar way than networks. This would allow the creation of more types of network topologies, therefore increasing the types of exercises and simulation that the system allows. Also, related with networks, implement additional network funcionalities, like Firewall, QoS, DNS or VPN, that are offered as services in Openstack, would make the system more complete and flexible in terms of network elements and configurations.

Related with the integration with Moodle and the use of the LTI protocol, it is possible to grade the activities solved in the external tool and send it back to Moodle. This, allied with the implementation of a tracker of the progress of the laboratory and the correction of the proposed exercises, would make the evaluation process much easier and autonomous for the teachers.

# Bibliography


[1] Rahman, M.A., Pakstas, A. Tools and Techniques for Teaching and Research in Network Design and Simulation. SN COMPUT. SCI. 4, 269 (2023). https://doi.org/10.1007/s42979-023-01684-6

[2] Advantages of linux - Javapoint. Available at: https://www.javatpoint.com/advantages-of-linux

[3] Liangxu, Sun & Wu, liansheng & Zhang, Yujun & Yin, Hang. (2013). Comparison between physical devices and simulator software for Cisco network technology teaching. 1357-1360. 10.1109/ICCSE.2013.6554134.

[4] lal, Dayananda, Behnam Ghorbani and Solmaz Vaghri. “A SURVEY ON THE USE OF GNS3 FOR VIRTUALIZING COMPUTER NETWORKS.” (2016).

[5] A. R. Khan, S. M. Bilal and M. Othman, ”A performance comparison of open source network simulators for wireless networks,” 2012 IEEE International Conference on Control System, Computing and Engineering, Penang, Malaysia, 2012, pp. 34-38, doi: 10.1109/ICCSCE.2012.6487111.

[6] Kabir, Mohammed & Islam, Syful & Hossain, Md & Hossain, Sazzad. (2014). Detail Comparison of Network Simulators. 10.13140/RG.2.1.3040.9128.

[7] M. Wannous and H. Nakano, ”NVLab, a Networking Virtual Web-Based Laboratory that Implements Virtualization and Virtual Network Computing Technologies,” in IEEE Transactions on Learning Technologies, vol. 3, no. 2, pp. 129-138, April-June 2010, doi: 10.1109/TLT.2009.31.

[8] S. J. Dutta and R. Bhattacharjee, ”Integration of Virtual Laboratories: A Step Toward Enhancing E-learning Technology,” 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), 2019, pp. 1-5, doi: 10.1109/I2CT45611.2019.9033848.

[9] Marquès, J.M., Lazaro, D., Juan, A.A., Vilajosana, X., Domingo, M. and Jorba, J. (2013), PlanetLab@UOC: A real lab over the Internet to experiment with distributed systems. Comput. Appl. Eng. Educ., 21: 265-275. https://doi.org/10.1002/cae.20468

[10] Agrela, Victor Jesus Farinha. “VM4LMS - Virtual Machines on Education.” Instituto Superior Técnico, 2020.

[11] Al Rawashdeh, A. Z., Mohammed, E. Y., Al Arab, A. R., Alara, M., Al-Rawashdeh, B., Al-Rawashdeh, B. (2021, May 17). Advantages and Disadvantages of Using e-Learning in University Education: Analyzing Students' Perspectives. Electronic Journal of E-Learning, 19(3), 107–117. https://doi.org/10.34190/ejel.19.3.2168

[12] D. Baron, "The 13 Must-Have Features of a Learning Management System", 2020. [Online]. Available: https://www.peoplefluent.com/blog/learning/13-must-have-features-of-a-learning-management-system/.

[13] M. R. Elabnody. A survey of top 10 open source learning management systems. Int J Scient Technol Res, 4(8):7–11, 2015.

[14] "Canvas LMS → Análise, Vantagens e Desvantagens", Bit4learn, 2021. [Online]. Available: https://bit4learn.com/pt/lms/canvas-lms/.

[15] "SAKAI EAD → Análise, Vantagens e Desvantagens", Bit4learn, 2021. [Online]. Available: https://bit4learn.com/pt/lms/sakai/.

[16] "Open edX LMS: The Learning Management System — eduNEXT", eduNEXT, 2021. [Online]. Available: https://www.edunext.co/discover/lms/.

[17] "LTI v2: An Introduction — IMS Global Learning Consortium", Imsglobal.org. [Online]. Available: http://www.imsglobal.org/lti-v2-introduction.

[18] "Basic Overview of How LTI works — IMS Global Learning Consortium", Imsglobal.org. [Online]. Available: http://www.imsglobal.org/basic-overview-how-lti-works. [Accessed: 31- May- 2021].

[19] "What is virtualization?", Available: https://opensource.com/resources/virtualization.

[20] 5 Different Types of Virtualization and Top 3 Virtualization Software", TekTools, 2020. [Online]. Available: https://www.tek-tools.com/apm/maximize-efficiency-with-virtualization-software.

[21] Key Differences Between Desktop Virtualization and Server Virtualization", Devonit.com, 2014. [Online]. Available: http://www.devonit.com/blog/key-differences-between-desktop-virtualization-and-server-virtualization.

[22] D. T. Vojnak, B. S. orđević, V. V. Timčenko and S. M. Štrbac, "Performance Comparison of the type-2 hypervisor VirtualBox and VMWare Workstation," 2019 27th Telecommunications Forum (TELFOR), 2019, pp. 1-4, doi: 10.1109/TELFOR48224.2019.8971213.

[23] M. Mahjoub, A. Mdhaffar, R. B. Halima and M. Jmaiel, "A Comparative Study of the Current Cloud Computing Technologies and Offers," 2011 First International Symposium on Network Cloud Computing and Applications, 2011, pp. 131-134, doi: 10.1109/NCCA.2011.28.

[24] S. Baca, ”Virtualization for Newbies: Five Types of Virtualization”, Globalknowledge.com, 2013. [Online]. Available: https://www.globalknowledge.com/us-en/resources/resource-library/articles/virtualization-for-newbies-five-types-of-virtualization/

[25] What Is vSphere? An Introduction to VMware’s Virtualization Platform”, New Horizons Worldwide, 2019. [Online]. Available: https://www.newhorizons.com/article/what-is-vsphere-an-introduction-to-vmware-virtualization-platform. [Accessed: 31- May- 2021].

[26] X. Wen, G. Gu, Q. Li, Y. Gao and X. Zhang, ”Comparison of open-source cloud management platforms: OpenStack and OpenNebula,” 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, 2012, pp. 2457-2461, doi: 10.1109/FSKD.2012.6234218.

[27] D. Turull, M. Hidell and P. Sjödin, ”Performance evaluation of openflow controllers for network virtualization,” 2014 IEEE 15th International Conference on High Performance Switching and Routing (HPSR), 2014, pp. 50-56, doi: 10.1109/HPSR.2014.6900881.

[28] M. Shin, K. Nam and H. Kim, ”Software-defined networking (SDN): A reference architecture and open APIs,” 2012 International Conference on ICT Convergence (ICTC), 2012, pp. 360-361, doi: 10.1109/ICTC.2012.6386859.

[29] K. Kaur, J. Singh, and N. S. Ghumman, “Mininet as software defined networking testing platform,” in International Conference on Communication, Computing Systems (ICCCS), 2014, pp. 139–42.

[30] M. Bose, ”VirtualBox Network Settings: All You Need to Know”, Official NAKIVO Blog, 2019. [Online]. Available: https://www.nakivo.com/blog/virtualbox-network-setting-guide/.

[31] Selecting the Network Connection Type for a Virtual Machine”, Docs.vmware.com, 2019. [Online]. Available: https://docs.vmware.com/en/VMware-Workstation-Pro/16.0/com.vmware.ws.using.doc/GUID-3B504F2F-7A0B-415F-AE01-62363A95D052.html.

[32] OpenStack Docs: OpenStack Networking”, Docs.openstack.org, 2020. [Online]. Available: https://docs.openstack.org/neutron/pike/admin/intro-os-networking.html.

[33] About vSphere Networking”, Docs.vmware.com, 2020. [Online]. Available: https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.networking.doc/GUID-35B40B0B-0C13-43B2-BC85-18C9C91BE2D4.html.

[34] Rawashdeh, Alaa Zuhir Al et al. “Advantages and Disadvantages of Using e-Learning in University Education: Analyzing Students’ Perspectives.” Electronic Journal of e-Learning 19 (2021): 107-117.

[35] Open Source Cloud Computing Platform Software - OpenStack. https://www.openstack.org/software/

[36] 3. Computing service: OpenStack Nova — RCC Documentation 0.2-beta documentation. https://docs.hpc.cam.ac.uk/cloud/userguide/03-nova.html

[37] “Compute API — Nova Documentation.” Docs.openstack.org, docs.openstack.org/api-ref/compute/.

[38] “Networking API V2.0 — Networking API Reference Documentation.” Docs.openstack.org, docs.openstack.org/api-ref/network/v2/.

[39] “Identity API v3 (CURRENT) — Keystone Documentation.” Docs.openstack.org, docs.openstack.org/api-ref/identity/v3/.

[40] Aslam, Fankar & Mohammed, Hawa & Lokhande, Prashant. (2015). Efficient Way Of Web Development Using Python And Flask.. International Journal of Advanced Research in Computer Science.

[41] T. E. Oliphant, ”Python for Scientific Computing,” in Computing in Science & Engineering, vol. 9, no. 3, pp. 10-20, May-June 2007, doi: 10.1109/MCSE.2007.58.

[42] Engineering, MIT ODL. “PyLTI: PyLTI Provides Python Implementation of IMS LTI Interface That Works with EdX.” PyPI, pypi.org/project/PyLTI/.

[43] Rasekgala, Mokgadi. “Paramiko- How to SSH and Transfer Files with Python.” Medium, 23 Nov. 2017, medium.com/@keagileageek/paramiko-how-to-ssh-and-file-transfers-with-python-75766179de73.

[44] “What Is Flask Python - Python Tutorial.” Pythonbasics.org, 2021, pythonbasics.org/what-is-flask-python/.

[45] Deery, Matthew. “The Flask Web Framework: A Beginner’s Guide.” Careerfoundry.com, 1 Aug. 2022, careerfoundry.com/en/blog/web-development/what-is-flask/.

[46] “Flask-Login — Flask-Login 0.4.1 Documentation.” Readthedocs.io, 2019, flask-login.readthedocs.io/en/latest/.

[47] “How to Use and Validate Web Forms with Flask-WTF — DigitalOcean.” Www.digitalocean.com, www.digitalocean.com/community/tutorials/how-to-use-and-validate-web-forms-with-flask-wtf.

[48] L. Junyan, X. Shiguo and L. Yijie, ”Application Research of Embedded Database SQLite,” 2009 International Forum on Information Technology and Applications, doi: 10.1109/IFITA.2009.408.

[49] “SQLite Home Page.” Sqlite.org, 2019, sqlite.org/index.html.

[50] “SQLAlchemy ORM Tutorial for Python Developers.” Auth0 - Blog, auth0.com/blog/sqlalchemy-orm-tutorial-for-python-developers/.

# 7

# Appendix A

| Nº | Question | Types of Answer |
|---|---|---|
| 1 | Name: | Open Answer |
| 2 | Areas of the curricular units that you teach (Networks, Distributed Systems, Security, Operating Systems, ...): | Open Answer |
| 3 | Course Units you teach: | Open Answer |
| 4 | Courses you teach: | Open Answer |
| 5 | In any of the CUs you teach, do students use virtual machines/images provided by teachers? | Yes/No% |
| 6 | The virtual machines provided are instantiated in: | Multiple Choice |
| 7 | The Virtual machines used are used for: | Multiple Choice |
| 8 | In any of the uses of virtual machines, do students need to create/configure these virtual machines or install the Operating System? | Yes/No |
| 9 | In any of the uses of virtual machines, do students program on the virtual machines provided? | Yes/No |
| 10 | In any of the uses of Virtual machines, do students change the configuration of the virtual machines provided? | Yes/No |
| 11 | In any of the uses of virtual machines, do students need multiple virtual machines simultaneously? | Yes/No |
| 12 | What is the physical network topology of this machine(s)? | Multiple Choice |
| 13 | Do students need to configure the network in the operating system? | Yes/No |

| | | |
|---|---|---|
| 14 | Are students required to configure the routers/gateways of these networks? | Yes/No |
| 15 | What type of addressing is available? | Multiple Choice |
| 16 | Briefly describe the type of exercises students solve using machines provided by the teachers: | Open Answer |
| 17 | Indicate which of the following LMS you know: | Yes/No |
| 18 | Indicate which of the following LMS you have worked (as a teacher or student): | Yes/No |
| 19 | If you have worked with any of the above LMS as a teacher, for what purpose did you use it? | Multiple Choice |
| 20 | Do you know OpenStack? | Yes/No |
| 21 | Have you ever used OpenStack? | Yes/No |
| 22 | If you have used Openstack before, please briefly describe your use of it: | Open Answer |
| 23 | Do you find it interesting to make already configured virtual machines available in a Cloud infrastructure for students to use? | Likert Scale (1-5) |
| 24 | Do you find it interesting to provide exercises based on the use of virtual machines configured by the teachers? | Likert Scale (1-5) |
| 25 | Do you find it interesting to provide examples of virtual machines that students can access to illustrate theoretical concepts? | Likert Scale (1-5) |
| 26 | If yes (answer $>$3) to any of the previous questions, please describe what possible uses the virtual machines accessible by students through an LMS would have in the courses you teach: | Open Answer |
| 27 | With the mandatory remote classes, did you feel that there were any situations that it would have been beneficial to provide students with virtual machines in Cloud infrastructures? | Yes/No |
| 28 | If yes, describe what uses these virtual machines would have in the CUs you teach: | Open Answer |
| 29 | Do you find it interesting to make multiple virtual machines available on the same network? | Likert Scale (1-5) |
| 30 | Do you find it interesting to make multiple virtual machines available on different networks? | Likert Scale (1-5) |
| 31 | Do you find it interesting to provide a private DNS service for machines? | Likert Scale (1-5) |
| 32 | Do you think it is interesting for students to configure the DNS service? | Likert Scale (1-5) |
| 33 | Do you think it would be interesting to provide a service to configure the firewalls of the machines on the network? | Likert Scale (1-5) |
| 34 | Do you think it is interesting for students to configure the firewalls of the machines? | Likert Scale (1-5) |
| 35 | Do you find it interesting to be able to control the QoS of network connections? | Likert Scale (1-5) |
| 36 | Besides the features mentioned above, do you have any more that you would like to see implemented? | Open Answer |
| 37 | When a working prototype of this system exists, are you interested in trying it out and evaluating it? | Yes/No |

# 8

# Appendix B

| Nº | Question | Answer |
|---|---|---|
| 1 | Name: | —– |
| 2 | Areas of the curricular units that you teach (Networks, Distributed Systems, Security, Operating Systems, ...): | Check figure 3.1 |
| 3 | Course Units you teach: | Check figure 3.2 |
| 4 | Courses you teach: | —– |
| 5 | In any of the CUs you teach, do students use virtual machines/images provided by teachers? | Yes:58% No:42% |
| 6 | The virtual machines provided are instantiated in: | Virtual Machine Manager: 100% |
| 7 | The Virtual machines used are used for: | Demonstration of theoretical concepts:16%<br>Exercises to be performed by the students: 26%<br>Support to the realization of the laboratories: 37%<br>Support for evaluation exercises: 21% |
| 8 | In any of the uses of virtual machines, do students need to create/configure these virtual machines or install the Operating System? | Yes:57% No:43% |
| 9 | In any of the uses of virtual machines, do students program on the virtual machines provided? | Yes:100% No:0% |

| | | |
|---|---|---|
| 10 | In any of the uses of Virtual machines, do students change the configuration of the virtual machines provided? | Yes:86% No:14% |
| 11 | In any of the uses of virtual machines, do students need multiple virtual machines Yesultaneously? | Yes:57% No:43% |
| 12 | What is the physical network topology of this machine(s)? | Single Local Area Network: 50%<br>Multiple Local Area Networks:50% |
| 13 | Do students need to configure the network in the operating system? | Yes:57% No:43% |
| 14 | Are students required to configure the routers/gateways of these networks? | Yes:43% No:57% |
| 15 | What type of addressing is available? | Machines with private addresses and Internet access: 100% |
| 16 | Briefly describe the type of exercises students solve using machines provided by the teachers: | Open Answer |
| 17 | Indicate which of the following LMS you know: | Moodle-Yes:100% No:0%<br>Canvas-Yes:8,3% No:91,7%<br>Sakai-Yes:0% No:100%<br>OpenedX-Yes:16,7% No:83,3% |
| 18 | Indicate which of the following LMS you have worked (as a teacher or student): | Moodle-Yes:100% No:0%<br>Canvas-Yes:0% No:100%<br>Sakai-Yes:0% No:100%<br>OpenedX-Yes:16,7% No:83,3% |
| 19 | Se If you have worked with any of the above LMS as a teacher, for what purpose did you use it? | Making content available:42%<br>Self-Assessment Exercises:5%<br>Summative assessment:47%<br>Outra:5% |
| 20 | Do you know OpenStack? | Yes:58% No:42% |
| 21 | Have you ever used OpenStack? | Yes:17% No:83% |
| 22 | If you have used Openstack before, please briefly describe your use of it: | Open Answer |
| 23 | Do you find it interesting to make already configured virtual machines available in a Cloud infrastructure for students to use? | Average value: 4.5 |
| 24 | Do you find it interesting to provide exercises based on the use of virtual machines configured by the teachers? | Average value: 4.75 |
| 25 | Do you find it interesting to provide examples of virtual machines that students can access to illustrate theoretical concepts? | Average value: 4.33 |
| 26 | If yes (answer $>$3) to any of the previous questions, please describe what possible uses the virtual machines accessible by students through an LMS would have in the courses you teach: | Open Answer |
| 27 | With the mandatory remote classes, did you feel that there were any situations that it would have been beneficial to provide students with virtual machines in Cloud infrastructures? | Yes:58% No:42% |
| 28 | If yes, describe what uses these virtual machines would have in the CUs you teach: | Open Answer |
| 29 | Do you find it interesting to make multiple virtual machines available on the same network? | Average value: 4.25 |

| | | |
|---|---|---|
| 30 | Do you find it interesting to make multiple virtual machines available on different networks? | Average value: 4.08 |
| 31 | Do you find it interesting to provide a private DNS service for machines? | Average value: 4.08 |
| 32 | Do you think it is interesting for students to configure the DNS service? | Average value: 4.17 |
| 33 | Do you think it would be interesting to provide a service to configure the firewalls of the machines on the network? | Average value: 4.08 |
| 34 | Do you think it is interesting for students to configure the firewalls of the machines? | Average value: 4.33 |
| 35 | Do you find it interesting to be able to control the QoS of network connections? | Average value: 4 |
| 36 | Besides the features mentioned above, do you have any more that you would like to see implemented? | Open Answer |
| 37 | When a working prototype of this system exists, are you interested in trying it out and evaluating it? | Yes:100% No:0% |