UAM
Universidad Autónoma de Madrid

Escuela Politécnica Superior

23
24

Bachelor thesis

VisUAM: a student progress visualization tool integrated into Moodle

Luis Alejandro Felice Arria

Escuela Politécnica Superior
Universidad Autónoma de Madrid
C\Francisco Tomás y Valiente nº 11

UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Bachelor as Computer Science and Engineering

BACHELOR THESIS

VisUAM: a student progress visualization tool integrated into Moodle

Author: Luis Alejandro Felice Arria
Advisor: José Luis Jorro Aragoneses

junio 2024

Luis Alejandro Felice Arria
VisUAM: a student progress visualization tool integrated into Moodle

Luis Alejandro Felice Arria
C\ Francisco Tomás y Valiente Nº 11

PRINTED IN SPAIN

Persistence is very important.
You must not resign unless
you are forced to resign.

Elon Musk

Agradecimientos

En primer lugar, quiero agradecer a Jose Luis Jorro, mi tutor de TFG, por su inestimable ayuda, orientación y paciencia durante todo el proceso de desarrollo de este proyecto.

Me gustaría expresar mi gratitud a los docentes y colegas de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid, quienes me han acompañado en mi camino hacia el título de ingeniería informática.

Agradezco profundamente a mi familia por siempre creer en mí y brindarme las oportunidades necesarias para aprender y desarrollarme personalmente.

Finalmente, quiero agradecer a mis compañeros de clase por su compañerismo y apoyo durante estos años de estudio.

Resumen

En los últimos años, la tecnología ha estado influyendo cada vez más en diversos aspectos de nuestra vida diaria, y la educación no es una excepción; este proceso se ha visto acelerado significativamente debido a eventos como la pandemia vivida hace unos años, en donde universidades se encontraron obligadas a transformar sus métodos de enseñanza. De esta manera, en el año académico 2023-2024, un grupo de docentes de la Universidad Autónoma de Madrid crearon una aplicación que permite tanto a los profesores como a los estudiantes participantes de un curso supervisar su progreso académico durante el mismo.

Anteriormente a este trabajo, los docentes que querían utilizar la aplicación tenían que procesar y cargar manualmente todos los datos del curso, y en etapas avanzadas del mismo, se encontraron con una gran cantidad de datos y actividades, lo que limitaba el potencial de la aplicación, debido al tiempo que llevaba cargar y procesar todos estos datos. En este contexto, la aplicación VisUAM, desarrollada en este trabajo, el cual se presenta como proyecto de innovación docente, automatiza la recopilación y procesamiento de datos necesarios para evaluar el progreso de cada estudiante durante un curso.

VisUAM es una aplicación que está desplegada en un servidor de la Escuela Politécnica Superior (EPS) de la Universidad Autónoma de Madrid (UAM) y está a disposición de todos los profesores que deseen utilizarla en sus cursos. Adicionalmente, la arquitectura de sistemas distribuidos basada en Docker permitió la creación de esta aplicación utilizando tecnologías como Django, Python e NGINX, entre otras que se describen en el presente trabajo. Además, esta aplicación ya ha sido sometida a varias etapas de pruebas para garantizar un funcionamiento óptimo.

Finalmente, esperamos que esta aplicación en un futuro ayude tanto a los estudiantes como a los profesores, mejorando los métodos de enseñanza de la universidad y potenciando el rendimiento de cada estudiante en cada asignatura que utilice esta aplicación.

Palabras clave

Django, Docker, Python, Learning Analytics, proyecto de innovación docente

Abstract

In recent years, technology has been increasingly influencing various aspects of our daily lives, and education is no exception; this process has been significantly accelerated due to events such as the pandemic experienced a few years ago, where universities were forced to transform their teaching methods. In this way, in the academic year 2023-2024, a group of teachers from the Universidad Autonoma de Madrid (UAM) created an application that allows both teachers and students to monitor their academic progress during a course.

Prior to this project, teachers who wanted to use the application had to manually process and load all the course data, and as the course progressed, they encountered a large volume of data, which limited the application's potential due to the time required to load and process all of it. In this regard, the VisUAM application, developed in this project and presented as an official "projecto de innovación docente", automates the gathering and processing of data required to evaluate each student's progress during a course.

VisUAM application is deployed on a server inside the Escuela Politécnica Superior (EPS) of the UAM and is available to all teachers who wish to use it in their courses. Additionally, the Docker-based distributed system architecture allowed the development of this application, which utilized technologies such as Django, Python, and NGINX, among others, as explained in this paper. Furthermore, this application has already been tested in multiple stages to assure optimal performance.

Finally, we hope that this application in the future will help both students and teachers by improving the university's teaching methods and enhancing the performance of each student in every subject that uses this application.

Keywords

Django, Docker, Python, Learning Analytics, proyecto de innovación docente

Table of Contents

Lists

List of figures

1 Introduction

This chapter will provide an explanation of the motivations behind the creation of the VisUAM application, including the technologies and reasons that served as inspiration, as well as an overview of fields like learning analytics. Furthermore, elucidated will be the development's goals that were addressed.

1.1. Motivation

Universities across Europe assign ETC points to each course, which is equivalent to roughly 25 to 30 hours of work [1]. For example, students at Universidad Autónoma de Madrid (UAM) are usually required to carry at least 60 ECTS per year, which corresponds to approximately 1700 hours of study time [2]. With all of this in mind, students experience a great deal of stress and effort during each of their university courses, which can have an impact on their overall performance and results during a course, as indicated in their marks.

According to the World Health Organization, in order for students to engage in full-time learning, they need to be emotionally stable and in good health [3]. In fact, anxiety over homework, exams, and coursework has a detrimental effect on students academic performance in reading, science, and arithmetic, according to an OECD survey [4]. Prior studies have demonstrated a direct correlation between the degrees of student participation and the experience of both pleasant and negative emotions. Positive emotions were linked to higher levels of student involvement among 293 American students in grades 7–10 during class. On the other hand, decreased engagement was linked to the frequency of unpleasant emotions [5]. This finding is significant because, as the results of a study performed by the National Union of Students demonstrate, learning involvement is essential for achievement. According to this poll, stress was the main factor impacting Australian university students' tertiary education between the ages of 17 and 25 [6]. Additionally, studies have shown that data-driven decisions in education can improve learning outcomes [7].

According to Auvinen et al [8], progress visualization may enhance student behavior and motivation. VisUAM aims to provide students with an in-depth overview of their academic journey, allowing them to make informed decisions about time allocation and study techniques. As a result, having a clear visual representation of their progress allows students to maximize their efforts and eventually achieve better performance and greater satisfaction with their learning experience.

Furthermore, this project seeks to provide the UAM with an alternate perspective on learning analytics research. The purpose of this application is to improve teaching, learning, and learning environments by leveraging educational data. Learning analytics are being used in various educational areas, including early childhood and graduate school [9], and the UAM will not be excluded from this conversation.

This initiative, in its essence, represents the nexus of technological innovation and educational theory, demonstrating how technology may support pedagogical practices and empower teachers and students to pursue academic achievement.

1.2. Objectives

Before and during development, a series of milestones were established in order to facilitate the development, but always with a main objective that was to create a fully functional web application that allows students and teachers through the Moodle platform to watch the academic progress of each participant of a course. During the analysis of the requirements of the application, the following objectives were defined:

O1 Integrate a web application in a Moodle Course.

O2 Extract a Moodle course information using web scrapping techniques.

O3 Design a distributed system architecture for the application that ensures portability and resilience.

O4 Release the application into a real Moodle course from UAM and test the application with real students.

In the following chapter, the state of the art will be discussed, where we will take a deep dive into technologies that were key part of the project, among these technologies we will find Django, Django Channels, Docker, etc. Moreover, in chapter 3 we will take a deep look into the application, and we will explain the different parts of the application, such as the server architecture, the application components, and the database. In chapter 4, we will go further into the many environments we utilize to test the application. We will also cover every test phase's defined objectives for every facet, as well as the tests conducted to meet the pre-established objectives. Furthermore, all of the findings and conclusions drawn from the examination of the tests will be presented. Chapter 5 will provide a comprehensive explanation of all the conclusions drawn from the project, including the aims attained and the ongoing work that this project left behind.

2 State of the Art

In this chapter, the study's topic, learning analytics, will be explained and presented, as well as how our application makes use of numerous components of this discipline. The dashboard concept will also be introduced, along with the many categories that may exist. Finally, we will go over all the technologies used in this application, as well as the key factors that affected their selection for development.

2.1. Learning Analytics

Auvinen et al [9] define Learning Analytics (LA) as "the measurement, collection, analysis, and reporting of data about learners and their contexts". According to this explanation, LA does not have its own data; rather, it exists to manage and assess other substantial datasets of learners and associated meta-data for their benefit. Other authors such as del Blanco et al [10] defined LA as *"a discipline that gathered and analyzed educational data with different purposes, such as seeking a pattern in the learning process and trends or problems in student performance"*.

Auvinen et al [9] provide a classification of data utilized in LA models, which is as follows:

Demographic information: This information consists of age, gender, location, birthplace, level of education, and employment status.

Behavioral data: This includes recordings of group and individual activities on video and audio, as well as data on student interactions from blogs, online activities, and discussion forums.

Assessment data: This comprises information from tests, assessments, and coursework.

Financial data: This is the learner's financial data store, containing funding sources, free status, tuition updates, and support allowances.

Historical progression data: This is the data stored about students after they graduate and includes alumni, employment information, location, forwarding address, and contact information, as well as degree type and class. [9]

Despite the fact that learning analytics is still in its early stages, companies like SEAtS Software, which specializes in developing software for educational institutions, have created a range of student-focused software solutions using actionable education analytics to assist students in making decisions at all levels more quickly and efficiently. The company's Student Engagement Analytics solution caught our attention since it aims to pinpoint areas where students struggle, how they learn, and how to use student engagement analytics to help their achievement [11]. As we can see in the figure 2.1, this solution contains a number of features that are similar to our planned application, as well as some additional intriguing features that may be added in a later edition. Furthermore, universities and businesses that are funding SEAtS projects, among them Cambridge University, London Kingston University, and Unitec Institute of Technology, show the bright future of these kinds of applications. Software products similar to the one described in this section are also supported by businesses such as Microsoft, Juniper Mist, and Canvas LMS.

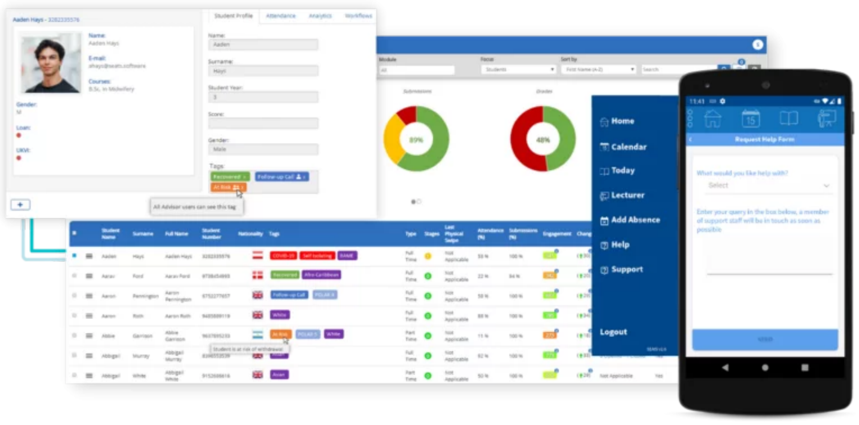


Figure 2.1: SEAtS Student Engagement Analytics solution

2.2. Learning Dashboards

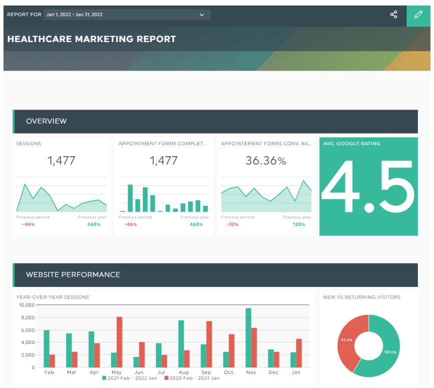
Dashboards are where users will engage with our application, therefore they are an essential component of this project. A dashboard is a visual representation of the most important data needed to achieve one or more goals, they can be defined as consolidated and presented on a single screen for easy monitoring [12]. A very practical way to classify the different types of dashboards is to enclose them by the role that they accomplish, Based on Stephen Phew's essay [12], we identified the following classifications.

Strategic Dashboards: they are the most common type of dashboard utilized today. They give decision-makers the concise summary they require to keep an eye on the company's prospects and overall health. This kind of dashboard focuses on simple performance evaluations (good and bad) and high-level performance metrics that might benefit from contextual information to help clarify the meaning, such as comparisons to targets and brief histories. The finest display mechanisms for this kind of dashboard are quite basic ones. These dashboards work best with static pictures taken on a monthly, weekly, or daily basis, as they provide long-term strategic guidance rather than requiring quick responses to constantly changing circumstances. Finally, they are often unidirectional displays that just show the current situation.

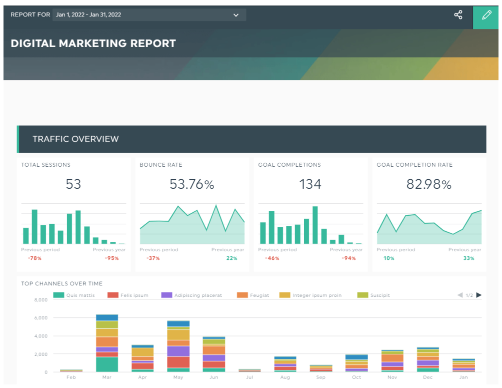
Analytical dashboards: these dashboards are quite similar to Strategic Dashboards, with the exception that they provide extra context, such as comparisons, historical data, and performance ratings. Analytic dashboards, like strategic dashboards, benefit from static data snapshots that are consistent throughout time. Sophisticated display media might be useful for analysts who desire advanced data analysis and are willing to learn how to use it. Analytical dashboards should allow for data exploration, including drilling down to underlying information, in order to make sense of the data and pinpoint core causes.

Operational Dashboards: Dashboards for monitoring operations must be developed differently from those for strategic decision-making or data analysis. The dynamic and instantaneous nature of operations has a significant impact on dashboard design. When monitoring operations, it's important to be aware of changing actions and occurrences that may demand immediate attention, such as when a robotic arm in a factory assembly line runs out of bolts to attach a vehicle door to the chassis. The display media on operational dashboards, like those on strategic dashboards, must be very clear in order to reduce errors and increase response time for the operator side. In contrast to strategic dashboards, operational dashboards must have the means to grab your attention immediately if an operation falls outside the acceptable threshold of performance. Also, the information that appears on operational dashboards is often more specific, providing a deeper level of detail.

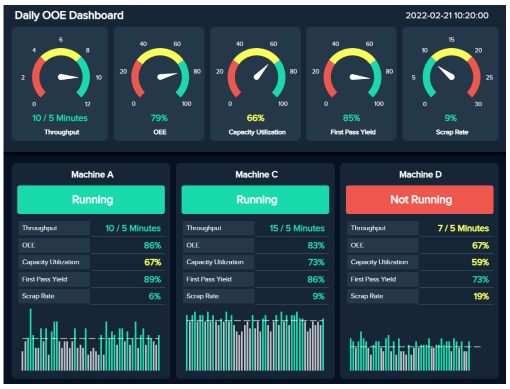
As previously said, the world of dashboards is vast and can bring many elements into play, but our application includes a strategic dashboard tailored to the educational setting. In this scenario, the dashboards that we will utilize in the application are educational dashboards. These dashboards are tools that help users uncover patterns, correlations, and pertinent data among a group of learning variables. However, in an educational context, various roles can be involved, ranging from students to professors, heads of study, and directors, among others. Each role has unique data exploration objectives based on their demands [13]. Dashboards in education can have several aims, including self-monitoring, monitoring of peers, and administrative monitoring, but certainly, each of this dashboard will depend on the type of role that the final user will have (student, teacher, between others). Moreover, it is more common that educational dashboards are more aimed at the teachers or course administrators than the students, the more common visualization elements being the bar diagrams, line diagrams and tables, which may be due to its simplicity [14].


REPORT FOR Jan 1, 2022 - Jan 31, 2022
HEALTHCARE MARKETING REPORT
OVERVIEW
SESSIONS
1,477
APPOINTMENT FORMS COMPLET...
1,477
APPOINTEMENT FORMS CONV. RA...
36.36%
AVG. GOOGLE RATING
4.5
WEBSITE PERFORMANCE
YEAR-OVER-YEAR SESSIONS
NEW VS RETURNING VISITORS

(a) Example of a strategic dashboard



(b) Example of a analytical dashboard



(c) Example of a operational dashboard

Figure 2.2: Examples of different types of dashboards

Everything mentioned above was extremely fascinating and helpful in the development of our application, especially when some of the previously mentioned information is reflected in educational software like the SEAtS software, which was previously mentioned and which uses learning analytics and educational data to create a number of educational dashboards.

2.3. Moodle

Moodle is a secure and integrated learning platform that allows instructors, administrators, and students to create personalized learning environments [15]. Furthermore, moodle is an open-source learning management system (LMS) with many customization options, including plugins available in a plugin shop. These plugins are developed and published by the same Moodle users, allowing projects like the one described in this paper to be implemented and made available to other users.

2.4. Development Refereces

In this section of the document, the different technologies that made possible the development of the VisUAM application will be exposed. There will be a brief introduction to each of the technologies listed, along with an explanation of their respective roles within the application and the justification behind their selection over alternatives.

2.4.1. Nginx

For web serving, reverse proxying, caching, load balancing, video streaming, and other uses, NGINX is available as open-source software. It began life as a web server intended for peak stability and performance. NGINX is not only an HTTP server; it can also be used as a reverse proxy and load balancer for HTTP, TCP, and UDP servers, as well as a proxy server for email (IMAP, POP3, and SMTP).

NGINX, in conjunction with the future technology (certbot), is responsible for delivering HTTPS connections to end users. NGINX was chosen for development because it was easy to install and configure in an Ubuntu server environment, but it is also a highly strong free resource that provides additional security and performance to the application.

2.4.2. CertBot

A component of Electronic Frontier Foundation (EFF) mission to encrypt the Internet is Certbot. HTTPS is the foundation for secure communication over the Internet. It requires the usage of a digital certificate, which enables browsers to authenticate web servers (e.g., is that really google.com?). Certificate authorities (CAs) are reliable third parties from whom web servers receive their certifications. Certbot is a user-friendly client that installs a certificate on a web server after retrieving it from Let's Encrypt, an open certificate authority founded by Mozilla, the Electronic Frontier Foundation, and others [16].

Among other reasons, Certbot was selected for this application because it eliminates the hassle associated with configuring and preserving a secure website certificate. All the aforementioned tasks are automated by Certbot and Let's Encrypt, enabling you to activate and control HTTPS with a few easy commands.

2.4.3. Django

Django is the principal technology used in this project. Django is a framework that allows the development of web applications. This framework is based on the Model-View-Controller (MVC) design pattern. Another useful feature of Django is that it allows the local testing of our web application using the Web Server Gateway Interface (WSGI), Additionally, Django has multiple add-ons that lead to more out-of-the-box developments, like Django channels, that will be further explained.

The reason this framework was chosen over others, like the Microsoft MVC framework.NET, is that Django is built on Python, a very strong yet user-friendly programming language with numerous modules that facilitate the creation of applications. Additionally, the development of this application was made possible by a number of Django projects and services, including the following:

Gunicorn

Gunicorn, often known as 'Green Unicorn', is a UNIX Python WSGI HTTP server. This worker model is a pre-fork that was ported from Ruby's Unicorn project. The Gunicorn server is easy to use, has low server resources, is fairly quick, and is widely compatible with a variety of web frameworks [17].

In or application, gunicorn will serve HTTPS connection for users, is the part of the application that will handle all request and send responses to users as we can see on the chapter 3. Gunicorn is capable of handling HTTPS configuration thanks to the certificates generated with certbot (explained in the section [16])

Django Channels

This is a Django project that aims to extend the Django capabilities to handle HTTP, and it allows the management of different long-running connection protocols such as MQTT, Web Socket, etc. Django Channels runs together with the Django native Asynchronous Server Gateway Interface (ASGI), a similar interface to WSGI [18]. In order to manage the connections and tasks, Django channels uses Daphne, this is a HTTP, HTTP2 and Web Socket protocol server for ASGI and ASGI-HTTP, developed to power Django Channels [19].

The web socket that connected the web scrapper to the main Django application was created using Django channels; this technology was really useful because it simplifies the entire web socket development and connection process. Django channels also manages any concurrency issues that the socket may encounter, which was one of the primary reasons for selecting this project to handle the application's web socket development.

2.4.4. Selenium

Selenium is a project that encompasses a variety of tools and libraries for automating web browsers. It provides add-ons a that simulate user interaction with browsers. Selenium is built around WebDriver, an interface for writing instruction sets that may be run in a variety of browsers.

Selenium is a project that encompasses a number of tools and frameworks used to automate web browsers. It offers plugins that simulate user interaction with browsers. Selenium's fundamental component is WebDriver, an interface for writing instruction sets that may be run in a variety of browsers.

3

VisUAM Aplication

In this section, we will go deeply into how our application is built. First, the program's architecture will be explained, along with each component and its role in the application. Finally, an overview of all of the dashboards will be provided, along with an explanation of their functionality.

3.1. Architecture

The VisTool application, is a light-weighted client web application, that is available for the Chrome web browser. As we will see in the following sections of this chapter, this application has different independent modules that communicate through WSS and HTTPS protocols.

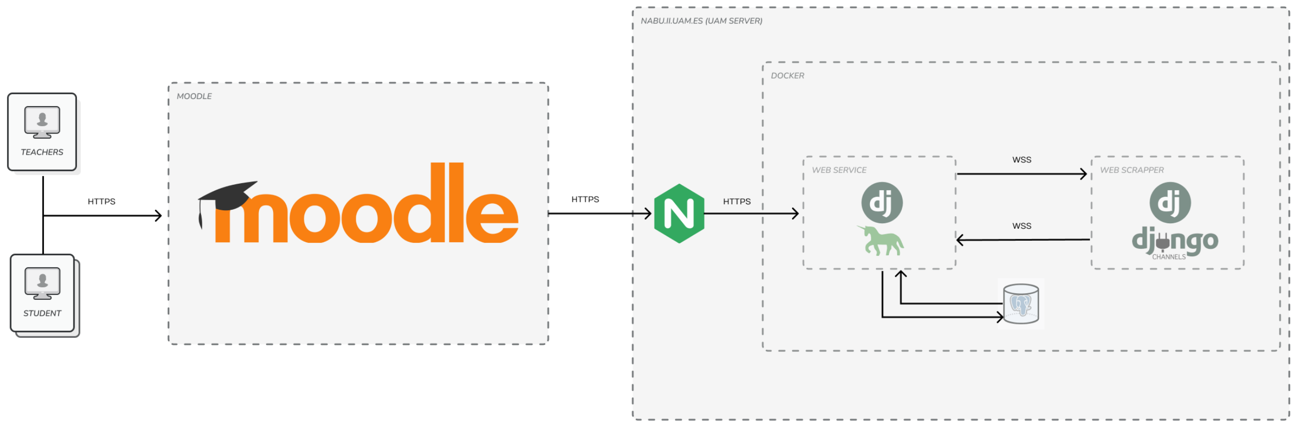


Figure 3.1: Architecture of the application

As we can see in the Figure 3.1, this application has different independent parts that will be explained in the following sections. It is important to denote that all the application is running inside Docker in independent container that are all connected through a docker network. More over, this docker is installed in a remote UAM server, and this server serves HTTPS connections using a NGINX reverse proxy (which was configured as part of the project).

In order to establish secure connections, we need to ensure that our server does a proper TLS protocol with the clients, that is why we need to create and install digital certificates to the server, therefore, we use certbot top fulfill this task.

Up next, each individual part of the application will be explained.

3.1.1. Web Service

This part of the application is the one responsible for ’front-end’ and ’back-end’ of our application. Each function is carried out by two native Django applications.

The front-end part of the application manages the requests using Gunicon, which applies the TLS (using the same certificates mentioned in Section 3.1) and HTTP 2.0 protocols, to ensure a good and secure connection.

3.1.2. Web Scraper

In the project the web scrapper is an independent synchronous job that will search, download and format all the information of the course that is in the moodle page.

This part of the project was developed using the Django Channels project. This project extends the Django abilities for managing HTTP and allows the management of WS communication protocol.

More over, the web scrapper handles the communications via events and handlers, is similar to a publisher consumer queue. All of this process is asynchronous and in a case that multiple request reached the web scrapper at the same time, the first one to reach is the one that will be attended and the others clients will be pushed into a waiting queue (all of this process is managed internally by the channel layers (agregar referencia a la explicacion de lo schannel layers). All the web scrapping is done with selenium

3.1.3. Database

To allow the persistence of the data within the project, a PostgreSQL database was chosen. In the following chapter, the database design will be shown and explained.

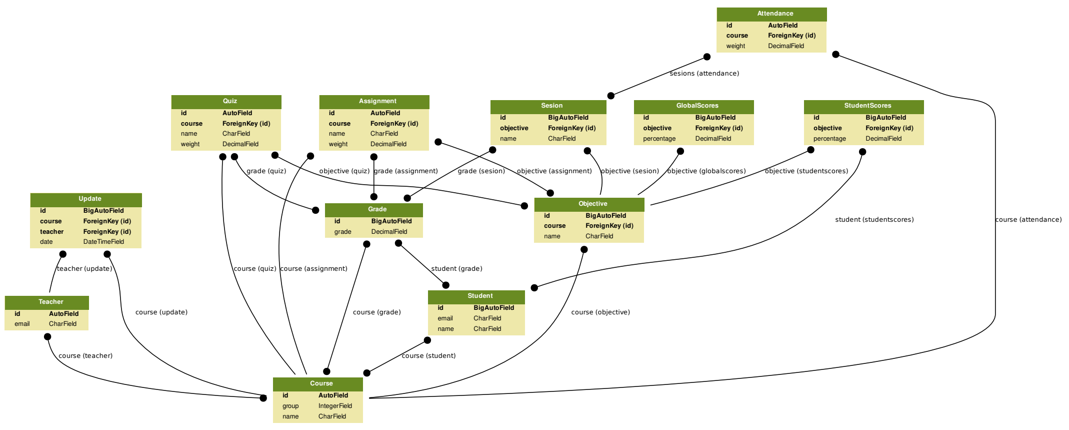


Figure 3.2: Database Diagram

Course

This table stores all the courses registered in the application. All groups in the application have a tuple key, because the same group name may exist in many courses, but the program does not allow two groups with the same name to coexist in the same course. The previously described key includes the group identification **"group"** (in the UAM, all groups are numbers, but if the application migrates to other universities, this field may need to be rethought) and the course name field **"name"**.

Student

The Student table, as the name implies, is in charge of storing all students inside a course. One interesting feature of this table is that all queries will be indexed by the **"email"** field, thus this field has an unique constraint in order to avoid repetitions.

Objective

This is one of our database's key tables, and its purpose is to record all the objectives that a course has (because, in the end, the objectives are the progress bars that each student or teacher will view in the application dashboard). In this table, objectives are recognized by their name (stored in the field **"name"**) and the course to which they belong (stored in the field **"course"**), which refers to the course table data. The objectives, like the course table, include a tuple key, which allows the objective name to be repeated only if it is in a different group or course.

GlobalScores

All of the students' progress percentages throughout a course are kept in this table. This table is essential to preventing pointless load screens and inefficient dashboard page transitions. In longer courses, this performance detail will be more evident. This table has a **"id"** field, a reference to the corresponding objective entry (which will be kept in the **"objective"** field), and a field for the percentage of progression that was accomplished (which will be kept in the **"percentage"** field). The objective table already contains that constraint, therefore there's no need to add the course as a relation for the table.

StudentScores

This table serves a very similar purpose to the previously described GlobalScores table, with the exception that it keeps each student's individual progress inside a course, or, to put it another way, the progress percentage for every objective for each student in a course. This table bears little structural differences from the GlobalScores table; the addition of a reference to the student table, which will be saved in the field **"student"**, allows for the relationship between the student and his or her relevant progress percentage.

Quiz and Assignmets

During development, it was discovered that the Quiz and Assignment activities were very similar, at least in terms of data, thus explaining them in one section will be useful.

These two tables store the quiz and assignment information from Moodle, respectively. Both have a unique identification field named **"id"**. A **"name"** field that will store the activity name, which will be the same as the one that appears on the Moodle course.The course and objective will be stored in the fields **"course"** and **"objective"**, respectively. Both of these columns make reference to the data held in the course and objective tables. Furthermore, in order to calculate the progression, the grade of these activities is needed, so they will be stored in the grade table explained up next. However, each activity stored in this table will have its respective grade assigned, and it will be done by the **"grade"** field, which will be a reference to its respective entry in the grade table.

Finally, when the course is configured, the teacher will manually assign a **"weight"** field to each activity. This weight represents the objective or objectives assigned to the activity. It is worth noting that one activity can have many objectives by making a tuple key with the **"course"** and **"name"** fields; this feature is quite valuable for teachers because numerous objectives are typically examined in a single assignment or quiz.

Attendance

There is an attendance activity in our application as well. The only distinction between this activity and the previous quiz and assignment activities is that each attendance activity includes a set of session activities that will be clarified at a later time. Every activity has three fields: **çourse"**, **weight"**, and identification (**ïd"** field). The course table is referenced in the "**course**" field, just like it is in the other activities table.

Sesions

ºThis table supplements the previously mentioned attendance table; it is simple, but it is essential for determining student attendance in a course. When you create the course, each session will have an identifier "**id**" and a "**name**" that will be the same as the session name in the Moodle course. In addition, each table entry (each session) will have the previously assigned objective, which will be in the "**objective**" field, which is a reference to the objective table.

Furthermore, each session will have a grade assigned to it through the "**grade**" field, and each grade will have a weight, which will be stored in the "**weight**" field. As a result, the attendance activity will have its desired grade, which will be calculated with the median of all the grades of the assistance inside each attendance activity. It is worth to mention that the "**grade**" field contains a reference to an entry in the grade table.

Grade

This is one of our database's most crucial tables; without it, we will be unable to calculate student progression. As the name implies, this table will contain all the grades from all the activities that each student completes during a course. This table contains a numerical field for storing grades (the "**grade**" field), as well as references to the student and course tables in the fields "**student**" and "**course**", respectively. This allows a single student to have multiple grades within one course.

Update

This table contains three fields: "**course**", "**teacher**", and "**date**". Despite its simplicity, it plays a significant purpose. After changing course information, the date of the modification is saved in the "**date**" field. The data stored in this table will be presented on the main student dashboard and made available to all course participants.

Teacher

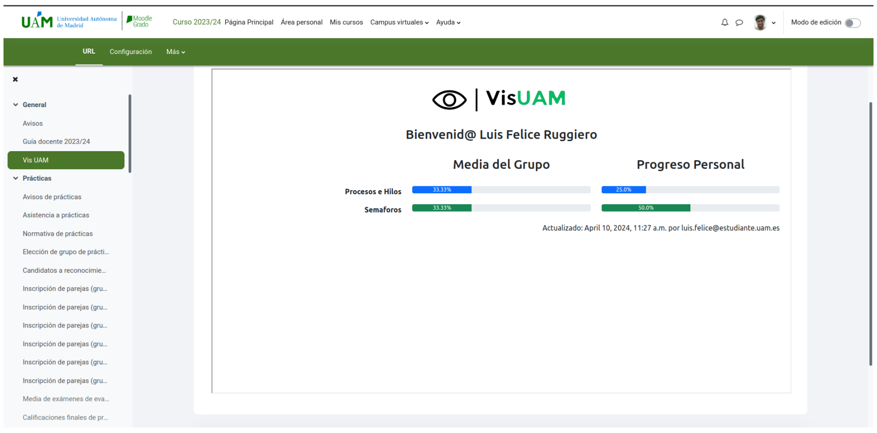
This table, as the name implies, contains all the teachers registered in the application. Teachers, like students, will be identified using the email address stored in the **“email”** field. To prevent the same teacher from appearing in the same course multiple times, a tuple key with fields **“email”** and **“course”** is used to store the course in which the teacher is registered (the **“course”** is a reference to the course table).

3.2. Dashboards

There are various dashboards in the VisUAM program, some of which are designed exclusively for teachers and others for students. While teacher dashboards are made for configuring and viewing the progress of every student enrolled in the course, student dashboards are simply intended to visualize the individual progress of each student. Let’s start by visualizing the student’s dashboard:

3.2.1. Student Dashboard

Students will have available only one non-interactive interface where they will see the progress of each one of the objectives that the teacher planned at the beginning of the course.


UAM Universidad Autónoma de Madrid
Moodle Grado
Curso 2023/24
Página Principal
Área personal
Mis cursos
Campus virtuales
Ayuda
Modo de edición
URL
Configuración
Más
General
Avisos
Guía docente 2023/24
Vis UAM
Prácticas
Avisos de prácticas
Asistencia a prácticas
Normativa de prácticas
Elección de grupo de prácti...
Candidatos a reconocimie...
Inscripción de parejas (gru...
Inscripción de parejas (gru...
Inscripción de parejas (gru...
Inscripción de parejas (gru...
Inscripción de parejas (gru...
Inscripción de parejas (gru...
Media de exámenes de eva...
Calificaciones finales de pr...
VisUAM

Bienvenid@ Luis Felice Ruggiero

	Media del Grupo	Progreso Personal
Procesos e Hilos	33.33%	25.0%
Semaforos	33.33%	50.0%

Actualizado: April 10, 2024, 11:27 a.m. por luis.felice@estudiante.uam.es

Figure 3.3: Student Dashboard

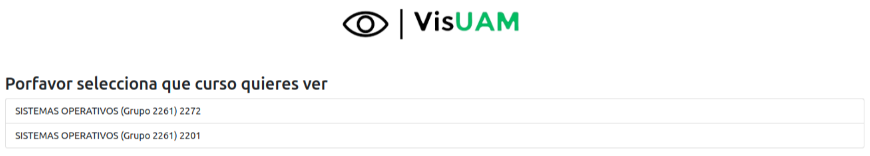
The student will see their progress as well as the progress of the entire group, as shown in the Figure 3.3. Each bar represents an objective that the teacher configures at the beginning of the course, and the student can use this information to compare their performance to that of the group.

This progress indicated in this section will be the mean of all the grades of activities for each aim. Depending on the results of this calculation, each student’s progress bar will have a different color. For example, if the student’s progression is 4 points below the average, the bar will be red; if the student’s progression is above the average, the bar will show a green color to the student; and finally, if the student's progression is close to the average progression of the course (1 and a half points above or below), the bar will be blue.

Finally, below all bars, students can see the last time the course information was updated, allowing them to know if the progress is up-to-date or not.

3.2.2. Teacher Dashboard

As we said previously, this dashboards where thinked to let the teacher visualize and configure each element of the course.



Porfavor selecciona que curso quieres ver

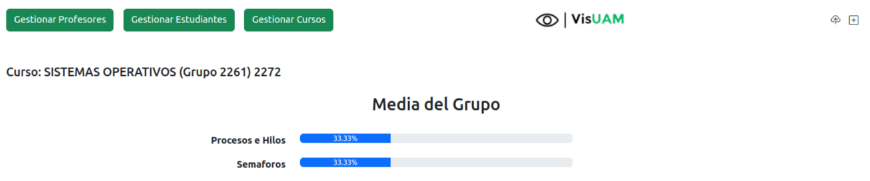
SISTEMAS OPERATIVOS (Grupo 2261) 2272

SISTEMAS OPERATIVOS (Grupo 2261) 2201

Figure 3.4: Teacher Landing Page

What we are seeing in the figure 3.4 is the landing page that the teachers will see if they have more than one course configure in the application. The objective of this page is to let the teacher choose which course they want to visualize. Once we choose the desire curse we will enter the main administration dashboard that we can see in the following figure.

As previously mentioned, the figure 3.5 shows the main page for teachers. In this page, we can mainly see the progress that all the students in our class have. As explained above, this progress is the average of all the grades of the activities that a course objective contains. This dashboard also has a navigation bar that has several options. Firstly, let us talk about the three green buttons that are on the left of the navigation bar; these buttons are meant to let the teacher manage all participants or change the current course instead. If we click on the first button labeled *Gestionar Profesores* we will be able to see what is presented on the figure 3.6.



Curso: SISTEMAS OPERATIVOS (Grupo 2261) 2272

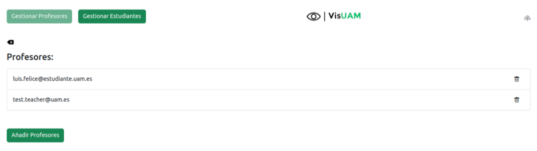
Media del Grupo

Procesos e Hilos 33.33%

Semaforos 33.33%

Figure 3.5: Teacher main dashboard.

In this teacher configuration page, we will be able to see a list of all the teachers that are part of the course, and also we will have the option to add new teachers to the course by clicking the button *Añadir Profesores*, which is located at the bottom of the page. The dashboard for adding teachers will be shown below in the figure 3.6(a).

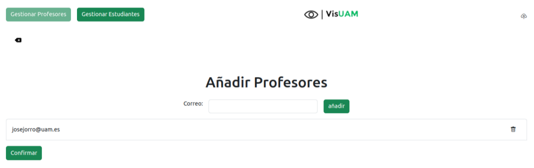


Profesores:

luis.felice@estudiante.uam.es

test.teacher@uam.es

(a) Dashboard that allows manage all the teachers enrolled in the course.

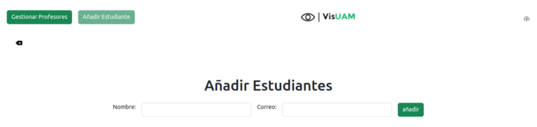


josejorro@uam.es

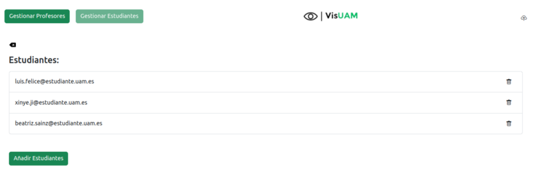
(b) Dashbard that allows the addition of a new teachers to the course

Figure 3.6: Screenshots of the teacher management dashboards.

In the Figure 3.7(a) we can see where the *Gestionar Estudiantes* button takes us. In this page, the teacher will be able to see a list of all the students enrolled on the course and also lets the teacher add or remove students from the course, in order to avoid the teacher needs to reconfigure the hole course if an error exists in the initial configuration. In the Figure 3.7(b) is shown the page where the teacher is able to add students.



(a) Dashboard that allows manage all the students enrolled in the course.



Estudiantes:

luis.felice@estudiante.uam.es

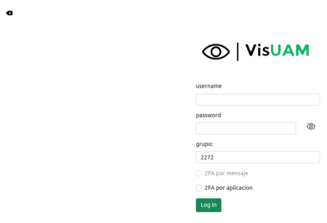
xinye.ji@estudiante.uam.es

beatriz.sainz@estudiante.uam.es

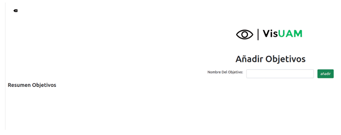
(b) Dashbard that allows the addition of a new students to the course

Figure 3.7: Screenshots of the student management dashboards.

Finally, we have to take a look again into Figure 3.5. In the navigation bar, on the right side, there are two icons left. The first icon (if we look at the image from left to right) is the one that a teacher would select, in order to trigger an update in the course, this button will take us to a page similar to the one that we can find in the user manual in the appendix A, this page can be seen at Figure 3.8(a), and it will request the teacher credentials in order to update the course information. On the other hand, we have the page shown in the Figure 3.8(b), this page allows the teacher to add new objectives after the course configuration; because the teacher is changing some configuration of the course, it will have to do an update operation after the addition o the new objectives.



(a) Dashboard that allows to create a new course



(b) Dashbard that allows the addition of new objectives to the new course

Figure 3.8: Screenshots of the new course creation dashboard and the dashboards that allows teacher to define the objectives to the new course.

4 System Evaluation

In this area, an overview of all the tests run on the application will be displayed. Each of the test phases will detail the many objectives and tests that were considered, as well as the outcomes and conclusions, decisions, or improvements that resulted from them.

4.1. Local Testing

This section will cover the goals of the evaluation of the application's first version as well as the outcomes of this test phase. It is important to note that the Bitnami/moodle image and docker were used to create a simulated Moodle server for this test phase. Every piece of activity and student data was generated artificially to mimic actual user involvement in a Moodle course.

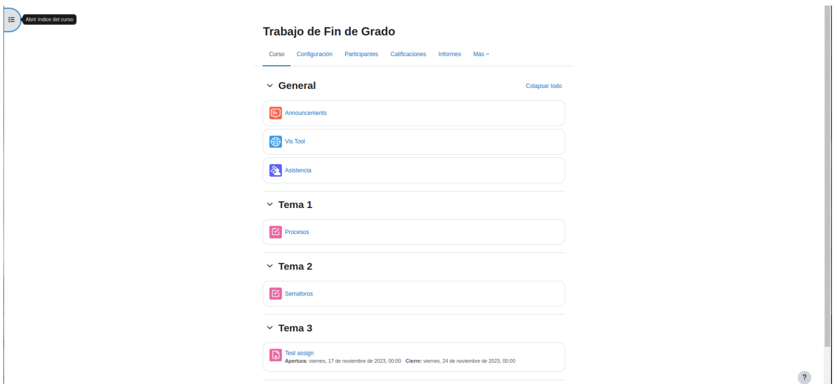
4.1.1. Environment Setup

As we mentioned before, these tests were designed to evaluate how the application performed in a Moodle environment. For this, a docker image developed by bitnami [1] (bitnami/moodle) was used. Once the mock server was running a course was created with a series of activities as we can see in the Figure 4.1. Is worth to mentioned that all the participants and activities

Following the setup of the previously described test environment, a number of goals were established. The goal of this test phase is to assess how well the application works in a Moodle environment by watching how the web scrapper uses the data from Moodle and how the dashboards handle and present the data that the web scrapping module has received. The following goals were intended to be documented in this assessment:

[1] https://hub.docker.com/r/bitnami/moodle/

- Successfully integrate the VisUAM project to a Moodle environment.
  - Prove that students and teachers can access to each as they desire dashboards.
  - Prove that the extracted Moodle data was available in the different dashboards for its visualization.
- Successfully extract and process the course information using web scrapping techniques.



Trabajo de Fin de Grado

General

Announcements

Vis Tool

Asistencia

Tema 1

Procesos

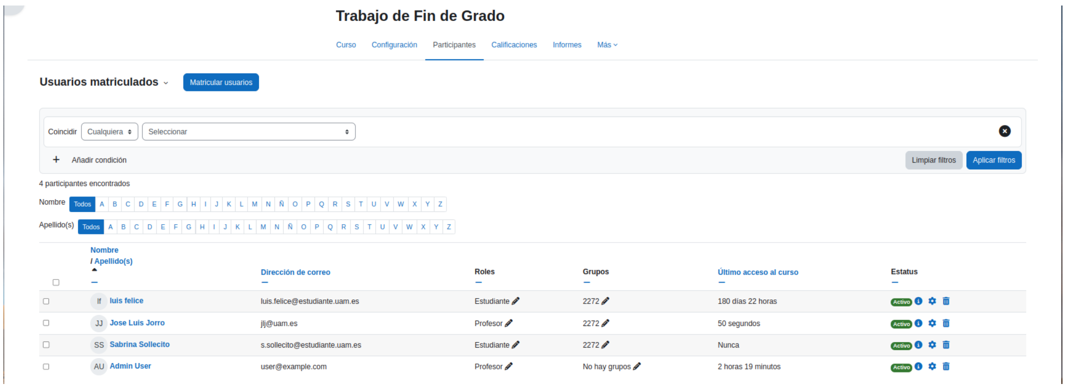
Tema 2

Semaforos

Tema 3

Test assign
Apertura: viernes, 17 de noviembre de 2023, 00:00 **Cierre:** viernes, 24 de noviembre de 2023, 00:00

(a) Acticvities in the Moodle test server.



Trabajo de Fin de Grado

Usuarios matriculados

4 participantes encontrados

Nombre / Apellido(s)	Dirección de correo	Roles	Grupos	Último acceso al curso	Estatus
lf luis felice	luis.felice@estudiante.uam.es	Estudiante	2272	180 días 22 horas	Activo
JJ Jose Luis Jorro	jlj@uam.es	Profesor	2272	50 segundos	Activo
SS Sabrina Sollecito	s.sollecito@estudiante.uam.es	Estudiante	2272	Nunca	Activo
AU Admin User	user@example.com	Profesor	No hay grupos	2 horas 19 minutos	Activo

(b) Participants in the Moodle test server.

Figure 4.1: Test Moodle course created inside docker.

4.1.2. Tests

A number of tests were run to confirm and enhance the application's functions in order to meet the goals outlined in the preceding section. The following tests have been run:

LT1 The application can be embedded in a Moodle environment.

LT2 A teacher can create a course in the application.

  LT2.1 A teacher can create a objectives in the course.

  LT2.2 The web scrapper can download the information of the Moodle course.

LT3 Teachers and students can access and use their dashboards.

LT4 A teacher can add or delete students to a existing course.

LT5 A teacher can add or delete teachers to the course.

LT6 The application handles multiple groups of the same course.

4.1.3. Results

Upon successful completion of all the tests presented in section 4.1.2, the initial version of the program was released. To delve deeper into the test details, we can observe on page 4 of the appendix A that test LT1 teaches us about specific information like the course id, name, or user email. Another test that led to important project decisions was LT2.2. Normally, a web scraper would navigate through Web elements and extract their information. This was the initial plan when developing the web scrapper to extract the activities from the Moodle course, but it failed to mention that, should web scrapping adopt this approach, certain naming conventions that affect the end user experience would need to be established. Consequently, the web scrapper locates and downloads course logs in order to extract course information. Ultimately, through the LT3 test, we discovered something crucial that needs to be changed in our application. This is that, as we will see in the chapter 5.1, Moodle does not provide information regarding the user role. As a result, the UAM email identification is required to distinguish between students and teachers.

4.2. Evaluation in real environment

After completing the local tests described in the Chapter 4.1, we needed to see how the application works within a real course in Moodle. For this reason, a test course was created with the assistance of the project tutor to test our application; however, due to student data protection, a copy of the students data could not be uploaded to this test course; thus, again with the assistance of our tutor, the application could be tested in a real Moodle course with a real volume of data. The two eventualities will be detailed in further detail in the following section.

4.2.1. Moodle test course

As previously stated, this test course was designed to examine how the application reacted in a genuine Moodle context. The tests that were run in this environment were designed to test functionalities and identify critical points that could lead to a severe error in the application; additionally, these tests were designed to stress the application enough to break it and adjust it to the Moodle constraints; not to mention that these tests will also work to adapt the application to the Moodle environment, thereby resolving transportation errors.

Environment Setup

As with our local tests, our application was deployed on Docker, with the exception that executing the bitnami/moodle image will not be required. Further throughout the application deployment, this was done on a Linux server situated within the Escuela Politécnica Superior (EPS) instead of running the docker locally as explained in the Section 4.1.2; more over a version control was maintained using GitHub, allowing us to easily deploy new versions in the server. This test environment was designed as a test field for polishing the application before deploying it in the final course for test students to test (more on that later). To mimic the volume of data, several activities were created, and course participants were simulated with the assistance of EPS teachers and students. After clarifying this, the following objectives were established:

- Adapt the current version of the application to the Moodle environment.
- Test the performance of the new functionality in the version of the application.
- Find and solve all the errors/bugs produced by the Moodle environment.

Tests

As we previously mentioned, these tests were thinked to test the overall functionality of the application, therefore all the tests explained in the section 4.1.2 were also executed in this environment additionally to the following tests:

RTT1 The application can be embedded in a real Moodle environment.

RTT2 Test the application's response to a semi-real volume of data.

RTT3 Test the web scrapper in the UAM Moodle.

Results

The RTT1 test revealed that Moodle does not support cross-origin embedding of web services. This is why the NGINX reverse proxy was added to the application, as explained in the Chapter 3. Further into the tests, we discovered another very important critical failure point when executing the tests RTT2; this critically error consists of the URLs exceeding the default length limits put by NGINX; this was because, as it will be explained later in the limitations, all the course information between the web scrapper and the main application is passed through a URL; therefore, the NGINX configurations must be adapted to make the application work.

This step of testing was quite odd because we will repeat every time a new feature was introduced, therefore there are much less specific tests than in the Section 4.1. After the test phase was successfully finished, a new version of the application was deployed on the next environment, which will be discussed later, to conduct additional testing.

Finally, one of the most essential tests in this phase was the RTT3, because the login in the mock Moodle server has no additional security, whereas the Moodle server requires two-factor authentication to log in. Following this, the online scrapper has to be updated to accommodate the additional authentication by reading and sending the token to the login dashboard, allowing the instructor to enter the token into the Microsoft Authenticator app and the scrapper to read the course data.

4.2.2. Moodle real course

Finally, we arrived at the final test environment that approved our application; a real Moodle course with actual students was used. This test environment focuses on user experience rather than application functionality; during this test phase, the project tutor acted as a user and tested all teacher-related capabilities. In addition, the tutor instructed his students to use the program and provide feedback to help enhance it. A user satisfaction survey was created and published for this course to track results. The survey will be explained in the next section.

Tests

The difference between these tests and the others mentioned in this chapter is that all of them were created by the project tutor, as previously stated, within a real Moodle course with a real volume of activities and students, so these tests aimed to see if the application could handle all of this data volume. The following tests were created:

Satisfaction Survey

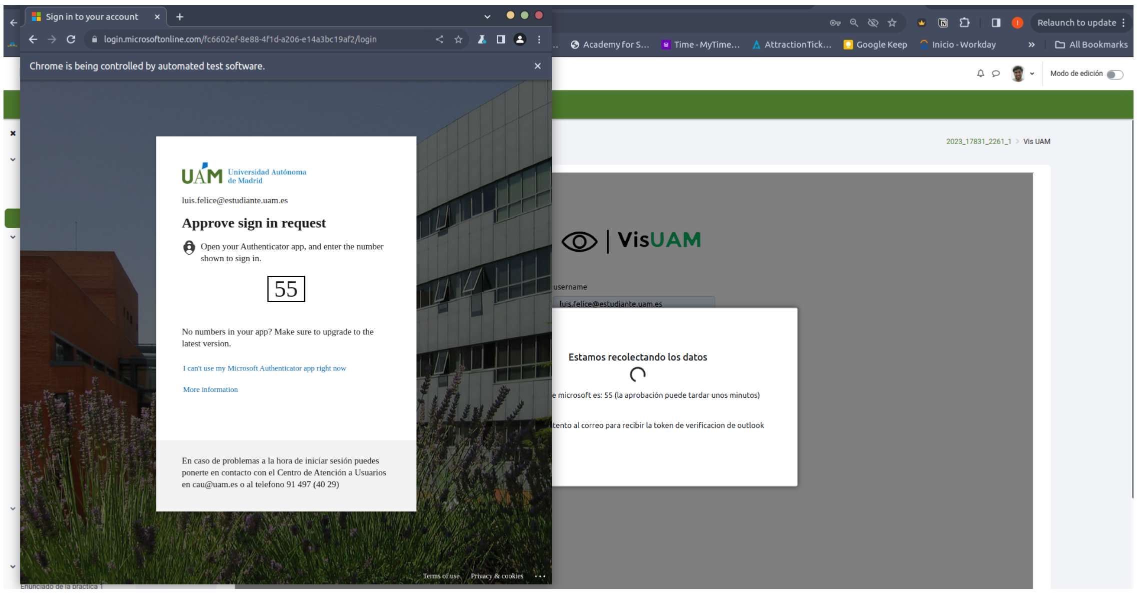
This survey was created to learn about the students' user experience while using the program. The primary goal of the study was to make significant modifications to the user interface in order to improve both user experience and application performance.

The survey questions could be responded in two ways, depending on the question. One option was a scale from 1 to 5 based on the user's pleasure; these questions were typically designed to determine what consumers thought about the user interface. On the other hand, other questions may be answered with a small amount of text, and these questions were designed to identify mistakes or improve program performance. All the survey questions are available in the Appendix B.

Results

In compared to the previous test phases, this phase was not as successful as predicted. On the one hand, this testing process has resulted in a fully functional application that teachers and students can use or integrate into Moodle. However, we were unable to improve our program as desired by the users because the students who served as test subjects participated in the satisfaction survey at a low rate, resulting in a non-liable result from which to draw conclusions.

Aside from the previously mentioned issue, this testing round resulted in a significant improvement. While doing the LT3 test in this environment, a major performance issue was discovered, which was connected to how we store and calculate the students' progress. At first, each time a user (teacher or student) loads a dashboard with a progress bar, all the calculations to show the results in the progress bar are performed at that time, resulting in a longer wait time and, in some cases, timeouts by the server, causing connection errors and crashing applications. This issue would be difficult to find in other test environments, as we said in the 5.1 section, due to the UAM's data protection policy, which makes it difficult to have the real data volume of a course. To solve this error, the database was redesigned until it reached its current state, as shown in the chapter 3.1.3, in order to store the calculation of each student's progress, which will now be done when the course was created, rather than each time the progress bars were displayed. This update has a significant positive impact on the user experience and application performance.



Chrome is being controlled by automated test software.

Universidad Autónoma de Madrid

luis.felice@estudiante.uam.es

Approve sign in request

Open your Authenticator app, and enter the number shown to sign in.

55

No numbers in your app? Make sure to upgrade to the latest version.

I can't use my Microsoft Authenticator app right now

More information

En caso de problemas a la hora de iniciar sesión puedes ponerte en contacto con el Centro de Atención a Usuarios en cau@uam.es o al telefono 91 497 (40 29)

VisUAM

username

luis.felice@estudiante.uam.es

Estamos recolectando los datos

e microsoft es: 55 (la aprobación puede tardar unos minutos)

tento al correo para recibir la token de verificacion de outlook

Figure 4.2: Web Scrapper reading the two-factor authentication token.

5

Conclusions and Future Work

This chapter will present the project's conclusions, stressing its primary objectives. Furthermore, all of the limits faced throughout the creation of the project will be presented, as well as the future work and evolution's that this project can have.

5.1. Limitations

Before we get into the project's results, it will be useful to discuss the various limitations discovered during the project's development.

During the project's analysis phase, one of the options for extracting information from Moodle courses was to use the API or web service option that Moodle provides to access the Moodle course database and extract data from the course; however, this option was quickly discarded because in order to enable the previously mentioned option, the UAM Moodle administrator had to enable the web service or API in each course that wished to use the application. Another option considered was to make the application an official Moodle plugging, so that no administrator would need to do any additional configurations and all teachers would have access to the application through the Moodle plugging store, but this option was not implemented because each Moodle plugging must go through a verification process, which would make meeting the desired project deadlines impossible. The web scrapper (described in Section 3.1.2) was designed to increase the application's complexity.

Another limitation was the UAM's data protection policy, which led in the establishment of two Moodle test courses, as shown in the Chapter 4.2. However, the majority of the problems that happened during the tests in the real Moodle course did not emerge during the tests in the Moodle course using simulated data. The key issue was that the project instructor was the only one having the authorizations to test the application in the actual Moodle course, causing delays in development because faults were resolved one at a time, and because we did not have the actual data volume, the analysis process that would lead to a solution to the problem was more difficult, resulting in delays in the delivery of a reliable version of the program.

5.2. Conclusions

As we observed at the start of this project, applications that incorporate learning analytics to assist students are becoming more prevalent, and they have the support of major tech companies such as Microsoft, which encourages us to continue developing this application. Although there is still work to be done to improve the application, some favorable feedback from EPS teachers encourages us to let it expand until it improves the course experience for numerous students.

A series of objectives were defined at the beginning of the project (Section 1.2), which were well completed. The objective O1 stated that we wanted to have an application that could be integrated with Moodle, so we created a web application that can be modified in Moodle; our application is contained in multiple docker containers, ensuring that it can be pulled from GitHub and executed on any server other than the one currently in use, thereby achieving the objective O2.

Aside from the goals met, the application is now being used in the operating systems course, specifically in the practical section, and it is being submitted as a component of an official UAM teaching innovation project. During the last month of instruction, a number of users used this program. However, because of certain issues that are detailed in Section 5.1, there were fewer users than anticipated during this test period.

Due to a lack of participation in the satisfaction survey, we were unable to demonstrate the positive impacts of progress visualization on students, as described in the Section 1.1.

If you are a teacher in the UAM, the program is available for your course, and the user manual, which includes installation instructions, is available in the Appendix A and the code is available in GitHub [1].

5.3. Future Work

The key task is to determine the effects of the application on student performance. This can be accomplished by administering the previously existing satisfaction survey to a larger set of students who have been using the application for at least half of a course. Another improvement that can be made to the project is to convert the program into a Moodle plugin; this would eliminate the need for a web scrapper because all information can be retrieved straight from Moodle and will make possible one functionality thought at the beginning of the project, which is that every time an activity reaches its due date, an automatic update of the course information is triggered; improving the teachers' users' experience.

[1] `https://github.com/LFelice2000/Vis_Tool/`

Another important evolution of the application is that, once a certain amount of data has been collected for each student, it would be very interesting to develop a recommendation system using machine learning to tell students what or which areas of the course they should reinforce in order to improve their performance even more.

Bibliography

[1] E. Commission, S. Directorate-General for Education, Youth, and Culture, *ECTS users' guide 2015*. Publications Office of the European Union, 2015.

[2] A. Bellogín, "Acto de bienvenida al mcurso 2023-2024."

[3] W. H. Organisation, "Health promoting schools," 1996.

[4] OECD, *PISA 2015 Results (Volume III)*. 2017.

[5] A. L. Reschly, E. S. Huebner, J. J. Appleton, and S. Antaramian, "Engagement as flourishing: The contribution of positive emotions and coping to adolescents' engagement at school and with learning," *Psychology in the Schools*, vol. 45, no. 5, pp. 419–431, 2008.

[6] D. RICKWOOD, N. Telford, S. O'Sullivan, D. CRISP, and R. Magyar, *National Union of Students National Tertiary Student Wellbeing Survey 2016*. Headspace, 2017.

[7] *Data-driven school improvement : linking data and learning*. Jan 2008.

[8] T. Auvinen, L. Hakulinen, and L. Malmi, "Increasing students' awareness of their behavior in online learning environments with visualizations and achievement badges," *IEEE Transactions on Learning Technologies*, vol. 8, no. 3, pp. 261–273, 2015.

[9] O. Adejo and T. Connolly, "Learning analytics in a shared-network educational environment: Ethical issues and countermeasures," *International journal of advanced computer science and applications: IJACSA*, vol. 8, no. 4, 2017.

[10] Á. del Blanco, Á. Serrano, M. Freire, I. Martínez-Ortiz, and B. Fernandez-Manjon, "E-learning standards and learning analytics. can data collection be improved by using standard data models?," *2013 IEEE Global Engineering Education Conference (EDUCON)*, pp. 1255–1261, 2013.

[11] "Student engagement."

[12] S. Few, "Information dashboard design : The effective visual communication of data / s. few.," 01 2006.

[13] "Beneficios de la aplicación del paradigma de líneas de productos software para generar dashboards en contextos educativos," Jul 2020.

[14] "Perceiving learning at a glance: A systematic literature review of learning dashboard research," Jan 2017.

[15] "Acerca de moodle - moodledocs," Jun 2024.

[16] "Introduction — certbot 2.10.0 documentation."

[17] "Gunicorn - wsgi server — gunicorn 22.0.0 documentation."

[18] "Django channels — channels 4.1.0 documentation," May 2024.

[19] "django/daphne," May 2024.

APPENDICES

A

User Mannual

Herramienta VisUAM
Manual de Usuario

Luis Felice
Escuela Politécnica Superior, UAM

June 10, 2024

1 Introducción

El siguiente manual está enfocado solo para los **profesores** que deseen instalar y utilizar la herramienta de visualización. Como se mencionó anteriormente, en este manual se expondrán los pasos a seguir para instalar satisfactoriamente la aplicación en un curso de Moodle de la Universidad Autónoma de Madrid (UAM). Sin más que agregar, les agradecemos por confiar y utilizar esta aplicación.

2 Requisitos Previos

- Ser profesor, con un correo de profesor válido (e.g. correo@uam.es).
- Tener permisos de edición en el curso en donde se desee instalar la aplicación.

3 Instalación

1. Buscar y acceder al curso de Moodle en donde se desee utilizar la herramienta.
2. Una vez en el curso, comprobar que el modo de edición está activado y se le debe dar clic a la opción ”Añadir una actividad o un recurso”

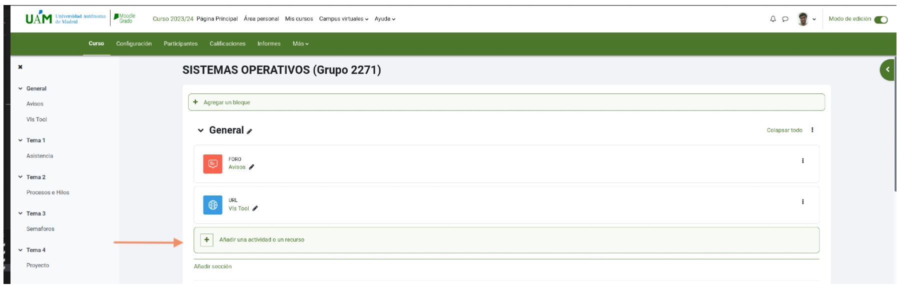


Figure 1: Seleccionando el recurso

3. Una vez en el menú, nos dirigimos a ”Recursos” y seleccionamos URL

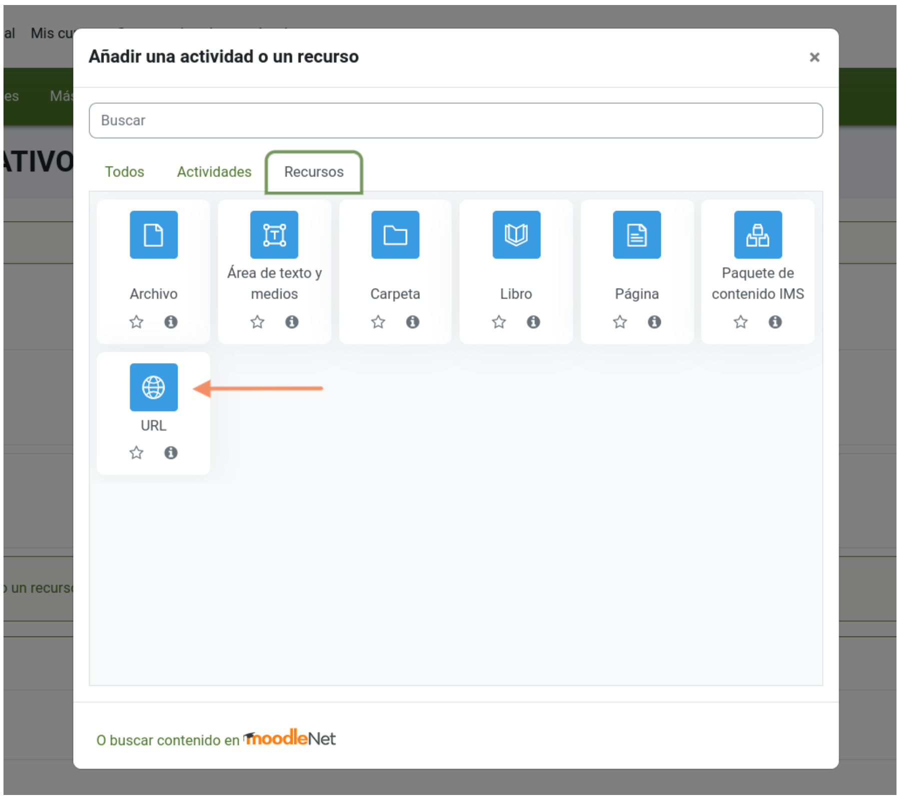
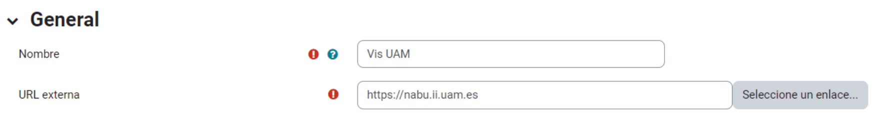


Figure 2: Seleccionando el recurso de url

4. Una vez en el menú de configuración, debemos darle los siguientes valores a las siguientes opciones:

   - Sección General:
     - **Nombre**: el que el usuario desee
     - **URL externa**: https://nabu.ii.uam.es



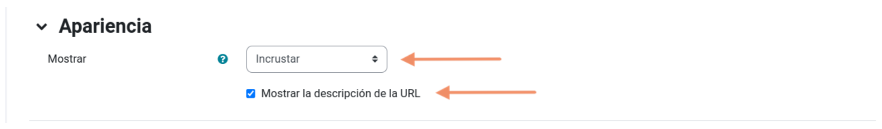
General

Nombre	Vis UAM
URL externa	https://nabu.ii.uam.es

Figure 3: Opciones generales de la aplicacion

   - Sección Apariencia:
     - **Mostrar**: incrustar

- Seleccionar el recuadro de **"Mostrar la descripción de la URL"**

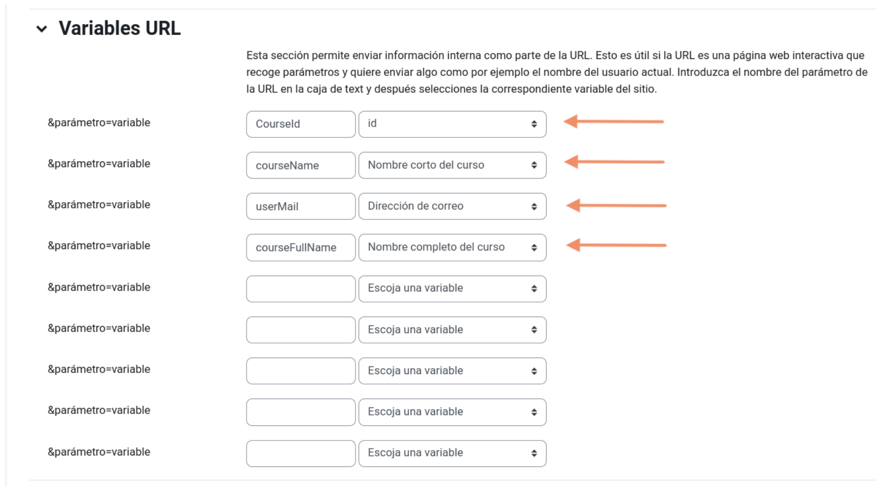


Apariencia

Mostrar	Incrustar
	✓ Mostrar la descripción de la URL

Figure 4: Configuración de la apariencia de la herramienta

- Sección Variables URL (**es importante escribir las variables tal y como están**):
  - **CourseId**: seleccionar la opción **id** bajo la sección **curso**.
  - **courseName**: seleccionar la opción **"Nombre corto del curso"** bajo la sección **curso**.
  - **userMail**: seleccionar la opción **"Dirección de correo"** bajo la sección **usuario**.
  - **courseFullName**: seleccionar la opción **"Nombre completo del curso"** bajo la sección **curso**.



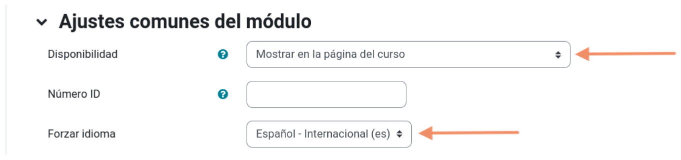
Variables URL

Esta sección permite enviar información interna como parte de la URL. Esto es útil si la URL es una página web interactiva que recoge parámetros y quiere enviar algo como por ejemplo el nombre del usuario actual. Introduzca el nombre del parámetro de la URL en la caja de text y después selecciones la correspondiente variable del sitio.

&parámetro=variable	CourseId	id
&parámetro=variable	courseName	Nombre corto del curso
&parámetro=variable	userMail	Dirección de correo
&parámetro=variable	courseFullName	Nombre completo del curso
&parámetro=variable		Escoja una variable
&parámetro=variable		Escoja una variable
&parámetro=variable		Escoja una variable
&parámetro=variable		Escoja una variable
&parámetro=variable		Escoja una variable

Figure 5: Configuración de las variables de la herramienta

- Ajustes comunes del módulo:
  - **Disponibilidad**: seleccionar la opción **Mostrar en la página del curso**.
  - **Forzar idioma**: seleccionar la opción **Español - Internacional (es)**.



Ajustes comunes del módulo

Disponibilidad: Mostrar en la página del curso

Número ID:

Forzar idioma: Español - Internacional (es)

Figure 6: Configuración de los ajustes comunes del módulo

Una vez que se haya terminado la configuración anterior correctamente, se podrá acceder a la aplicación donde se nos mostrara la pantalla de configuración, como podemos observar en la siguiente imagen:

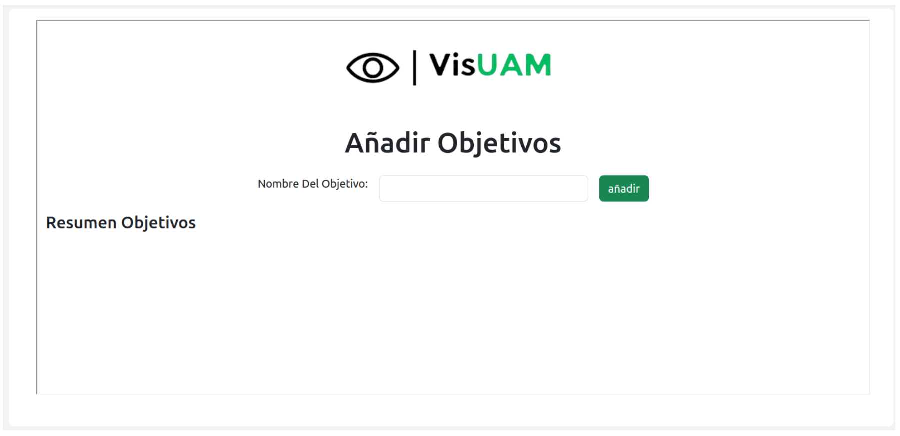


Figure 7: Pantalla de configuración de la aplicación

4 Utilización

La aplicación puede ser utilizada de dos maneras, dependiendo del rol del usuario (profesor o estudiante), a continuación, se mostrará como usarla dependiendo del rol

4.1 Utilización para Profesores

(a) (si ya se configuraron los objetivos y sus pesos saltar al punto **(d)**) En la siguiente figura se puede observar la pantalla inicial de configuración (está solo se mostrará una vez, cuando el curso no esté configurado), se deberán añadir los objetivos del curso uno a uno, una vez estén todos, se deberá dar al botón de **"confirmar"** para continuar.

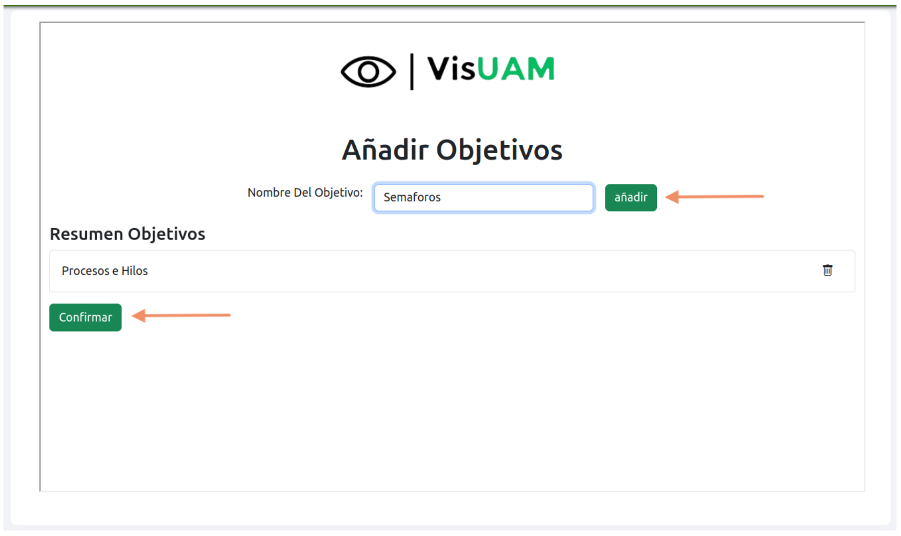


Figure 8: Pantalla de configuración de los objetivos

(b) En la pantalla que podemos ver en la siguiente figura, deberemos introducir nuestras credenciales de Moodle para recolectar los datos del curso (**importante:** cabe acotar que solo se guardara el correo en la base de datos, no la contraseña).

- **username**: correo de la UAM (solo de profesor con identificador de **uam.es**)
- **password**: tu contraseña de acceso a Moodle.

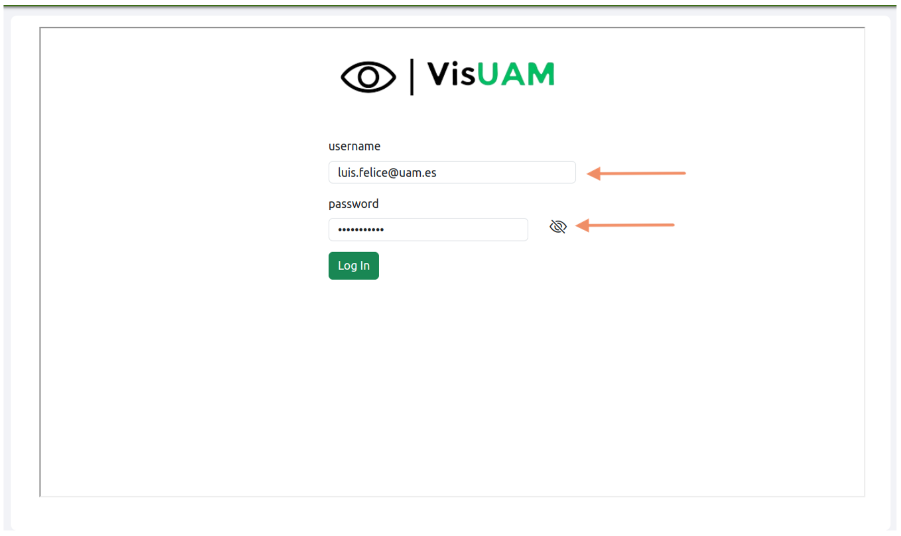
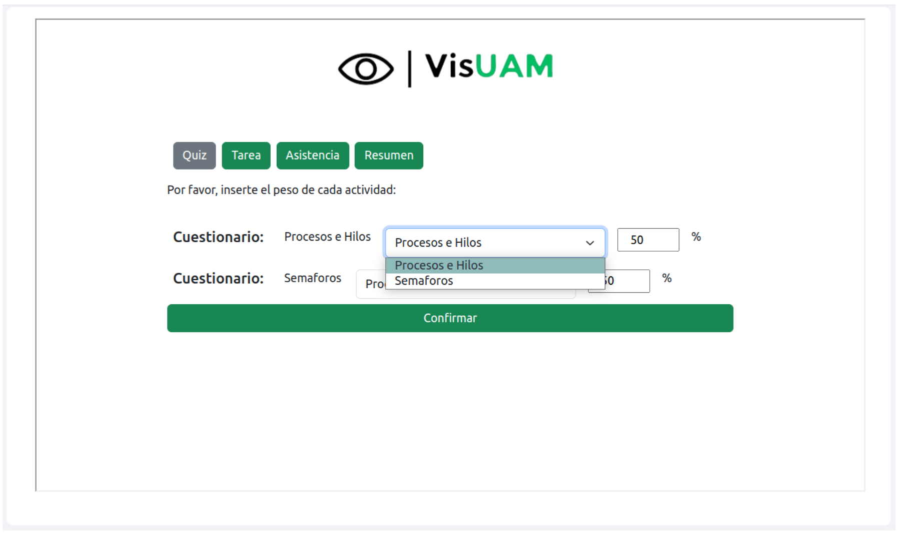


Figure 9: Pantalla para iniciar la recopilación de datos

Una vez que se haga clic al botón **”Log in”** se empezara la recopilación de datos y se solicitara al profesor que introduzca el token de verificación en la aplicación de **Microsoft Authenticator** para poder recopilar los datos del curso.

(c) En el siguiente paso, se solicitará al profesor que asigne objetivos a las actividades del curso, indicando también el peso de cada una dentro del objetivo asignado.

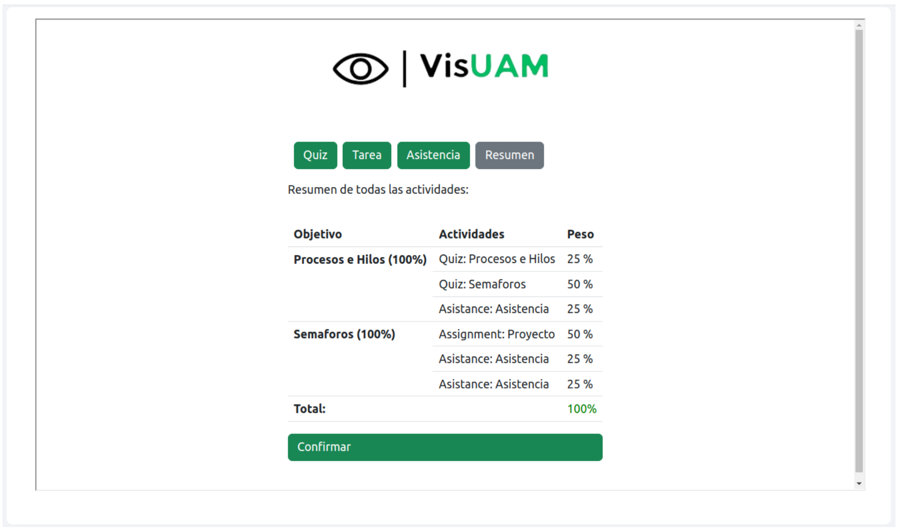


Por favor, inserte el peso de cada actividad:

Cuestionario:	Procesos e Hilos	Procesos e Hilos	50	%
Cuestionario:	Semaforos	Pro	50	%

Figure 10: Configuración de actividades

Una vez configuradas las actividades, antes de crear el curso se mostrará un resumen para confirmar las opciones seleccionadas.

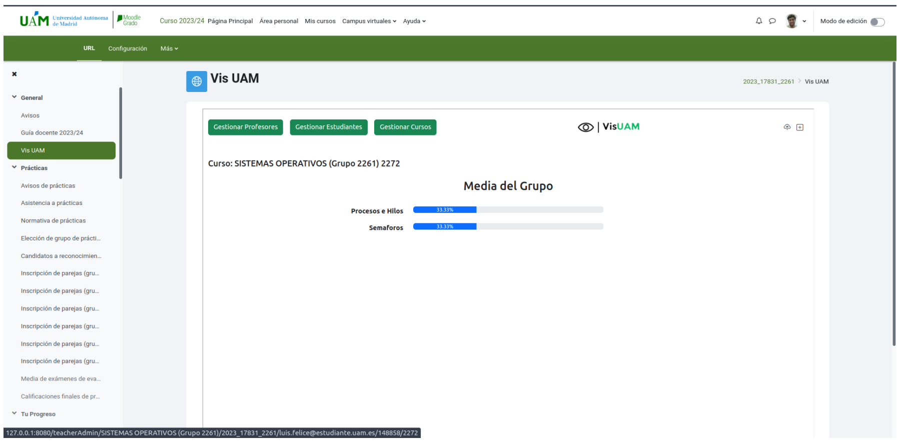


Resumen de todas las actividades:

Objetivo	Actividades	Peso
Procesos e Hilos (100%)	Quiz: Procesos e Hilos	25 %
	Quiz: Semaforos	50 %
	Asistance: Asistencia	25 %
Semaforos (100%)	Assignment: Proyecto	50 %
	Asistance: Asistencia	25 %
	Asistance: Asistencia	25 %
Total:		100%

Figure 11: Resumen de la configuración del curso

(d) Una vez configurado el curso se le desplegará la página de administración de la herramienta (solo disponible para profesores)



Curso: SISTEMAS OPERATIVOS (Grupo 2261) 2272

Media del Grupo

Procesos e Hilos	33.33%
Semaforos	33.33%

127.0.0.1:8080/teacherAdmin/SISTEMAS OPERATIVOS (Grupo 2261)/2023_17831_2261/luis.felice@estudiante.uam.es/148858/2272

Figure 12: Resumen de la configuración del curso

B
Survey Questions

1.- How would you rate your overall experience using the app?

2.- What aspects of the application negatively affected the experience (if any)?

3.- Do you think the app developed is easy to use?

4.- Which functionality(s) of the app have you found difficult to use?

5.- Did the application meet your functional expectations?

6.- What functionality(s) would you like to see in the application?

7.- How would you evaluate the visual design of the app?

8.- What improvements would you make to the application's graphic section?

9.- Was the visual representation of your progress in the course clear and understandable?

10.- Do you think the app would help you better manage your time and effort during the course?

11.- What feature would you add to the app to improve your time and effort management?

12.- Were there any features that you think should be added or improved?