UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

MÁSTER UNIVERSITARIO EN
INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

Design and Development of an Emotion-aware Learning Analytics system based on Machine Learning Techniques and Semantic Task Automation

ENRIQUE SÁNCHEZ TOLBAÑOS

2019

TRABAJO DE FIN DE MÁSTER

Título:	Diseño y desarrollo de un sistema de Análisis de Aprendizaje consciente de las emociones basado en Técnicas de Aprendizaje de Máquinas y Automatización de Tareas Semánticas
Título (inglés):	Design and Development of an Emotion-aware Learning Analytics system based on Machine Learning Techniques and Semantic Task Automation
Autor:	Enrique Sánchez Tolbaños
Tutor:	Carlos Ángel Iglesias Fernández
Departamento:	Departamento de Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:	—
Vocal:	—
Secretario:	—
Suplente:	—

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN
Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes

TRABAJO DE FIN DE MÁSTER

Design and Development of an Emotion-aware Learning Analytics system based on Machine Learning Techniques and Semantic Task Automation

JUNIO 2019

Resumen

El número de plataformas de aprendizaje ha crecido en los últimos años debido a los avances en la computacion en la nube, la accesibilidad tecnológica, y la tendencia de las personas a mejorar constantemente sus conocimientos y habilidades. Por esta razón, se han realizado innumerables investigaciones para mejorar su uso. Este trabajo está centrado en dos líneas. En primer lugar, los sistemas conscientes a las emociones en el ámbito educativo se basan en el uso de diferentes herramientas para reconocer las emociones de los estudiantes y adaptar las lecciones a su estado de ánimo. En segundo lugar, el objetivo principal del Análisis del Aprendizaje es aprovechar los conjuntos de datos académicos para inferir, crear, y predecir nueva información que ayude a mejorar el proceso de aprendizaje.

Una vez definidas las bases de este proyecto, su principal desarrollo ha consistido en la creación de un sistema capaz de detectar el estado de ánimo de los alumnos en el curso, y más concretamente, durante la realización de las diferentes actividades académicas. Este sistema se ha integrado en una de las plataformas de aprendizaje más utilizadas: Moodle.

Con el propósito de visualizar los datos recogidos por este sistema, ha sido necesario implementar un conjunto de visualizaciones en dos dashboards, diseñados para profesores y alumnos respectivamente. De la misma forma, estos datos han sido analizados con técnicas de Aprendizaje Automático para inferir relaciones, casos atípicos, o tendencias.

Por último, para aprovechar las capacidades del detector de emociones implementado, se ha desarrollado una nueva versión de Ewetasker, una plataforma semántica de automatización de tareas. A través de ella, los alumnos son capaces de adaptar el entorno a sus emociones, mejorando su comodidad mientras realizan las diferentes tareas del curso.

En resumen, el objetivo de este proyecto ha sido mejorar el estado de ánimo de los alumnos a través de los desarrollos realizados y, en consecuencia, su rendimiento académico.

Palabras clave: Plataformas de aprendizaje, Reconocimiento de emociones, Dashboard, Machine Learning, Tecnologías Semánticas, Internet de las cosas, Automatización basada en eventos, SPARQL, Elastic.

Abstract

The number of e-learning platforms has grown in recent years due to advances in cloud computing, ease of access to technology, and the trend of people to constantly improve their knowledge and skills. For this reason, a great deal of research has been carried out to improve the use of these platforms. This work focuses on two research streams. First, Emotion-aware systems are based on the use of different tools to capture students' emotions in order to adapt the lessons to their mood. Secondly, the main goal of Learning Analytics is to harness educational data sets to infer, create, and predict new information that helps to improve learning process.

Once the basis of this project have been defined, its main development has consisted of the implementation of a system capable of detecting the mood of students in the course and during the performance of different activities. This system has been integrated into one of the most used e-learning platforms: Moodle.

With the purpose of displaying the data collected by this system, it has been necessary to implement a set of visualizations in two dashboards, designed for teachers and students respectively. In the same manner, this data has been analyzed with Machine Learning techniques to infer relations, outliers, or trends.

Finally, to take advantage of the capabilities of the implemented emotion detector, a new version of Ewetasker, a semantic task automation platform, has been developed. Through it, students are able to adapt the environment to their emotions, improving their comfort when performing tasks.

To summarize, the aim of this project has been to improve students mood through the different developments carried out, and consequently, their academic performance.

Keywords: E-learning, Emotion recognition, Dashboard, Machine Learning, Semantic technologies, Internet of Things, Event-based automation, SPARQL, Elastic.

Agradecimientos

Me gustaría dar las gracias a Carlos Ángel por el tiempo que me ha dedicado y por darme la oportunidad de trabajar en el Grupo de Sistemas Inteligentes.

A Óscar, Fernando, Álvaro y especialmente a Sergio por estar siempre dispuestos a ayudarme.

A Eduardo, Diego, Francisco, Pablo, Tasio y Daniel por colaborar en la realización de este proyecto.

A Alberto, Manuel y Rodrigo por acompañarme en estos años de máster.

A Iván por compartir enseñanzas, valores y sentimientos en rojo y blanco.

Por último, gracias a mis padres por apoyarme siempre.

Contents

List of Figures

CHAPTER 1

Introduction

This chapter introduces the context of the project, including a general review of all the different parts that are discussed in the project. In the same way, the main lines of research that are currently being carried out on E-learning are presented, as well as the motivation for the development of this project. It also breaks down a series of objectives to be carried out during the realization of the project. Moreover, it introduces the structure of the document with an overview of each chapter.

1.1 Context

Nowadays, e-learning [39] is a key piece in learning context due to the advantages and possibilities of its use in different academic environments, whether in distance learning, self-learning, or face-to-face learning. For this reason, it inspires the research and development efforts of numerous companies and universities, which seek to improve different methods of learning as well as academic results obtained. For this purpose of improvement, different ways appear, as the case of Emotion-aware [21] and Learning Analytics (LA) [63] systems.

Firstly, Emotion-aware systems are based on cognitive computing [32] and are capable of detecting human emotions through techniques such as image recognition or voice tone analysis. Using these systems, this type of tasks can be performed on a larger scale, with better results, and reducing human-machine interactions.

Secondly, LA systems are fed with data from the learning environment to conduct a comprehensive analysis to understand the relevance and relationship between different factors that appear in the learning process, with the ultimate aim of improving this process. This analysis is usually carried out using Machine Learning (ML) [74] techniques and algorithms, which are capable of making predictions based on datasets.

As interest in e-learning grows, so does the number of smart devices in our homes and offices, known as Internet of Things (IoT) [1]. All these devices look for making life easier to people. However, due to the great variety of smart devices, it is necessary to use a system that is capable of integrating them to coordinate and enhance their capabilities. To facilitate the integration of these devices together with web services, task automation platforms appear.

In this project, we try to offer an improvement for e-learning platforms, through an Emotion-aware Learning Analytics system. Moreover, to support and complement the development of this goal, we use semantic technologies for task automation.

1.2 Motivation

The motivation for this project stems from the growing interest in the application of techniques such as cognitive computing or ML for the improvement of many areas of society. Within these fields is the academic field, which alone arouses significant interest, due to the continuing need for learning in today's society, and the increase in the number of methodologies or ways of learning as provided by e-learning platforms.

Through the use of these techniques it is possible to apply theories and methodologies [33], based on psychology, to improve the learning process, taking into account factors such as the emotional in non-traditional academic environments (e.g. distance learning).

Finally, it is intended to take advantage for this same purpose of the growing emergence of intelligent devices around us. By integrating these devices into a learning environment, it is possible to improve students mood and performance, as it has already been observed in other related works [44].

1.3 Project goals

The main goals of this project are:

- Design and development of a system capable of detecting the mood of students in the course and during the performance of different activities.
- Design and development of a visualization system to display the data collected by the emotion-aware system along with academic data.
- Design and development of an analytical learning system to predict the state of mind and academic results, depending on the different activities carried out in the course.
- Integration of the system in a smart educational environment through a semantic task automation platform.

1.4 Structure of this document

In this section we provide a brief overview of the chapters included in this document. The structure is the following:

Chapter 1 explains the context in which this project is developed, mainly the e-learning environment. In addition, it describes the main goals to achieve in this project, as well as the motivation and the structure of this document that is being read.

Chapter 2 provides a description of the main technologies on which this project relies, among them, we can find e-ĺearning platform, emotion-recognition tools, big data technologies for storing, managing and visualizing, ML technologies, and semantic task automation.

Chapter 3 presents the current state of Emotion Aware E-learning platforms, the most used emotion recognition tools, the main related researches carried out in recent years, as well as our own vision of the topic.

Chapter 4 presents the current state of Learning Analytics, the steps that involved the analysis process, the main related researches carried out in recent years, as well as our own proposal.

Chapter 5 describes the architecture of the project, including the design phase and implementation details. A general picture of the prototype architecture, as well as the project in which this Master Thesis is framed. Then, all the modules developed in the prototype are widely detailed.

Chapter 6 describes a selected use case inside a specific scenario. In this chapter it is explained how the whole system works when an online Python course is carried out in a Moodle platform.

Chapter 7 discusses the conclusions drawn from this project, as well as the problems faced in its development. Finally, we focus in the possible next step to be done for a future work.

CHAPTER 2

Enabling Technologies

This chapter offers a brief review of the main technologies that have made possible this project, as well as some of the related published works. Firstly, the learning platform used is reviewed as well as the emotion-recognition tools. More technical sections detail the data storage and visualization solutions. Finally, the task automation platform and the ontology implemented for its operation are presented.

2.1 E-learning platforms

E-learning platforms are one of the most widely used tools for learning today. These platforms can be used both for distance learning courses and as a complement to face-to-face courses. Their importance is such, that several studies [48] have already been carried out in which their characteristics are compared, and an attempt has been made to discern which platform is the most suitable according to the learning needs.

However, it can be considered that all these platforms basically work in the same way. The teacher is in charge of creating the modules for each course he teaches. In these modules, the educational resources necessary for learning are created and stored. In the same way, follow-up activities are elaborated to evaluate the student's academic progress. The student, therefore, accesses the modules of the platform and initiates or complements his learning process with the available resources. Finally, these platforms must be governed by an administrator who is responsible for installation, deployment, customisation and security tasks among others.

The potential of these platforms is great, and it is possible to list the most relevant capabilities they offer:

- Share learning resources in different formats such as text, presentations, images, audio and video.
- Carry out self-evaluation activities such as tests, problems, questionnaires or games.
- Provide communication tools, between teachers and students, such as forums, chats, private messages, and video conversations.
- Evaluate the student's learning with tasks, exercises and exams to be carried out online.
- Manage groups of students, learning plans, subjects or marks.

Within the scope of this project, we will use the following e-learning platform as a base for our work.

2.1.1 Moodle

Moodle [47] (Modular object-oriented dynamic learning environment) is a Learning Management System known to be the most relevant e-learning platform due to the fact that it is used by more than 90 million people around the world. It has been designed *"to provide educators, administrators and learners with a single robust, secure and integrated system to create personalised learning environments"* [1].

This success lies in the fact that it is an open source system distributed under the GNU General Public License [2]. It has been developed in PHP [3] language and it is continually being updated by its community and a team of dedicated full-time developers.

Attending to their advantages and features we can find the following ones:

- It is easy to use for teachers and students due to its simple interface. In the same way, a complete documentation is available to adapt each deployment to concrete cases of use. It also breaks the language barriers, its interface and documentation are available in more than 120 languages.
- It provides a comprehensive set of learning tools, such as quizzes, forums or resource sharing, easily adaptable to the goals of the course. Similarly, the modular architecture it implements allows developers to create plugins and integrate external applications to achieve specific functionalities.
- It can be scaled from a few students to millions, being a platform used in different contexts apart from education such as business or government.
- It is a robust, secure and private platform that implements mechanisms, which are constantly updated, against unauthorised access, data loss and misuse.
- It is a web-based platform and so can be accessed from anywhere in the world. It also provides a full cross-browser compatibility for plenty of devices including a mobile phone.

[1] https://docs.moodle.org/36/en/About_Moodle
[2] https://www.gnu.org/licenses/gpl-3.0.ht
[3] http://php.net/

2.2 Emotion-Recognition Tools

In the last decade there has been an increase in the number of emotion recognition tools. Similarly, there has been a notable improvement in the quality of the recognition they perform. All this has provoked a growing interest on the part of researchers, who no longer focus exclusively on improving these tools but seek to apply their advantages in various scenarios such as online learning.

These tools aim to use different sources to detect and identify the emotional state of people. In our research we will use both tools based on physiological reactions and tools that implement facial recognition techniques. In the same way, we will include within these tools the attention detector.

2.2.1 Clmtrackr

Clmtrackr [75] is a JavaScript library for fitting facial models to faces in images and video, and can be used for getting precise positions of facial features in an image, or precisely tracking faces in video. The library is empowered by a fitting algorithm based on a paper [61] by Jason Saragih & Simon Lucey. This paper proposed *"a principled optimisation strategy, where a non-parametric representation of the landmark distributions is maximised within a hierarchy of smoothed estimates"* to obtain a improved facial fitting model.

Due to this, Clmtrackr is capable of tracking a face and outputs the coordinate positions of the face model as an array, following the numbering of the model shown in 2.1. Moreover, this library provides some generic face models that were trained on the MUCT face database [4] and some additional self-annotated images.

Clmtrackr was created by Audun Mathias Øygard and it is an open source project distributed under the MIT License [5]. In the last few years, this library has been improved adding some new features, among other, we found the emotion detection.

The operation of the emotion detection module is simple: the library recognises user's face through a web cam and obtains its coordinates in the same way as explained in 2.1. Attending to the position of each coordinate, it is capable of detecting user's emotion. Despite of the fact that Clmtrackr reference documentation specifies it is possible to differentiate

[4] http://www.milbo.org/muct/
[5] https://opensource.org/licenses/MIT

among six emotions (anger, sadness, surprise, happiness, disgust and fear), a previous research, carried out by Carlos Marzal Romón [36], suggests not to use the last two of them because of their malfunctioning.

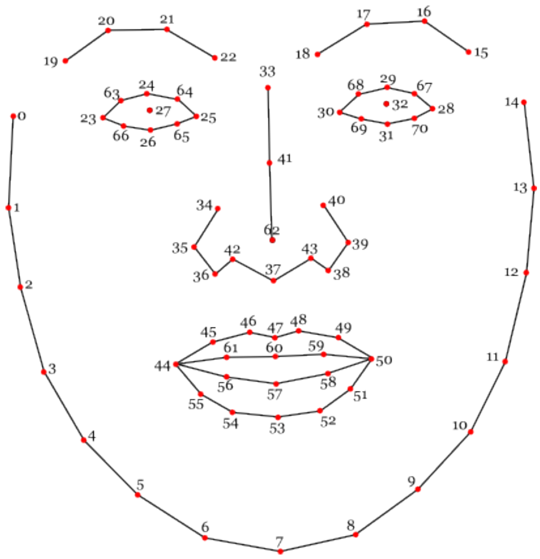

Figure 2.1: Clmtrackr Facial Model Annotation [75]

In this project, we will use Clmtrackr to detect students' emotions in Moodle platform.

2.2.2 Webgazer

WebGazer [49] is an eye tracking library that uses common webcams to infer the eye-gaze locations of web visitors on a page in real time. The eye tracking model it contains self-calibrates by watching web visitors interact with the web page, and trains a mapping between the features of the eye and positions on the screen. This way, it is capable of detecting user attention.

Internally, it is composed of two modules. The tracker module controls how eyes are detected, and the regression module determines how the regression model is learned, and how predictions are made based on the eye patches extracted from the tracker module. In addition, it provides the capability of working with different track modules to get features as gaze, eye pupils or faces. In the same way, it is possible to swap between different regression modules. To develop this project we have chosen Clmtrackr (presented before) and a ridge regression module (capable of mapping pixels from the detected eyes to locations on the screen).

This open-source tool is developed in JavaScript and can be used in practically any website because it runs entirely in the client browser. We will integrate this tool in Moodle's environment in order to track students visual interactions.

2.2.3 Empatica E4 wristband

The Empatica E4 wristband [19] is a wearable research device that offers real-time physiological data acquisition and software for in-depth analysis and visualisation. This device is equipped with the following suite of sensors:

- PPG Sensor. Measures Blood Volume Pulse (BVP), from which heart rate variability can be derived.
- 3-axis Accelerometer. Captures motion-based activity.
- EDA Sensor (GSR Sensor). Measures the constantly fluctuating changes in certain electrical properties of the skin.
- Infrared Thermopile. Reads peripheral skin temperature. .

Empatica provides integrations with the Android and iOS platforms using the SDK to allow real-time data streaming from E4 devices. In the scope of this project, we will use the E4 device to obtain the stress levels of the students' through its EDA sensor. Recent researches [62] [3] have linked the distributions of EDA peak height and instantaneous peak rate with high levels of stress.

Figure 2.2: Empatica E4 wristband

2.3 Data Storage Technologies

The election of a data storage technology is a critical point when developing an information technology project. For choosing correctly, it is vital to know the features of the data we want to storage. Of all these features, the two most relevant (data size and complexity) related with the existing database types as shown in 2.3.

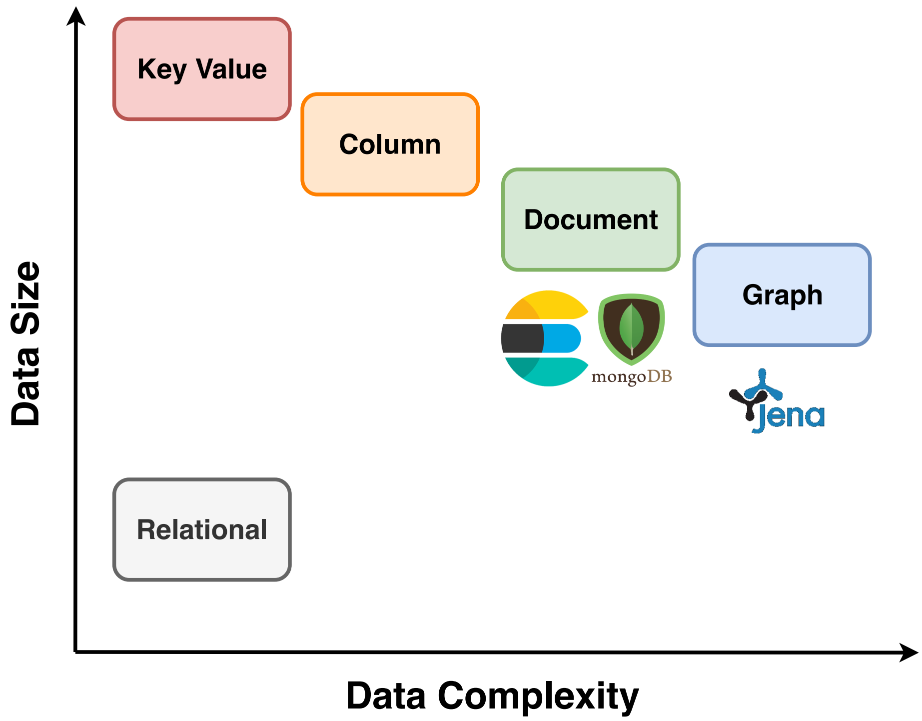

Figure 2.3: Database Classification Chart

The diversity of the applications implemented and developed in this project makes necessary to use three different types of databases: MongoDB, Elasticsearch and Fuseki.

2.3.1 MongoDB

MongoDB [13] is a NoSQL database, in other words, a non-relational database. The way it works is simple. The database contains collections which are made up of documents. These documents are, in turn, composed of fields, where information is stored.

As we explained, it is oriented to documents and more specifically to JSON [6] documents. This is an advantage because JSON format is widely used in web applications. Moreover,

[6] https://www.json.org/

the treatment of these documents is done in a flexible way, being able to modify each data entry in the document dynamically. Another advantage is that it facilitates data analysis using ad hoc queries, indexing, and real time aggregations. In addition, it is a distributed database, providing horizontal scaling and high availability.

This database is one of the most used storage technologies, with more than 9 million of downloads around the world. It is a free and open-source tool. Newer versions are published under the Server Side Public License (SSPL) v1 [7] and it provides integrations for more than ten programming languages as Python or Java.

In this project, we will use MongoDB in Ewetasker platform to store users data and in Emotion Moodle platform to store courses data.

2.3.2 Elasticsearch

Elasticsearch [17] is a distributed, RESTful search and analytics engine highly scalable. It is designed to be a central data store and it is generally used as the underlying search engine that powers applications that have complex search features and requirements.

Its widespread use is due to the fact that provides a HTTP web interface to interact through queries based on JSON. In the same way, the responses to these queries are schema-free JSON documents.

The way Elasticsearch works is complex, but it provides great performance. To do so, this tool is formed by indices which are divided into shards. The shards are stored in distributed nodes, and each shard can be replicated more than once. These nodes act as a coordinator to delegate operations involving different shards.

This search server is based on Lucene [8] library (Java) and it has been developed within an Elastic toolkit. For its use, it provides clients in the main programming languages such as Java, SQL, .NET, PHP or Python. According to different rankings of database popularity [28][69][14], Elasticsearch was in the top ten most used databases in 2018.

In this project, we will use Elasticsearch in Ewetasker platform to store usage data and in Emotion Moodle platform to store students and emotion data.

[7] https://www.mongodb.com/licensing/server-side-public-license
[8] https://lucene.apache.org/core/

2.3.3 Fuseki

Fuseki [31] is a SPARQL server based on Apache Jena framework. It is known to be a solution as a storage layer for ontologies. To do so, it provides the SPARQL 1.1 protocols for query and update as well as the SPARQL Graph Store protocol. Among its advantages stand out its security, by using Apache Shiro's framework [9], and its interface which allows server monitoring and administration.

This tool has been developed by Jena, so it incorporates Jena's text query and spatial query. In addition, its uses are diverse. It can be used as an operating system service, as a Java web application, and as a standalone server. In some cases, Fuseki is used to provide the protocol engine for other RDF query and storage systems.

In this project, we will use Fuseki in Ewetasker platform to store channels information and automation rules in N3 through Ewe ontology.

2.4 Data Managing Libraries

Data management tools are vital for extracting, visualizing, and creating information from collected data. In this project we will use two of Python's most relevant libraries.

2.4.1 Pandas

Pandas [37] is an open source library for Python that provides easy-to-use data structures and data analysis tools with a high-performance. It is mainly based on the use of objects called DataFrames. These objects implement a two-dimensional data structure with integrated indexing, which is easily adapted to commonly used data structures such as csv, text files, Microsoft Excel or SQL databases. For this reason, Python with pandas is used in a wide variety of academic and commercial domains.

Among the most relevant features of this library stand out:

- Intelligent data alignment and integrated handling of missing data.
- Flexible reshaping and pivoting of data sets.

[9] https://shiro.apache.org/

- Aggregating or transforming data with a powerful group by engine allowing split-apply-combine operations on data sets.
- High performance merging and joining of data sets.
- Date range generation and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.

2.4.2 Numpy

NumPy [70] is the wide used package for scientific computing with Python. This open source tool provides mainly a powerful N-dimensional array object and useful linear algebra, Fourier transform, and random number capabilities. In the context of data managing and Machine Learning, this tool is used to make transformations to numerical format for entries that have textual characteristics, in such a way that they are understandable for ML algorithms.

2.5 Machine Learning

As with data processing tools, there are numerous options available for the application of machine learning techniques. In this project, they will be used in conjunction with data management tools to carry out mainly classification tasks. For this reason, the Scikit-learn library available for Python has been chosen.

2.5.1 Scikit-learn

Scikit-learn [51] is defined as a set of simple and efficient tools for data mining and data analysis. It is based on tools mentioned above such as Numpy and others also oriented to data processing such as Scipy and Matplotlib. It is also an open source solution. Among the capabilities it offers, the following stand out:

- Classification. Identifying to which category an object belongs to. Algorithms: SVM, nearest neighbors, random forest, etc.
- Regression. Predicting a continuous-valued attribute associated with an object. Algorithms: SVR, ridge regression, Lasso, etc.

- Clustering. Automatic grouping of similar objects into sets. Algorithms: k-Means, spectral clustering, mean-shift, etc.
- Dimensionality reduction. Reducing the number of random variables to consider. Algorithms: PCA, feature selection, non-negative matrix factorization.
- Model selection. Comparing, validating and choosing parameters and models. Modules: grid search, cross validation, metrics.
- Preprocessing. Feature extraction and normalization. Modules: preprocessing, feature extraction.

2.6 Data Visualisation Technologies

Today, data visualisation technologies are as important as data collection technologies. They give us the ability to visualise and discover relationships, patterns, and trends in data, and as a result, they allow us to better understand the meaning of data.

2.6.1 Sefarad

Sefarad is a web tool for visualisation of semantic data developed by Intelligent Systems Group (GSI) at Universidad Politécnica de Madrid (UPM) . It is capable of making linked data queries in two different kind of databases: ElasticSearch and Fuseki. In this manner, it facilitates the creation and use of dashboards by providing generic software that can be easily adapted to the users' needs.

The Sefarad's visualisations available are very different depending on the features of the data to be shown, and the purpose user wants to have of them. In addition, these visualisations are interactive, so, it is possible to navigate through the features, properties and relations of the data for a deeper and more complete view of them.

Sefarad 3.0
The future of the web today

Each of these visualisations is based on Polymer Web Components. These web components are a set of web platform APIs that allows users to create new custom, reusable, encapsulated HTML tags to use in web applications. To make use of them, only is needed a modern browser capable of implementing Javascript libraries or HTML frameworks.

In this project, we will use Sefarad to show the usage data of the task automation platform developed. To know more about this tool a complete specification is available.

2.6.2 Kibana

Kibana [29] is an open-source data visualization and exploration plugin for Elasticsearch, used for log and time-series analytics, application monitoring, and operational intelligence use cases. It is widely used because of its visualization capabilities of the content indexed on an Elasticsearch cluster. It is part of the Elastic Stack toolkit (Elasticsearch, Kibana, Beats, and Logstash) [10], designed by Elastic company for the ingestion, storage, and display of large amounts of data.

This tool offers a wide variety of visualizations such as histograms, line graphs, pie charts, heat maps, and geographic maps. Among its main features, the following stand out: interactive charts, pre-built aggregations and filters for Elasticsearch data, and easily accessible dashboards.

In the context of this project we will use Kibana to implement dashboards for the data visualization of the e-learning platform.

[10] https://www.elastic.co/es/

2.7 Semantic Task Automation

Task automation applications are the norm due to the recent appearance of numerous so-called smart devices. These applications base their operation on the definition of rules. These rules are defined by two main facts: a condition and a consequence. If certain conditions set out in the rule are met, a set of actions also specified in the rule must be performed. The parameters used in the rules can be defined in a wide variety of languages or formats [54], and also include definitions of all kinds of domains, from subjective facts such as emotions to objective facts such as temperature.

Due to its enormous potential, in recent years numerous applications have appeared that allow to automate tasks by defining rules. Among all the available applications, IFTTT (If This, Then That) [11] stands out, which bases its operation on the rules explained above, and offers users the possibility of using both a web application and a mobile application. However, this project will use a semantic rule-based task automation platform that has been developed and improved for better integration in different contexts.

2.7.1 Task automation

Task automation [11] is defined as the use of intelligent systems and cognitive technologies to implement tasks or processes and control their performance, minimising human intervention. The following steps are usually followed in a task automation process:

- A definition of what each task does and how they relate to each other.
- A schedule of when the task is done and its duration.
- A specification of what resources, equipment, and tools are needed to perform the task.
- A tracking system to control and monitor the development of the task.

[11] https://ifttt.com/

2.7.2 Notation 3

As mentioned above, our task automation platform is based on semantic rules. For this purpose, the rules are defined using Notation3 (N3) [6], a Semantic Web logic. This logic is based on the use of triples called Turtle [4], which is an RDF [12] data serialization format. Through these triples it is possible to define both the rules and the agents that intervene in them, and therefore, makes possible the interaction and understanding between different domains such as Internet services and physical devices.

The World Wide Web Consortium (W3C) defined as targets of N3 [6]: to optimize expression of data and logic in the same language; to allow rules to be integrated smoothly with RDF; to allow quoting so that statements about statements can be made, and to be as readable, natural, and symmetrical as possible. In the same way, it also defines the main characteristics of the language: URI abbreviation using prefixes which are bound to a namespace (using @prefix) following XML style, repetition of objects and predicates using a comma "," or a semicolon ";" respectively; bnode syntax with a certain properties; formulae allowing N3 graphs to be quoted within N3 graphs using { and }; variables and quantification to allow rules, etc to be expressed; and a simple and consistent grammar.

2.7.3 Eye

Once it is possible to define rules through the N3 language, it is necessary to have a tool that is capable of evaluating, according to what is specified in the rules, whether the conditions for launching the actions have been met. This tool is EYE (Euler YAP Engine) [15], a reasoning engine that supports the Semantic Web layers. Among its characteristics, it stands out that it carries out semi-directional reasoning and supports Euler's paths.

However, the choice of this reasoner is given by its high performance and optimization when evaluating rules in N3 language. This tool is developed in Prolog, so it translates the rules it receives in N3 to Coherent Logic Prolog language to be evaluated by the reasoning engine that implements called YAP (Yet Another Prolog).

[12]https://www.w3.org/RDF/

2.7.4 EWE Ontology

Evented WEb Ontology (EWE) [10] is a standardized data schema (also referred as "ontology" or "vocabulary") designed to describe elements within Task Automation Services enabling rule interoperability. EWE ontology was developed to achieve the following objectives:

- Enable to publish raw data from Task Automation Services (Rules and Channels) online and in compliance with current and future Internet trends.
- Enable Rule interoperability.
- Provide a base vocabulary for building domain specific vocabularies.

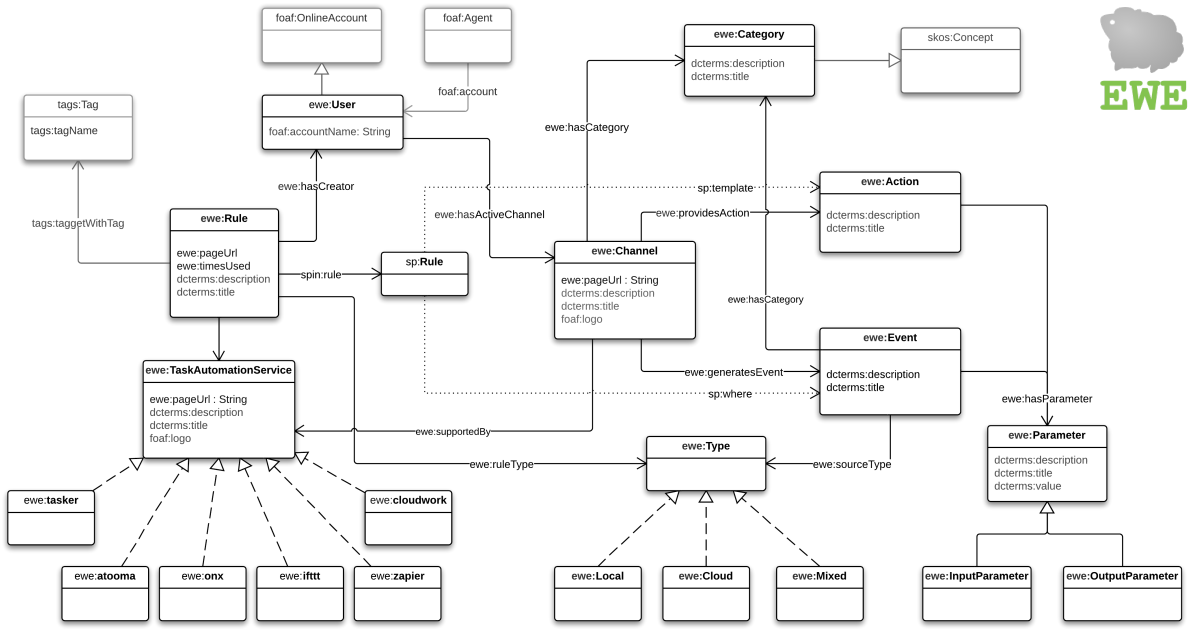

Figure 2.4: EWE Class Diagram [10]

As shown in 2.4, EWE ontology is divided in four main classes: Channel, Event, Action and Rule.

- ***Channel:*** It defines subjects which can generate Events, provide Actions or both. Sensors and actuators are also described as channels, therefore they produce events or provide actions.
- ***Event:*** It defines the realization of a fact. These facts are defined on a channel, but they are used on the rules. When a rule is created, one or more events should be

defined to trigger the action. Events also let users describe with parameters under which conditions should they be triggered.

- ***Action:*** It defines an operation or process that can be performed by a channel. Like events, they are defined on the channel creation. Besides, when creating a rule, we can define parameters necessaries to make the action.
- ***Rule:*** It is defined as an "Event-Condition-Action" (ECA) rule. This rule is triggered by an event, and then, it executes an Action. Its purpose is to define the connections between the Events and Actions present on two or more channels. In order to perform this, when user creates it, includes the configuration parameters set for both of them.

CHAPTER 3

Emotion Aware E-learning Systems

The use of e-learning platforms has increased in the last decade. Despite of their success, difficulties to follow the lessons have appeared as the same way as they appear in traditional learning methods. In order to improve the academic achievement of students, user's emotion is being considered essential. There are many implementations that try to detect user's emotion during learning to evaluate and adapt the lessons. In this project, a new implementation of an emotion aware e-learning platform is being deployed to adapt the smart environment to users' emotions and improve their learning experience. To do so, a review of the existing models in affective e-learning is made, and then, our own vision about the topic is introduced.

3.1 Introduction

The amount of e-learning platforms has grown in recent years produced by the progress in cloud computing, the user's ease of access to technology, and the trend of people to constantly improve their knowledge and skills. These platforms as edX [1] or Coursera [2] allow students from all over the world to enrol in thousand courses, being the number of massive open online courses registrants more than ten million [53]. However, the total number of e-learning users must be much higher since universities, high-schools and private schools use platforms as Moodle to complement their face-to-face lessons.

The technological capabilities of these platforms have reached a point in which is very difficult to improve them. Features as video conversations in streaming or online exercises allow students to follow lessons practically in the same way as they used to in person. Despite of this, there are still some paths to explore in order to get the maximum advantage out of the lessons.

One of these paths is related with the students' emotions. In recent years emotions in a learning process have become an important field of studying by psychologists. It is known that emotions are relevant to improve the academic results because they are close related with the attention. A student with positive emotions will pay more attention to the lesson, which will result in a successful learning and a huge use of memory[72]. Other studies talk about the influence of emotions in the way people act [66] and some others describe the connection between emotions and encouragement when students try to learn [52].

Off all these theories, it is highlighted the model of affective in education by Kort et al. [33]. This model, shown in 3.1, identifies four phases of emotion and suggests six emotion axes (anxiety-confidence, ennui-fascination, frustration-euphoria, dispirited-enthusiasm, terror-excitement, and humiliated-proud) that can be experienced by students while learning.

The model is explained by the authors as follows: *"A typical learning experience involves a range of emotions, cycling students around this four quadrant cognitive-emotive space as they learn. (...) It is important to recognise that a range of emotions occurs naturally in a real learning process, and it is not simply the case that the positive emotions are the good ones. (...) We do not foresee trying to keep the student in Quadrant I, but rather to help him see that the cyclic nature is natural in learning science, mathematics,*

[1] https://www.edx.org/es
[2] https://www.coursera.org/

engineering or technology, and that when he lands in the negative half is an inevitable part of the learning cycle. Our aim is to help students to keep orbiting the loop, teaching them to propel themselves, especially after a setback".

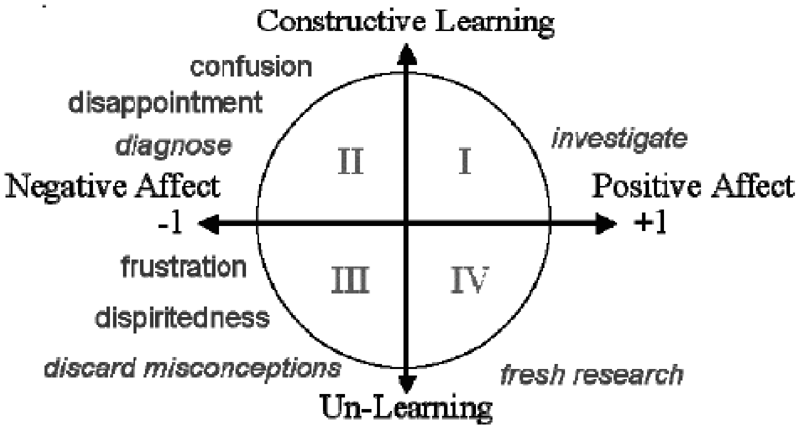

Figure 3.1: Kort et al. Four Quadrant model relating phases of learning to emotions

Following this model, recognising students emotions seems to be essential to stimulate them in a learning process. To do so, first it is important to know how emotion awareness can be carried out. M. Feidakis [21] defines emotion awareness as *"the implicit or explicit collection of emotion data and the recognition of emotion patterns"*. This definition is carefully explained by the author distinguishing three types of emotion captures:

- **Implicit detection of emotion data.** Physiological signals or the motor-behavioural activity of emotions, that is, heartbeat, facial expressions, voice intonations, etc.
- **Explicit input of emotion data.** First-person subjective report of feelings.
- **Recognition of emotion patterns.** Identification of the human's emotions based on observation.

Returning to the education field, two types of environments exist. In a traditional classroom, teachers can find out their students' emotional reactions while learning by glancing at their face expressions while they are attending, doing example exercises on the blackboard or asking them at the end of the lesson. In this way, teachers can gauge the effectiveness of their teaching strategies, determining how students feel according to the topics, activities and evaluations of the lesson.

On the other hand, in an e-learning platform, students experience emotional reactions just as they do in an ordinary classroom. They have to face the same situations like understanding the lesson, doing the proposed activities or being evaluated. Nevertheless, e-learning students have to do all of this without interacting personally with other classmates or the teacher, which can result in negative emotions as frustration or sloth.

Despite of the importance of recognising students emotions in e-learning platforms, there are no many real implementations of these platforms that include emotion recognition. The inclusion of this capability can multiple the benefits of distance learning and can be a value added to the platforms in charge of providing this service. For these reasons, emotion recognition in order to be used by the own students or by the responsible teacher has caught the eye of many researchers. The experimental implementations found about this topic and the tools they have used to capture emotions are going to be presented here under.

3.2 Available Tools for Emotion Recognition

As M. Feidakis explained in his study of Emotion-Aware Systems that we mentioned before, there are three types of tools to carry out an emotion recognition. Two of them are well known because they are used in face-to-face communication. When people talk the first communication tool is the language, which can be used by the speaker to express explicitly or implicitly his emotions. The second tool is the corporal expression that includes facial expressions, eye contact, corporal posture or nonverbal gestures with implicit meanings. The last tool of this set is related with the biological responses that the human body performs when people feel an emotion. This kind of responses cannot be perceived without the help of medical tools that allows to monitoring human organ activity.

The use of each of these tools entails a number of advantages and disadvantages that can affect since the person involved in the experiment to the results of itself. For this reason it is fundamental to know them in detail in order to understand the implications and consequences of their use in the developments of emotion-aware e-learning platforms that we are going to present later.

3.2.1 Self-Reporting Tools

The self-reporting tools have their origin in the psychology. They are widely used because of their simplicity and their low cost, apart of being a non intrusive tool. A self-report can be defined as any test, measure, or survey that relies on the individual's own report of their symptoms, behaviours, beliefs, or attitudes [3]. This kind of reports can be carried out by filling a physical or electronic questionnaire or making a personal interview.

Like we have mentioned before, the main advantage of this tool in a research environment is to be very inexpensive respect to any other report tool. This allows researchers to take a huge number of measures in their experiments which can be translated into better results for their investigations. Moreover, its simplicity permits to obtain the results quickly.

However, this kind of reports has some disadvantages too. Firstly, it is important that respondents answer with honesty. In these questionnaires, people try to give socially acceptable answers rather than truthful ones. Another problem is that people might not have introspective ability to answer correctly to the questions. Lastly, it may appear some problems related to the obtaining of the report like a wrong interpretation or formulation of the questions, a wrong sampling bias or the use of unadjusted rating scales.

If we focus on the use of these reports as a tool for measuring people emotions, we found two kinds:

- **Verbal self-reporting:** The questionnaires include open-ended questions and rating scales in order to know the respondent's emotions and their level of intensity. Other methodologies include the use of conversations or written diaries that allow to express the emotions with own-words.
- **Nonverbal self-reporting:** The questionnaires substitute the words to express emotions with elements which have a close meaning in the respondent culture. They suppose an advantage because of their simplicity but they cannot use specific emotions in contrast with verbal reports.

3.2.2 Emotion-Recognition Tools

These tools have been used for many years through the intervention of expert psychologists who identified the emotions of the patient based on their body posture, tone of voice or facial expression. However, in recent years, improvements in the field of artificial intelligence [32] have allowed the same analysis of emotions using emotion recognition devices and without the help of psychologists.

These techniques have the advantage of being non-invasive so they are easier to imple-

[3] https://www.verywellmind.com/definition-of-self-report-425267

ment and the user does not need to take any additional action to obtain the measurements. Moreover, they do not need specialised equipment, since most use common devices such as a web-cam or a microphone.

On the other hand, they cannot be considered as very reliable because the user can try to deceive the system. In addition, they are relatively recent tools that need time to be fine-tuned in order to obtain reliable results.

The following list includes the most relevant emotion-recognition tools used in recent publications:

- **Facial Detection.** It is based on the measurement of geometric features of user's face [23] [65].
- **Voice Tone Detection.** It is based on the extraction of voice quality features that are related to speech emotions [73].
- **Behaviour Detection.** It is based on the recognition of body expressions related to emotional states and in the measurement of frequency and speed of user's movements [23] [5].
- **Sentiment Analysis.** It is based on the lexical and entity analysis of what people write to classify it following an emotion model [59] [30].

3.2.3 Neurophysiology Tools

The use of neurophysiology sensors makes it possible to obtain, unlike previous tools, objective measurements of users' emotions. To this end, by using different type of sensors, data is obtained from the signals produced by the human body in response to certain events, emotions or stimuli.

The advantages of using these sensors are mainly two. The first one is that measures are truthful because body responses to emotions are involuntary, which means that cannot be controlled by people. The second one is that there is an extensive catalogue of neurophysiology sensors, each one specialised in the measurement of one concrete body signal that can be related to a specific emotion.

Otherwise, compared to self-reports, these are very specific tools with a very high cost. Apart from the acquisition of the equipment, technical knowledge is needed to use the equipment. Besides, most of these sensors are uncomfortable for users, making them difficult to use and being themselves a source of negative emotions.

Below is a list of the most commonly used neurophysiology sensors [4] for emotion measurements.

- **EEG.** Electroencephalography (EEG) is the recording of electrical activity along the scalp measured by an EEG Sensor or EEG Cap. It gives us information about the activity level of the brain.
- **SCP.** Slow Cortical Potentials (SCP's) are very slow electrical shifts in brain activity. They change periodically from electrically negative to positive and play a significant role in the regulation of attention.
- **EMG.** The ElectroMyoGraphy (EMG) is the recording of the electrical activity produced when muscles are contracted.
- **ECG.** An electrocardiography (ECG) is the electro-physiological measurement of the heart muscle signal. It is used for measuring the heart rate and the regularity of heartbeats.
- **BVP.** A photoplethysmography (PPG) transmits an infrared light through the tissue and measures the absorption of this light by the blood flowing through the vessels. It can be used to sense the rate of blood flow as controlled by the heart beats, also called Blood Volume Pulse (BVP).
- **HRV.** Heart rate variability (HRV) is the variation in time between consecutive heartbeats. The acceleration and deceleration of the heart rate reflects the body's ability to self regulate and maintain homeostasis.
- **Respiration.** By measuring respiration, you can tell something about the quality of breathing, such as frequency, intensity and duration.
- **SC.** Skin Conductance (SC) is the measurement of the electrical conductivity of the skin. It reflects the level of psychological or physiological arousal, elicited by cognition or emotions.
- **Peripheral Temperature.** Peripheral skin temperature is a reflection of the blood flow through the vessels under the skin and serves as an indicator of sympathetic activity in the body. When the body is sympathetically activated, blood vessels in

[4] https://www.mindmedia.com/

the periphery constrict and blood shifts away from the hands and feet into the muscles of the arms, legs and into the head.

- **EOG.** ElectroOculoGraphy (EOG) measures eye movement. As the eye moves, the EOG measurement detects changes in the electrical charge between the cornea and retina. In this way, eye movement can be deducted.

- **Accelerometer.** It measures the static acceleration of gravity in applications where the orientation of segments or parts of the human body is determined. In addition, it measures dynamic acceleration resulting from motion, shock or vibration.

3.3 Related research projects

The tools, that have been presented before, are the basis of the research that have been carried out in the last years to create emotion-aware systems for e-learning platforms. Each implementation use one or a combination of these tools depending on the different objectives or the researchers and depending on the features of the e-learning platform that is used.

The majority of these investigations have in common the use of a dashboard to show the emotions captured in the learning process. As we explained before, these works do not only look for collecting as much as possible data about students emotions, otherwise they pretend to show the data in an effective way to be used by teachers and students to improve their results.

The most relevant researches on the subject are presented below.

3.3.1 AffectVis, an affect-aware dashboard for students

The aim of this research [16] is to evaluate four possible visualizations that represent the emotional states of the students in a classroom. These visualizations are carried out in order to be used by the own students to improve their affective state and consequently their academic results.

The authors define the perceived usefulness of these dashboards as *"the perception of students about the importance of each one of the visualizations for the learning process"*. For this reason, during the research performance, the dashboards have been improved with the help of feedback from students who had knowledge of visualization techniques.

The experiment took place in two universities with two different kind of students depending if they had previous knowledge in the use of dashboards. The students had to complete different tasks as they usually do during their lessons but, when they finished, they had to answer a set of questions in an on-line survey to indicate their affective state during the lesson. In the same way, context information about the students such as gender, age, and previous knowledge was asked.

The evaluation consisted of a series of questions about the ease of use, perceived usefulness, or interpretation of each visualization. Therefore, the tool used to capture students' emotions at the end of the activity and during the use of the visualizations was a self-reporting questionnaire.

These visualizations formed the dashboard named AffectVis, which objective was *"to allow learners to reflect their own affective states and their connection with specific learning activities"*. To do so, four types of visualizations were developed using JavaScript language in order to be used in web browsers. Each visualization is presented below.

- **Radial Visualization.** This visualization uses a set of polar bars that shows the average of student's emotions that appear when performing each different kind of activity.

- **Timeline Visualization.** The timeline visualization shows the evolution of time dedication of each student during the course, as well as the average time dedication and the emotional evolution of the whole class.

- **Heatmap Visualization.** This visualization shows the evolution of each student's emotions by day, week, and month, and also shows the intensity of each of them.

- **Scatterplot Visualization.** In this visualization the abscissa axis corresponds to the day and time when the emotion was captured and the ordinate axis with each emotion and its frequency.

The results obtained in this research shows that simpler visualizations as the timeline seems more useful to students with no knowledge in visualization techniques. For this reason, the conclusion is visualizations have to been designed in a way that will be understandable for most people.

3.3.2 TEA Model, a new model of emotions

The objective of this paper [56] was to discover students emotion during the course for identifying problems or difficulties in their learning process. This emotion data was presented to the students in a variety of visualizations which looked for promoting self-reflection in order to improve their academic results. Once visualizations were created, they were integrated in a learning system named PresenceClick [57], which is able to register the interactions occurred in a class between students and their teacher.

With that in mind, authors propose a new model of emotions, named TEA Model (Twelve Emotions in Academia), which is the result of combination and improvement of previous emotions models developed for learning environments. This model is based on a positive/negative emotion spectrum with six positive emotions (enjoyment, hope, pride, confidence, excitement and interest) and six negative ones (anxiety, anger, shame, hopelessness, boredom and frustration).

To capture students emotions, self-reporting tool was used adapted to the proposed model. The questionnaire consisted of evaluating sentences referred to the lesson such as *"I enjoy working in the subject activities during the week"*, and on the other hand, students had to evaluate the intensity of the emotions they had experienced. These evaluations were based on a six grade scale to define the student's level of agreement with the sentences.

The collected data were presented to the students through different types of visualizations. Firstly, evolution of emotions over time is shown in 3.2 with three visualizations: bubbles chart, stacked bars and box-plot chart. Each emotion was assigned a colour to be better understood by students. The aim of these was to allow students to know which emotions had felt according to the moment of the course and to compare their own emotions with the average of the group.

Secondly, a visualization based on a squares schema shows specifically the results of the self-reported tests. The squares, shown in 3.3, represents the negative or positive answer of each person to each question, standing out the answer of the student that is using the dashboard. This way, the student can analyze his emotional evolution during the course by comparing with the others students.

By showing the dashboard to the students who participated in the study, it was concluded that most of them found the visualizations useful for improving their learning. In spite of this, most of the students did not find the emotion recording system comfortable and motivating enough to use it daily. These considerations were born in mind to implement them in a new module for the PresenceClick system named EmotionModule.

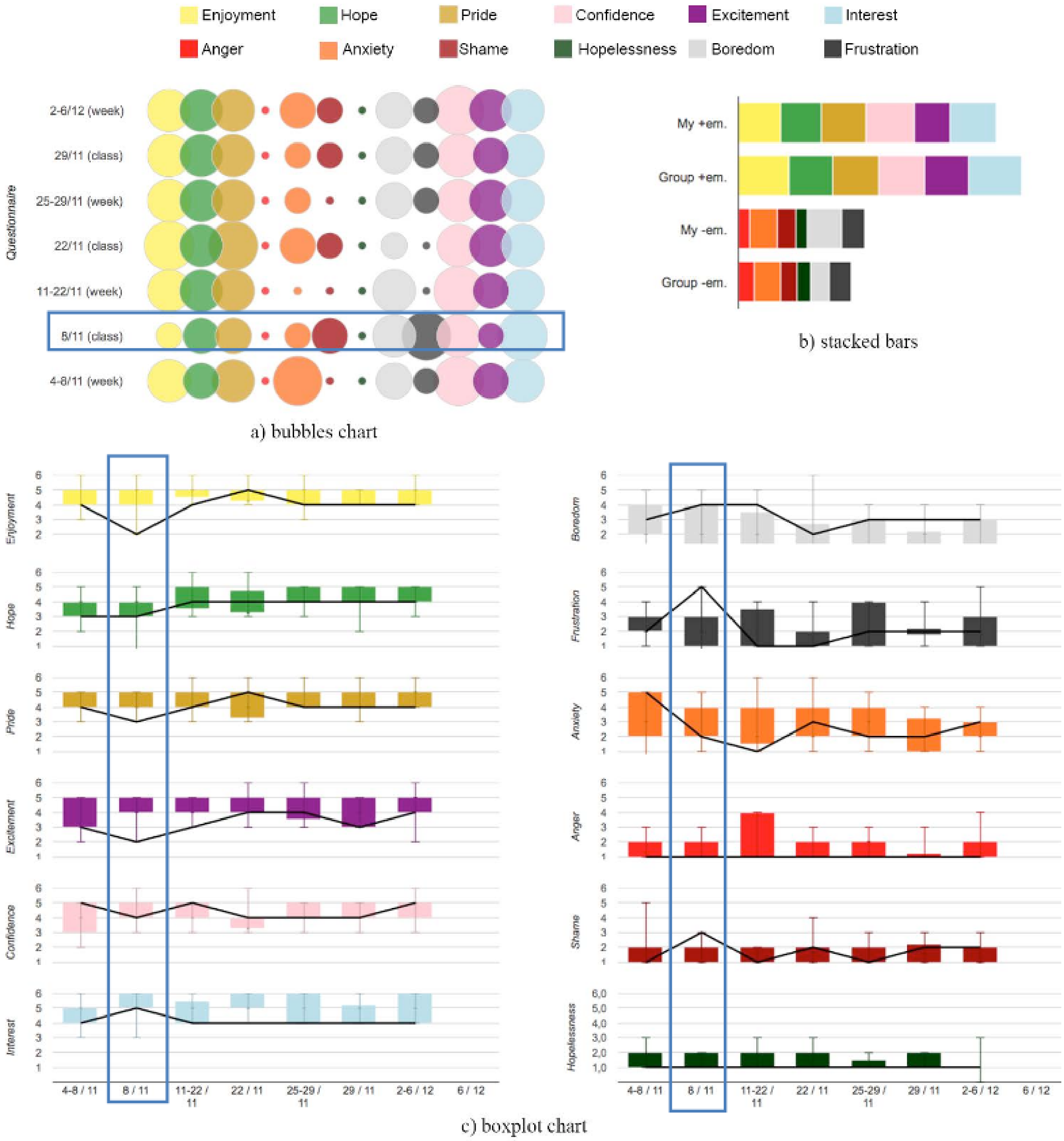

Figure 3.2: TEA emotion visualization graphs [56]

Finally, once made the implementation of this module, its usefulness was tested during an entire semester. The results showed that more than a half of the students found positive to analyze their emotions. Moreover, almost a 75% of students improve their academic results, confirming the usefulness of the dashboard.

Figure 3.3: TEA emotion visualization of the squares schema [56]

3.3.3 MADE Teacher's Dashboard

The following research presents the development of a web application called MADE (Multi-Dimensional Emotion Analytics) Teacher's Dashboard [26]. Its purpose was to collect and show the emotions experienced by students of an online learning platform in order to provide a tool with which teachers can adapt lessons to the students' mood.

As we mentioned before in the introduction, this kind of developments born out of Kort el al.'s model of affective in education. This model talks about emotional phases that all student have to pass through in a learning process. Knowing in which phase are each student, teacher can provide emotional support that helps students in getting over learning difficulties.

To do so, technologies based on artificial intelligence were chosen as a tool to capture emotions. In this case, it was used a JavaScript library named Clmtrackr, which allowed to detect facial expressions of emotions through a web-cam. This tool was able to recognise six kinds of emotions (disgust, fear, joy, surprise, sadness and anger) and their intensity.

The implementation of the dashboard was carried out on an existing mathematical learning platform named MADE Ratio. In this way, with the help of the emotion detection library, it was possible to relate the emotional state of the student with the learning activity he was doing. The proposed visualization, shown in 3.4 consisted of a graph showing the different emotions of the student and their evolution over time, including a pie graph showing the proportion of emotions detected in a specific period of time.

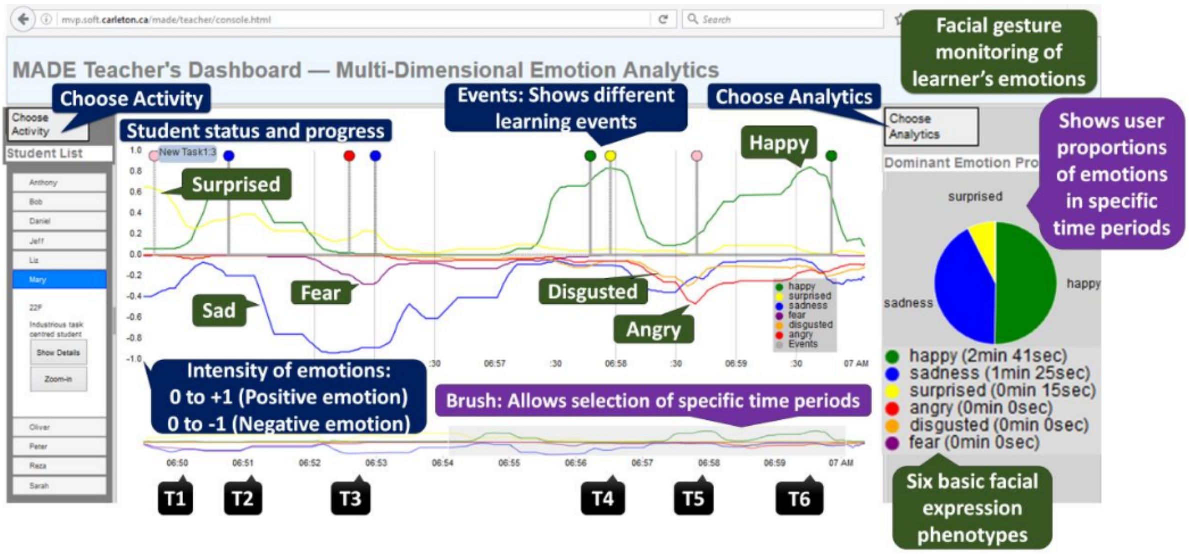

Figure 3.4: The MADE Teacher's Dashboard [26]

The conclusion of this project is that using appropriate technologies such as emotion detection and web visualization tools can overcome the difficulties of affective communication that arise between a teacher and a student in an e-learning platform.

3.3.4 EMODA (EMOtion DAshboard)

The starting point of this work, named EMODA: a Tutor Oriented Multimodal and Contextual Emotional Dashboard [20], was to improve the learning outcomes of language e-learning platform students. For this purpose, students' emotions were captured through different techniques of emotion recognition and a dashboard with the results was developed to help the teacher.

The platform used, called Speakplus [5], allows distance learning of languages through two types of tasks: exercises and tasks to be performed individually by the students, and live lessons with the teacher through streaming video. In this research, more importance was given to live sessions, capturing students' emotions in four ways: audio, video, self-report, and interaction trace.

First, a web form was developed with which students could record their emotions before and after the lessons. This interface was divided in two parts. On the one hand, a two-dimensional graph to select the positive or negative levels of valence and arousal. On the other hand, an emotion selector with which the intensity of each emotion (neutral, happiness, surprise, sadness, disgust, contempt, and anger) could be measured on a scale from 1 to 100.

Secondly, the video sessions of both the students and the teacher were recorded. An algorithm was implemented that divided the videos into frames to send them to Microsoft's emotion recognition service (MS API) [38]. This service recognises facial expressions in a video and is able to classify emotions using a discrete model of emotions. For each frame sent, a JSON response was obtained with the emotions detected and their intensity.

Thirdly, for student recordings only, an algorithm was developed to extract the audio from the video recording and send it to the Beyond Verbal service (BV API) [71]. This service classifies the emotions from an audio file using a two-dimensional model of emotions (arousal and valence). The response of this service is an array of detected emotions associated with time intervals.

Finally, data were collected on interactions between students and teachers in order to contextualise emotion data. To do this, the actions carried out by each user were stored, as well as the elements that intervened in them, such as documents, presentations, or exercises, and the moment in which they took place.

Once data had been obtained, it was analyzed. In order to do this, since the models were different, it was necessary to adapt the audio and video results, leaving aside the self-reports. Finding a correlation between them, it was concluded that they were not correlated except by using the coefficient MIC. However, there was a relationship between the positive and negative peaks of the emotions recorded in both sources. These relationships in concrete moments were considered as *"more likely to represent effective emotional reactions during the learning session"*, so they were the basis of the visualizations implemented.

[5] https://speakplus.fr

The final result was a dashboard, shown in 3.5, through which teachers could access to the emotional data of individual students. This dashboard had three types of visualizations. The overall visualization included an objective (emotion recognition tools) and subjective (self-report) summary of the emotions experienced by the chosen student during the learning sessions. The timeline graph shows the contextual information collected along with the emotions that were recorded at each moment. Finally, the time segment information provides a more detailed picture of the different emotions captured at a selected moment.

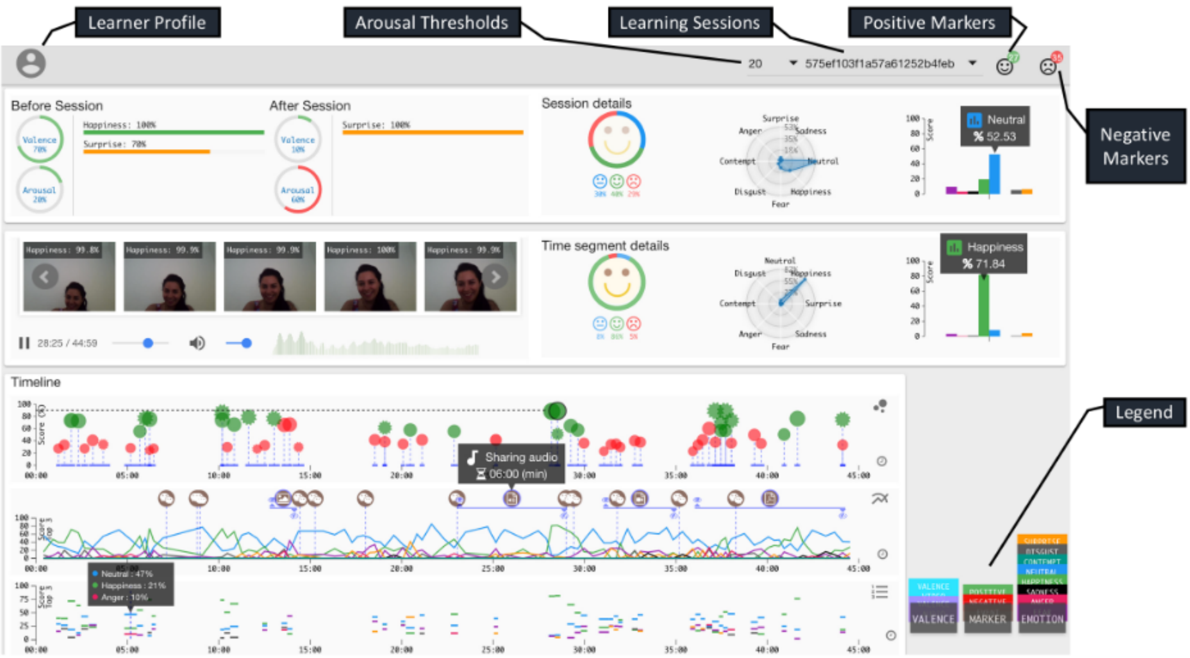

Figure 3.5: The EMODA dashboard [20]

As a conclusion, this project managed to develop a dashboard in which different sources of emotional information were unified, at the same time as contextual data were collected, in order to offer a more complete vision of the affective evolution of a student in an online learning platform. Apart from the achievements made in capturing emotions and their visualization, the evaluation of the teachers who tested the dashboard was positive. However, as in previous researches, users prefer simpler visualizations because they are easier to understand.

3.3.5 IntelliEye xMOOC Benchmark Tool

The last research we are going to analyze was aimed at evaluating the performance of gaze tracking tools to detect attention losses of students participating in an MOOC. The basis of this paper, titled Webcam-based Attention Tracking in Online Learning: A Feasibility Study [55], was to use the different research carried out in previous years on inattention prediction, applied to an online learning environment such as a Massive Open Online Course (MOOC). More specifically, the aim was to apply gaze detection for this purpose, overcoming the restrictions that this type of software normally imposed as a considerable time delay between detections.

The following tools were used to carry out this: two open-source browser-based software frameworks for gaze and face detection, WebGazer.js and tracking.js, and a third hardware-based solution named Tobii eye tracker [6]. In the same way, 50 tasks were defined, which can be performed by a learner while doing MOOC activities like watching a video, differentiated into three categories: face-miss tasks (e.g. take a sip from a cup of coffee), likely-face-miss tasks (e.g. take a note on a paper), and face-hit tasks (e.g. position yourself in the chair).

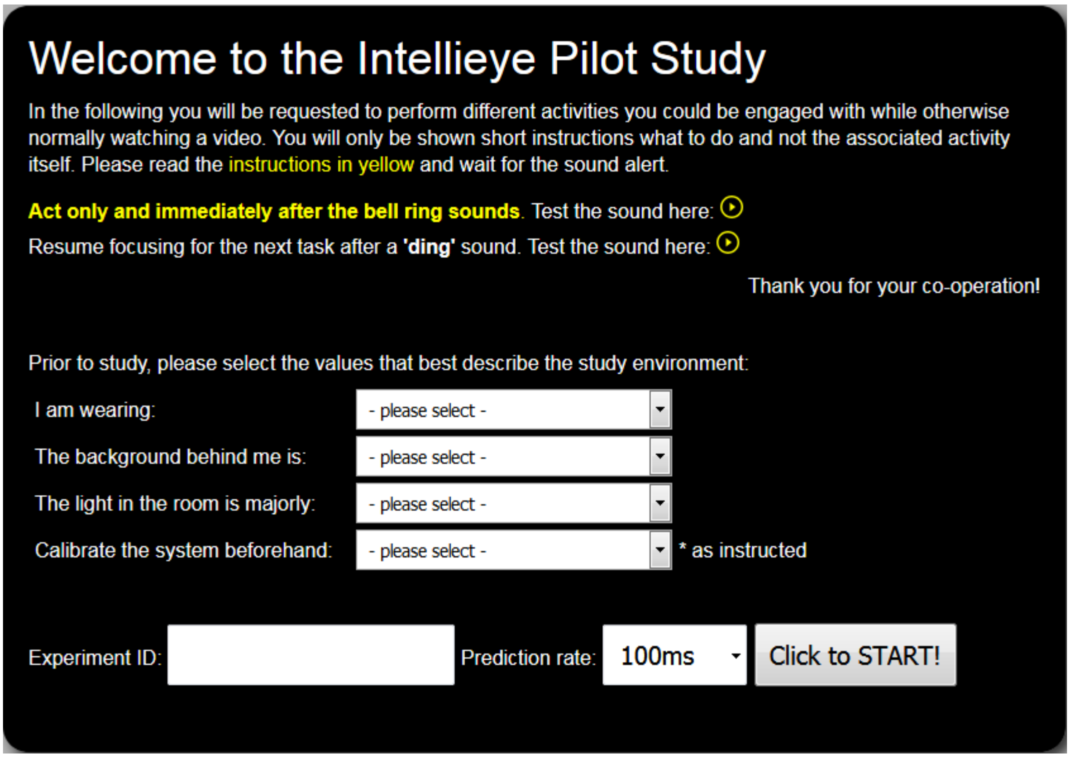

Figure 3.6: Opening screen of IntelliEye Benchmark Tool [55]

The experiment involved 20 participants who performed the 50 previously designed tasks while carrying out MOOC activities. The conclusion achieved proved that the professional dedicated hardware solution obtain better results than the open-source software-based solutions both in respect to detection performance and processing speed. Despite of this,

[6] https://www.tobiipro.com/product-listing/tobii-pro-x2-30/

hardware solution usage does not fit outside of a controlled lab setting mainly due to its cost.

The open-source web application developed, named IntelliEye Benchmark Tool, implements finally Webgazer engine because of being the best software-based solution of the two that have been tested. This application, shown in 3.6, is currently available on Github [7]. The way it works allows to carry out user studies where participants have to perform certain tasks in front of a webcam while it collects participant performance data.

3.4 Project basis for the development of an Emotion-Aware System for e-Learning

Once the main related research projects are known, we present our own vision for the implementation of an Emotion Aware E-learning system. This vision is based on the model developed by Kort et al. [33] which relates learning phases with emotions. Knowing the emotions that students experience every day as during lessons or assessment activities, it will be possible to adapt the contents, environment, and assessments in such a way as to improve the students' mood, motivation and consequently their academic results. Likewise, self-reflection and self-knowledge activities will be promoted among students, which will help them to better understand the emotions they experience, and therefore, to carry out actions for their improvement.

The development of this project will focus primarily on the collection of data related to students emotions, as well as other physical, personal and academic data that can be related to each other. Similarly, a dashboard will be developed for two differentiated profiles, students and teachers. In this way, everyone will have access to personalized data and visualizations that will help them understand both their academic performance and the evolution of their feelings during the course. Finally, part of the collected data will be used to adapt the working conditions in real time to students mood, through a Semantic Task Automation platform.

The motivation and objectives of each part of this development are explained below.

[7] https://github.com/trx350/xMOOC_benchmark

3.4.1 Data Collection

Data collection will be based primarily on the emotions experienced by students during the course. However, it is also necessary to collect other types of data such as the grades obtained, the percentage of time the lesson has been attended, or the level of stress experienced during an exam. The following tools will be used to obtain these data:

- **Emotions:** self-report and webcam-based emotion-recognition.
- **Attention Rate:** webcam-based gazer-recognition.
- **Stress Level:** neurophysiological tool to measure the skin electrical conductivity.
- **Academic and Personal Data:** e-learning platform.

3.4.2 Data Visualization

Data visualization will allow to establish cause and effect relationships from the collected data. For this purpose, a series of visualizations will be implemented on a dashboard that will allow to show information such as the evolution of emotions in different phases of the course, the evolution of grades related to emotions or the number of approved based on the attention rate.

This information will serve as an aid to improve the academic performance of both teachers and students, as well as to improve students' mood.

3.4.3 Smart Environment Adaptation

Finally, Smart Environment Adaption will serve as a first step in real time to try to improve students' mood while they take the lesson or perform assessment activities. For this purpose, some of the measures taken by the tools presented above, will be sent to the Semantic Task Automation platform to trigger actions such as playing relaxing music, changing the intensity of light, or recommending a break.

CHAPTER 4

Learning Analytics

The field of Learning Analytics has attracted the attention of many researchers in recent years. It is located at the intersection of learning and information technology and involves other related fields like business intelligence, web analytics, academic analytics, educational data mining, and action analytics. The main goal of Learning Analytics is to develop systems that harness educational data sets to support the learning experience. In this project, data collected from the emotion aware e-learning platform presented in Chapter 3 are being used to improve students mood and to perform best-fitted course activities. For this, a review of the existing papers in Learning Analytics is made, and then, our own vision about the topic is introduced.

4.1 Introduction

Today, continuous advances in computing, and more specifically in Artificial Intelligence [58], coupled with the emergence of numerous data capture techniques, have resulted in a growing trend to use both techniques to improve virtually any area of people's lives. Within this trend and closely related to the field of study of this project appears the concept of LA.

This term has been defined in numerous occasions, being the most recognized and used in the different research works the one coined by Siemens [63] and the one defined in the first international Conference on Learning Analytics and Knowledge (LAK 2011) [1]. The first one defines it as *"the use of intelligent data, learner-produced data, and analysis models to discover information and social connections, and to predict and advise on learning"*. The second one, given in LAK 2011, defines it as *"the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs"*.

The ultimate goal beyond these definitions is ultimately to improve the learning process through the use of new data. These new data complement and extend those obtained through traditional approaches such as the analysis of grades and attrition rates. This takes on even greater dimension if we consider the environment of an e-learning platform in which interactions such as comments in forums, time viewing content, or emotions experienced during learning are captured simply and on a large scale. In the same way, other barriers existing in traditional methodology are overcome, such as the low quality and scarcity of the data obtained and a significant delay between the moment in which symptoms are detected and the implementation of a solution.

The field of LA is formed by an intersection of different related research fields. As with its definition, there are several approaches that include different fields as the basis for LA. In this project, the conceptual framework elaborated by M.A. Chatti et al. [9] has been taken as a reference, as it is the one that best adapts to the objectives to be addressed. A summary of the related fields is presented below.

- **Academic Analytics.** This term, first coined by Goldstein & Katz [27], related the concepts of business intelligence, web analytics, and education. However, the most extended definition was described by Campbell & Oblinger [8] as *"an engine to make*

[1]https://tekri.athabascau.ca/analytics/

decisions or guide actions that consists of five steps: capture, report, predict, act, and refine".

- **Action Analytics.** This concept, also called Action Research, is based on the use of the data obtained in the learning process to investigate, reflect and propose solutions that improve the results achieved. Norris et al. [46] defined these methods as follows: *"analytics capabilities and practices that are powerful, immediate, and lead to outcomes that are useful to a wide variety of stakeholders".*

- **Educational Data Mining.** This field of research is based on the use of data mining techniques such as grouping, classification, association, visualization, statistics and prediction to discover models that improve the learning process. In recent years, a great deal of research has been done as reflected in the Journal of Educational Data Mining (JEDM) [2].

- **Recommender Systems.** These systems are responsible for drawing conclusions from data obtained through capture or prediction techniques to make recommendations. In this case, the recommendations can be addressed directly to the student or go through the teacher first.

- **Adaptive Learning.** This term groups together a set of solutions that arise as a result of a LA process. These solutions are based on the individual modelling of students to adapt learning techniques and processes to their states and specific needs in such a way that their academic results are improved.

Once converging fields in LA are known, it is important to understand the processes that take place.

4.2 Learning Analytics Process

Being a research field based on different subfields, with different approaches, and with different objectives, and due to its growing success among research communities, there are different ways or schemes to define a LA process.

The main existing processes have been collected and compared by T. Elias [18] in 4.1. This figure shows the steps that are carried out in each vision of the process and finally it

[2] https://jedm.educationaldatamining.org/index.php/JEDM

shows in the last column a compilation in which all the possible steps appear following a logical order.

Knowledge Continuum	Five Steps of Analytics	Web Analytics Objectives	Collective Applications Model	Processes of Learning Analytics
		Define goals	Select	Select
Data	Capture	Measure	Capture	Capture
Information	Report		Aggregate	Aggregate & Report
Knowledge	Predict		Process	Predict
Wisdom	Act	Use	Display	Use
	Refine			Refine
		Share		Share

Figure 4.1: Comparison of Analytics frameworks and models by T. Elias [18]

In this project, as explained in the previous section, we will focus on the approach proposed by M.A. Chatti et al. [9], because of its simplicity and efficiency in integrating and defining existing steps in the other approaches. This process is divided into three main steps: data collection and pre-processing, analytics and action, and post-processing.

1. **Data collection and pre-processing.** The first step should be to collect the most relevant data from the educational environment. These data can be very varied and come from several sources as seen in Chapter 3. Similarly, we may find a disproportionate amount of data if we work on an e-learning platform. For this reason, it is essential to pre-process the data in order to eliminate irrelevant data, anonymize private data if they exist, and transform the data into a format suitable for later use.

2. **Analytics and action.** In this step, different data analysis techniques must be applied, such as pattern discovery or visualization. Machine Learning (ML) techniques can also be applied to predict what effects the application of different methodologies, activities, etc. will have on the students. Once a solution has been analysed, understood and proposed to improve the learning process thanks to the information collected, it is time to carry it out.

3. **Post-processing.** Once solutions resulting from the analysis have been implemented, it is necessary to evaluate the process followed in order to improve it. This evaluation can result in obtaining new data that had not been taken into account, pre-processing better oriented to the final objective, defining new attributes and metrics, modifying analysis variables or choosing different prediction and analysis methods.

These steps occur cyclically as shown in 4.2.

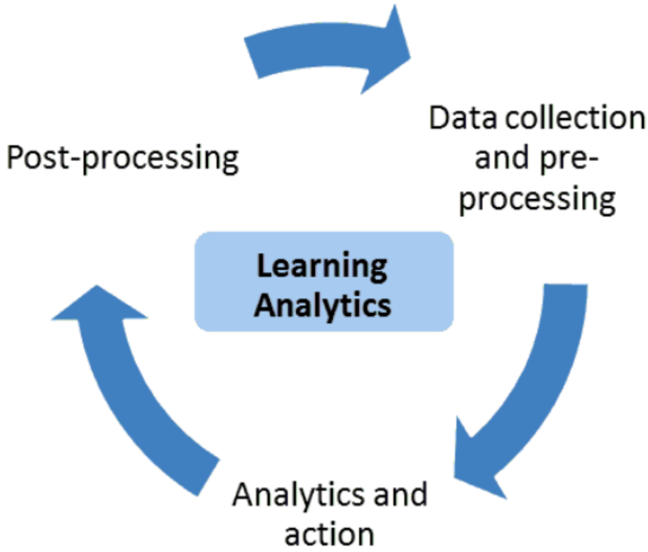

Figure 4.2: Learning Analytics Process by M.A. Chatti et al. [9]

4.3 Related research projects

As mentioned in 4.1, Learning Analytics (LA) is a research field with growing interest today. This has been reflected in the emergence of numerous LA-based research that seeks to improve the academic outcomes of students participating in experiments.

The most relevant researches on the subject are presented below.

4.3.1 Course Signals at Purdue: Using LA to Increase Student Success

The aim of this paper [2] was to ensure that students in their first year of college at Purdue do not abandon their studies and have a continuous orientation of their academic performance. To this end, an LA system, named Course Signals, was developed to allow the integration of each student's real-time performance data along with demographic information and academic history from past years. Through these data, predictions were made about the student's academic future.

Course Signals was defined as *"a student success system that allows faculty to provide meaningful feedback to student based on predictive models"*. The predictive algorithm, called the student success algorithm (SSA), used data from several university sources in addition to the Purdue students' own information. To make the predictions, four components were weighted: performance (grades obtained to date), effort (number of interactions with the e-learning platform), academic history (own and compared with other students in the same course), and the student's demographic characteristics. The predictions obtained were based on the signs of a traffic light to indicate the probability of success of the student.

These predictions were sent by email directly to the student or shared through tutorials. In addition, each student was provided with help resources depending on their needs. The functioning of this system allowed to obtain predictions based on multiple factors that happen in the context of a student, so the results were quite reliable.

The results obtained with the implementation of this system could not have been more revealing. The students who used Course Signals obtained better grades and a lower drop-out rate than their peers in the same semester who did not use this system. Similarly, different surveys were carried out among teachers and students in which the great usefulness of the system was manifested.

4.3.2 Affective States and State Tests to Predict Learning Outcomes

The goal of this project [50], entitled *"Affective States and State Tests: Investigating How Affect and Engagement during the School Year Predict End-of-Year Learning Outcomes"* was to investigate the relationship between affective state and student behaviour. For this purpose, different user interaction data were collected in an e-learning platform, which were later correlated with the activities carried out. Finally, the correlation between these data and the score obtained by the students in the final evaluation test was analysed.

The platform used for the development of the project was ASSISTments [3], and the aim of the course was to learn mathematics. During the time the students carried out the platform's activities, field measurements were made about the affective state and behaviour of a small group of students. These measurements were based on the student's work context, actions, utterances, facial expressions, body language, and interactions with teachers or fellow students. Thanks to this, the development of automatic affection detectors was achieved, capable of detecting this set of states: boredom, frustration, engaged concentra-

[3] https://www.assistments.org/

tion, confusion, off-task behaviour, gaming, and other. Similarly, metrics were stored from interactions with the platform such as the time taken to respond, the number of correct responses, or the number of times help was requested.

For the creation of the Machine Learning model, a separate detector was developed for each affective state or behavior. These detectors were evaluated using 5-fold and balanced subsets. Classification algorithms, including decision trees, step regression, JRip or Naive Bayes, were chosen because they were not prone to overfiting. Feature selection was conducted using forward selection with stepwise regression. This technique improves the model by adding features until the model is no longer improved. To avoid overfitting, cross-validated kappa with a value equal to or less than zero was used in single-feature models. The performance of the detectors was evaluated using the following metrics. The A' metric is the probability that the model will be able to discriminate a randomly chosen positive case from a randomly chosen negative case. Cohen's kappa assesses the degree to which the model is better than a random detection. Lastly, F-score measures the model's accuracy.

Once the learning model was developed, it was applied to a larger data set with interations of up to 1400 students. With the results obtained, Pearson's correlation was used to observe the relationship between the affective state and the grades obtained. The final result of the research confirmed that affect/behaviour and performance can be used together to predict a student's academic performance on final assessment tests. Therefore, the usefulness of this type of systems that integrate affective sensors for students with reporting and visualization tools for teachers was confirmed.

4.3.3 Using Learning Analytics to improve teamwork assessment

The objective of this study [25], named *"Using Learning Analytics to improve teamwork assessment"*, was to develop a system based on LA that would improve the learning process when students work in teams. This system allowed teachers to prevent problems in the teams and make decisions that favored the progress of the team.

The method on which the LA system was based is the CTMTC method (Comprehensive Training Model of the Teamwork Competence) [35]. This method *"allows the individual, group and result competencies of teamwork to be monitored and evaluated"*. In order to do this, the method proposed to collect data from three types of sources present in a virtual learning environment: e-learning platform forums, cloud storage platform interactions and files, and wiki tools. With this data, the system provided an interface for querying information about the team or a particular member. In addition, it allowed the creation of warning rules based on thresholds established according to the number of interactions.

The study was carried out with 110 students from the Technical University of Madrid divided into 19 groups and lasted three months. The data sources used were: Moodle, Dropbox, and an external wiki. The analysis was based on the observation of correlations between student interactions and individual academic performance assessed through inputs in the Moodle forum.

The most outstanding showed that there is a relationship between the number of interactions and the rating obtained. In spite of this, this information did not allow us to understand in depth how the internal performance of the group had been. For this reason, the relationship between active interactions (e.g. writing a message) and academic performance was analyzed. This new analysis confirmed that there was an even stronger relationship between these two facts.

To summarize, this LA system provided a teamwork conflict detector, an interactions database, and a performance analysis tool. Through this system, teachers were in a better position to conduct an evaluation of teamwork for each student.

4.4 Learning Analytics project model

The research collected in previous sections has served as a source of inspiration for the development of our own LA model. This model is closely linked to the data extraction capabilities of our Emotion Aware E-learning System. In the same way, it is also necessary to take into account the operative and temporal capacities for the application of our model, which differ from the majority of study cases seen, in which the number of students involved in the experiment is greater, as well as the time dedicated to the collection and analysis of data by the implemented system.

Once this has been explained, it is important to point out that our model is fundamentally oriented on emotions, and more specifically on the emotions experienced by the students when they are evaluated. The intensity of these emotions will be directly related to the grades obtained in each test. In the same way, other factors related to learning may be taken into account, such as the level of attention paid, the subject evaluated, or the time spent in solving the questionnaires. Finally, the possible relationship with other personal factors such as sex or age will also be studied.

Given the impossibility of knowing the final result of the analyses to be carried out, it cannot be possible to define specific objectives to be achieved. Nevertheless, it is necessary to define a set of ideas that help to orient the analysis towards certain characteristics or relationships. These ideas may be modified, expanded or discarded in the implementation phase, depending on the characteristics of the resulting data set.

The starting point for these ideas is the teacher's point of view, since he will be the only one who can access all the data collected, and will be in charge of adapting the academic activities based on the results obtained in this analysis. Some ideas for the analysis may be such as: searching for questionnaires with similar emotions; finding the correlation between different emotions and each test; obtaining the difference in emotions between the students with the best and worst grades; seeing the evolution of emotions throughout the course; classifying students by emotions experienced; or differentiating emotions collected based on sex or age.

These analyses seek to better understand the emotions experienced by students in the learning process, as well as their relationship with academic performance, which will surely serve for an improvement in the activities carried out in the course, and in the relationship with the students, in a distance learning environment.

CHAPTER 5

Architecture

In this Master Thesis we have developed a Emotion Aware system for e-learning platforms capable of detecting students mood and attention. This system implements several modules, that allow us the storage of different data coming from the interactions of the students with the platform, as well as its later analysis and visualization. It will also have a task automation module to make the most of all its features.

The main purpose of this chapter is to explain the architecture of this project, going through the design phase and the implementation details. Firstly, we will present a global vision about the project architecture in the overview, looking at the modules which form the system and its connections. Next, we will focus on each module explaining its function in this project.

5.1 Introduction

The development of this project is mainly based on a system for e-learning platforms capable of recognizing emotions. To make this possible, it is necessary, first of all, that the system can be integrated into the chosen e-learning platform, in our case Moodle. Secondly, the system must be able to recognize and store emotions during the user's interactions with the platform. These interactions can take place in a variety of ways, so different recognition modules with different characteristics are needed. Thirdly, data must become valuable information for users. For this reason, it is necessary to implement a dashboard that shows the data in an enriching way. Finally, if making automatic adaptations of the smart environment is wished, it is necessary to deploy a task automation system that perform actions triggered by the interactions with the e-learning platform.

To sumarize, this chapter is divided in five main sections. The overview of the system developed is presented in Section 5.2. Section 5.3 describes the e-learning platform used and its integration in this project. Section 5.4 details the system that includes all modules to capture, storage, and analyze students emotions. Section 5.5 presents a visualization module to show the information about the course and its students. Finally, Section 5.6 explains the operation of the task automation platform.

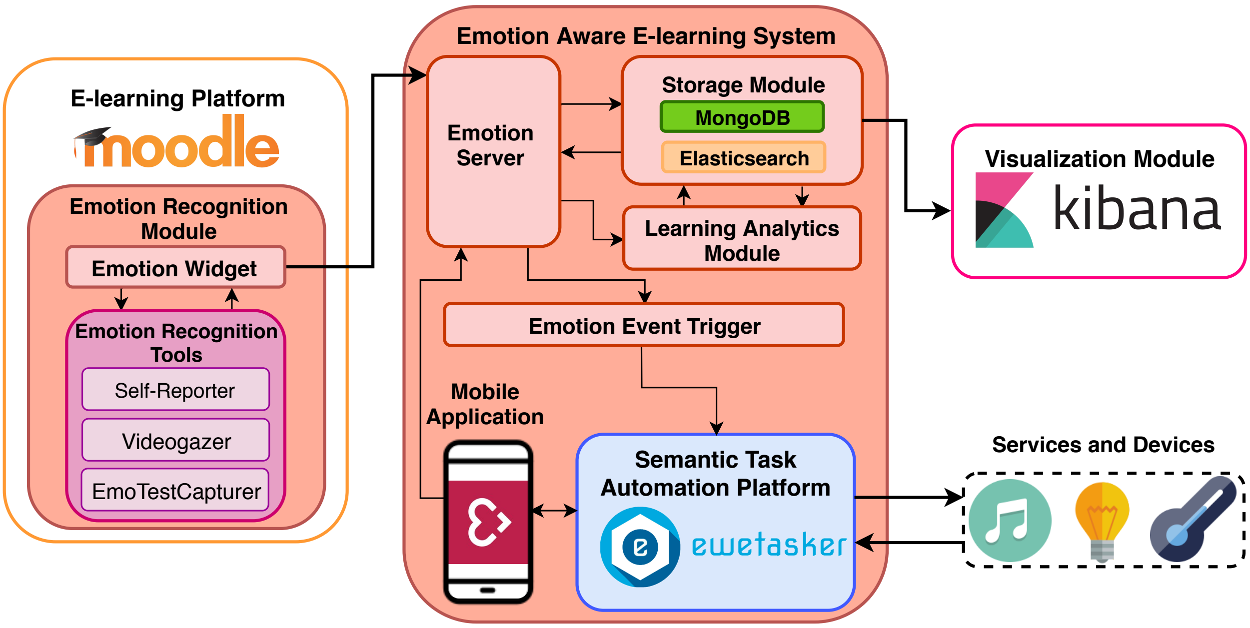

Figure 5.1: Architecture

5.2 Overview

In this section we will present the global architecture of the project, defining the different subsystems that participates in the entire system and describing interactions between modules. We can identify the following subsystems:

- ***Emotion Aware E-learning System:*** This is the main system of the project. Its main function is to collect and store academic and emotions data that is produced through students interactions with the ***E-learning Platform***. It is divided into two main modules, Emotion Recognition module and Storage module, which communicate with each other through the ***Emotion Server***. The ***Emotion Recognition Module*** implements a JavaScript web interface, named ***Emotion Widget***, that is inserted into the e-learning platform. Within it we find a series of sub-modules called ***Emotion Recognition Tools***: ***Self-Reporter*** submodule collects students own emotional status vision; ***Videogazer*** captures if students pay attention to the video lessons; and ***EmoTestCapturer*** recognizes students mood in real time when performing an exam. The data collected is stored in the ***Storage Module***, more precisely in ***Elasticsearch*** database. In this module we also can find a ***MongoDB***, which is in charge of storing data related to EmotionWidget working. Moreover, it implements a ***Learning Analytics*** module to perform classification, data exploration and visualization tasks using ML techniques. Lastly, the received data is also processed in the ***Emotion Event Trigger*** module to send events to the Semantic Task Automation Platform.

- ***E-learning Platform:*** This platform implements an online learning environment. In this project we use ***Moodle*** as a solution, due to the ease of integration with Emotion Aware E-learning System. This platform is responsible for hosting a course with their respective lessons and evaluation activities. It also stores students' academic data and implements an application logic that allows us to manage both the features of the course and those of its participants.

- ***Visualization Module:*** This module enables building personalized visualizations and dashboards. It is based on the ***Kibana*** platform due to its perfect adaptation to data stored in Elasticsearch. It implements two dashboards: one oriented to teachers, with visualizations based on the data of the set of students, and another one for students, with academic and emotional data.

- ***Semantic Task Automation Platform:*** This project has opted for the use of the ***Ewetasker*** platform, due to its simple integration with smart devices and services.

Thanks to its semantic foundation, an integration with Emotion Aware E-learning System has been implemented that allows the creation of automations based on events that occurred in the learning platform. This platform uses a series of sub-modules for its operation that have been developed, improved and adapted in this project.

- ***Mobile Application:*** This application, developed in Android, is a part of the Semantic Task Automation system which receives signals and sends events to Ewetasker, and also performs actions in the own device when certain conditions are met. In this project we use its integration with the ***Empatica*** smartband to measure the stress level of the students when they take an exam. These measurements are sent to the Emotion Aware E-learning System along with the rest of the data for their processing.

The modular architecture of the system and the relationships between modules are presented in Figure 5.1. The flow of data begins, as mentioned above, with the students interactions with the platform. The collected data is sent from the Emotion Recognition module to the Emotion Server. Once there, data is processed for later storage and to launch events to the Semantic Task Automation Platform. These events can trigger automation rules that order actions on both smart devices and services. Finally, the stored data is required by the Visualization Module for transformation and visualization.

5.3 E-learning Platform

The platform that serves as the basis for our project is the E-learning Platform, and more specifically Moodle. This platform allows us to create online courses quickly and easily, thanks to its friendly web interface. In addition, its use is widespread in the academic world, so it is not a barrier for teachers or students, giving relevance to our case study.

The Moodle release deployed in this project is 3.6.2, a version with identical features to the current version (3.6.3). Among the many capabilities and functionalities it provides, we have used the ones that best fit to achieve our objectives. This selection is summarized below:

- **Video.** This tool is a very effective way to complement face-to-face learning or directly give lessons from the platform. Because of this potential, Moodle allows two options for video playback: videos stored on the platform itself, or videos stored on external platforms. In this project the second option will be used, specifically, the videos will be hosted on the Youtube platform. This platform allows us to easily share videos with its own player through HTML code. These videos will be the base material for learning in the case studies carried out and will be used to measure the attention during playback.

- **Questionnaries.** The so-called Quiz are one of Moodle's strong points. This tool allows us to create evaluation activities with a wide variety of question types (multiple choice, true or false, or short answers among others). In addition, it implements features such as time limit or question mixer, very useful for the proper development of the evaluation. Finally, Moodle is responsible for storing the results of these questionnaires in multiple formats as we will see later. In this project we will use the questionnaires to assess the academic performance of students as well as to create a stress environment in which the emotions they experience can be measured.
- **Html Block.** This tool can be placed as a standard Moodle block and be used to add text, multimedia, or widgets to any page of the course. For this reason, this feature perfectly adapts to our project needs, and will be used to connect our course with the Emotion Aware E-learning System.
- **Course Managing.** The options that Moodle implements for the management and control of its courses are countless. However, in the scope of this project, we will mainly use the options related to the management of the students participating in the course. Through them, we will be able to add sets of students stored in csv files as well as obtain the results of the evaluation activities in the same format. These results will be part of the data set used in visualization and analysis using ML techniques.

By using these features, it is possible to create an online learning course, in which our system of emotion recognition can be integrated in such a way that it does not affect negatively the user experience of the platform, but it does improve their academic performance and mood, as we have proposed in the goals of the project.

5.4 Emotion Aware E-learning System

The main functions of the Emotion Aware E-learning System (from now on, named EAES) are to capture, process and store the data obtained from the E-learning Platform, in other words, it is data processing oriented. This is reflected in Figure 5.2, where the modules involved in the process, the origin of the data, and the path data follows until being stored or sent to the automation platform are shown.

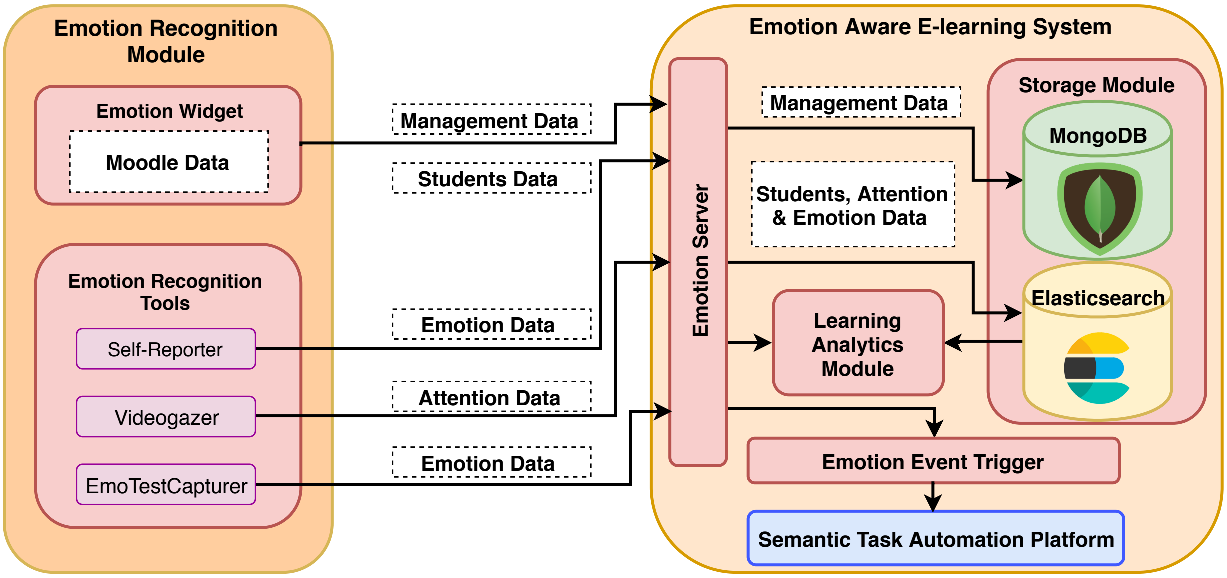

Figure 5.2: Emotion Aware E-learning System Flow Diagram

The following sections detail the operation and purpose of each module.

5.4.1 Emotion Recognition Module

The Emotion Recognition module captures the data used in the system. A web interface has been developed through a widget or web component. This component, called Emotion Widget, is inserted into Moodle using the HTML blocks explained above. By means of them, the design and functionalities of the Moodle courses web interface are maintained, adding the improvements and capabilities of our system in a non-introsive and efficient way.

The implementation of this module consists of the development of a web interface that connects Moodle with EAES. This web interface has to meet two main requirements. In the first place, it has to be able to implement the emotion recognition technologies chosen to carry out the proposed case studies. In the second place, it has to be a simple and user-friendly interface, taking into account the spatial limitations imposed by Moodle's own interface, and not hindering the usual e-learning platform users' workflow.

For these reasons, it was decided to develop a web interface in JavaScript, shown in Figure 5.3, which facilitates integration with the emotion recognition libraries Clmtracker and Webgazer. This web interface is oriented to two differentiated profiles: students and teachers. Through this division, it is possible to better understand the functionalities developed for each profile. The list of features that each view implements is shown below:

- **Student Management View.** This view allows students to register for EAES if they have not been registered by the teacher. Once logged into the system, it shows which tests have been enabled to be performed together with the emotion recognition tool. Finally, it also provides a link to access each student's personal dashboard.
- **Self-reporting View.** This view allows students to perform a self-report test indicating the level of intensity of the emotions they have experienced during the day.
- **Video Gazer View.** This view allows students to follow video lessons using the VideoGazer sub-module to measure their attention.
- **Emotion Test View.** This view allows students to perform Moodle quizzes while collecting the emotions they experience through the EmoTestCapturer sub-module.
- **Emotion Automation View.** This view allows students to learn from course slides while sending the emotions they experience to the automation platform in order to adapt the smart environment.
- **Teacher Management View.** This view allows teacher to show enabled tests and to enable tests new ones to be performed together with the emotion recognition tool. Moreover, it provides a tool for uploading csv files with students and grades data so that they can be ingested by EAES for processing and later viewing. Lastly, it also provides a link to access teachers's course dashboard.

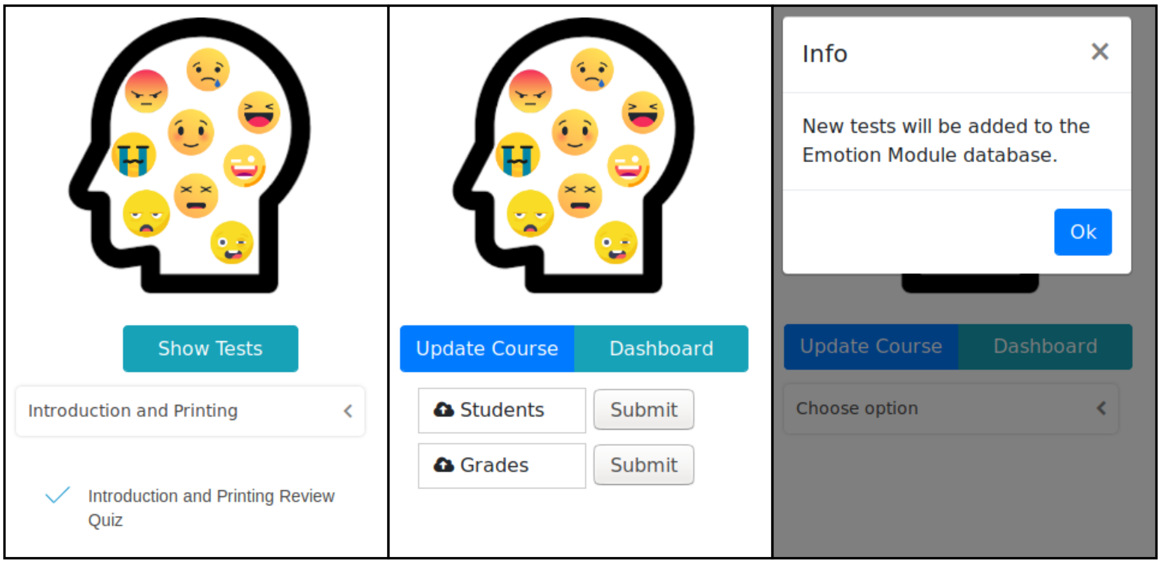

Figure 5.3: Student Management and Teacher Management example views

As we have mentioned, this module implements each of the sub-modules in charge of collecting a different type of data. The set of these sub-modules is called Emotion Recognition Tools and will be explained in more detail in Section 5.4.1.1.

5.4.1.1 Emotion Recognition Tools

The Emotion Recognition Tools are the fundamental part of the EAES since they provide the capabilities to obtain data related to the students emotions, the basis on which this project is built. Within this set of tools we have three kinds, which address the needs and use cases that we have identified during the previous study conducted in Chapter 3, and the objectives and case studies themselves raised for this work. The development, functionality, and operation of each tool are presented below.

Self-Reporter. This submodule implements a form in which students can record the emotions they are experiencing each day of the course, as well as the level of intensity of these emotions. This data allows us to perform a constant monitoring of students mood throughout the course using real metrics that have their origin in the subject itself. Similarly, these data allow us to get a correlation with measurements obtained from other sub-modules, and with the academic results achieved in each phase of the course.

In order to do this, it implements a web component, shown in Figure 5.5, that allows students to express the state of four emotions (anger, sadness, surprise, and happiness) at four levels of intensity (nil, low, medium, high). The four emotions have been chosen in such a way that they are similar to those provided by the EmoTestCapturer sub-module, thus facilitating further analysis and simplifying the daily self-reporting process for students. The operation of this module results in the storage in the EAES of each emotion intensity, along with the date of completion of the self-report and the student identifier number.

Videogazer. This submodule implements a tool that allows us to detect the attention of students in video lessons. These measurements will become even more important if they are being correlated with the emotion data and the grades obtained in the questionnaire about video topic. In addition, it is possible to relate the different phases of the video lesson with the attention and interest produced by each of them.

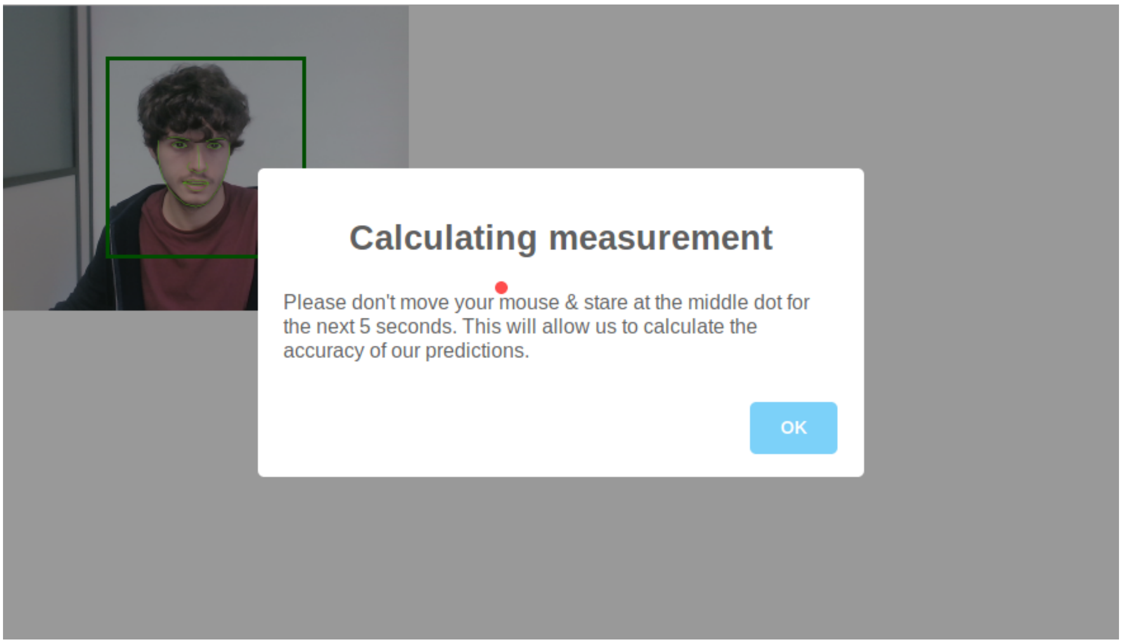

Figure 5.4: Videogazer Interface

To this end, a web component has been developed that allows videos to be viewed while measurements are being made. This component is based on the WebGazer library, which is able to infer the eye-gaze locations of web visitors on a page in real time using a webcam. To make use of this functionality, a framework has been defined in which the video player is inserted. Thanks to this frame, it is possible to obtain the location of the player in the web page, and consequently, to know if the position of user view corresponds with that location. In this way, it is possible to know when the user pays attention to the lesson. However, it is important to note that the tool must first be calibrated for proper operation.

Through the use of this tool, the following data related to the students attention are obtained: the average attention per second and the average attention to the video lesson. In addition, other types of data related to the video will be also stored in EAES, such as the viewing date, the duration of the video, the subject, or the student's identifier number.

EmoTestCapturer. This sub-module implements a tool that allows us to capture the emotion of students when they are solving a questionnaire. As in the other two submodules, the emotion data collected are more interesting if they are related to other data, such as the evolution of the mood during the week prior to the evaluation, or the attention given to the lesson corresponding to the test.

For this purpose, a web component has been developed that is activated when students take a test. This component is empowered by Clmtracker library, which is able to detect the emotion of users through the position of the coordinates that draw their face. Like the Videogazer module, it makes use of a webcam for its function.

The data obtained through this tool are numerous. Firstly, as mentioned above, the evolution per second of each of the four emotions (anger, sadness, surprise, and happiness) is collected on a scale from zero to one, during test time. Secondly, the average of each emotion is obtained. Finally, other types of data related to the video will be also stored in EAES, such as the duration of the questionnaire, the subject, or the student's identifier.

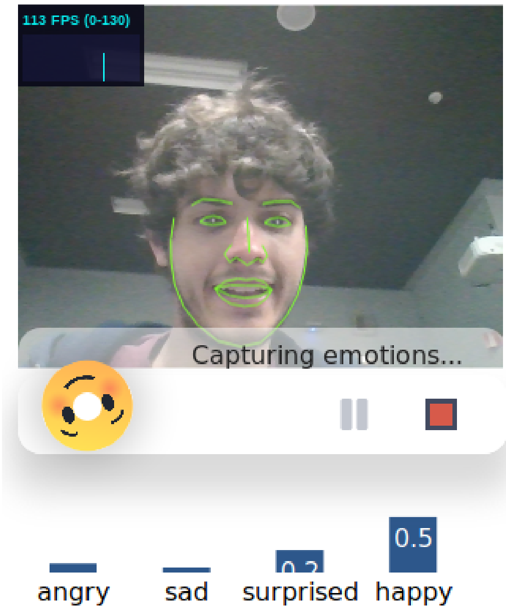

Figure 5.5: EmoTestCapturer Interface

5.4.2 Emotion Server

The main function of the Emotion Server is to connect the Emotion Recognition module to the EAES. To do so, it implements a REST API that is in charge of storing, updating, and returning the necessary data for the operation of the Emotion Widget. In the same manner, it is in charge of processing the data obtained by this module to be saved in the storage module, in such a way that a future visualization of them is facilitated. Lastly, it acts as a proxy for the Emotion Event Trigger module, that processes and sends them to the automation platform.

This server, developed in NodeJS, implements three models developed specifically to interact with the databases that make up the Storage Module. These three models are summarized below. First, a model that defines the operations that can be carried out in MongoDB. Second, a model for data storage in Elasticsearch. And finally, the model for files, that, by means of two python scripts powered by pandas, is in charge of processing the csv files uploaded to the Emotion Recognition module. These files can contain two types of information: the personal information of the students or the grades obtained in each questionnaire. This processing allows us the data ingestion by Elasticsearch.

5.4.3 Learning Analytics Module

The objective of this module is to carry out analysis tasks that allow the teacher to better understand the evolution of the students mood in the performance of the course activities, as well as to detect anomalous cases, trends and correlations. In order to achieve this, it makes use of the capabilities provided by Pandas, Numpy and Scikit-learn libraries for Python.

First, it obtains the data of each student stored in Elasticsearch, and saves it in csv files. Secondly, it preprocesses them using Pandas and stores them in a single dataset called result.csv in which each row corresponds to a student. Finally, it performs an exploratory analysis of the data, using ML techniques, including classification using K-Means, finding correlations between features, or creating new features.

The results of this analysis are shown to the teacher as visualizations. These visualizations are generated according to the different features of the data, and can be displayed as tables or bar charts. To facilitate teacher access to this information, each visualization is saved in an image file, an when process is finished all this files are stored in a zip file that can be downloaded later.

5.4.4 Emotion Event Trigger Module

This module has been developed with the purpose of launching events that trigger actions in the students smart environment through the Semantic Task Automation Platform. These events are generated by the module from the data of the students interactions with the e-learning platform. The final objective is to adapt the workspace in real time based on the emotions experienced by each student when carrying out activities related to the course.

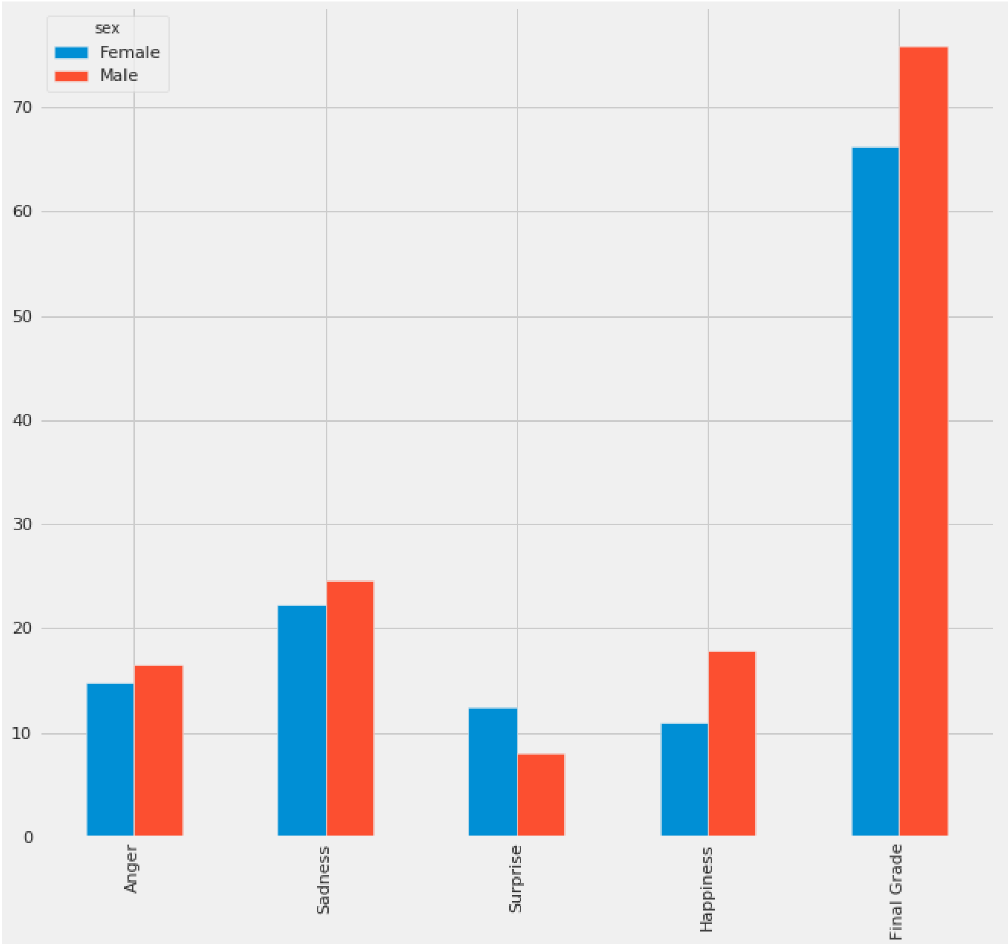

Figure 5.6: Learning Analytics Bar Chart example

Due to the large amount of data coming to EAES from different sources, it is necessary to process the data in order to infer the most relevant occurrences in a given period of time. Like this, the automation platform is not overloaded with events, and efficient and meaningful automations are carried out, in such a way that the purpose of improving the student's well-being can be fulfilled. This process includes mainly the two following tasks:

- **Calculating averages by minute.** Features extracted from the data whose occurrence is recurrent or continuous, and which are not important in themselves. However, put in the perspective of one minute, if the calculus of their average overcomes a threshold, it may be sufficiently important to trigger automations (e.g. attention average or emotions average).
- **Extraction of relevant features.** Features extracted from the data that, given the relevance of their occurrence, may be important enough to launch events (e.g. grades obtained, emotions self-reported or course changes).

Listing 5.1: Event generated by the average detection of an emotion by Emotion Recognition Module.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix ewe: <http://gsi.dit.upm.es/ontologies/ewe/ns/>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix string: <http://www.w3.org/2000/10/swap/string#>.
@prefix math: <http://www.w3.org/2000/10/swap/math#>.

ewe:EmotionDetectedElearning rdf:type ewe:EmotionDetectedElearning;
    rdf:type ewe:Event;
    rdfs:domain ewe:EmotionAwareElearningSystem.
ewe:EmotionDetected ewe:hasParameter ewe:ElearningEmotion. https://www.
   overleaf.com/project/5c49ac96fed0274904532954
ewe:ElearningEmotion rdf:type ewe:ElearningEmotion;
    rdf:value "sadness".
ewe:EmotionDetected ewe:hasParameter ewe:EmotionIntensity.
ewe:EmotionIntensity rdf:type ewe:EmotionIntensity;
    rdf:value 0.84 .

{ ?A ?B ?C } => { ?A ?B ?C }.
```

Given the semantic nature of the automation platform used for the development of this project, it has been necessary to implement through the EWE ontology a channel for EAES. This channel, explained in Section 5.6, specifies the events that can be launched by the Emotion Event Trigger module, along with their respective parameters. List 5.4 shows an example of an event written in Notation3. The event has been generated by the detection of *"sadness"* with an average intensity of 84% in a minute, when a student is reading the course slides .

5.4.5 Storage Module

The storage module consists of two databases, MongoDB and Elasticsearch, which connect to the Emotion Recognition module via the Emotion Server. On the one hand, the data needed for the system logic is stored, and, on the other hand, the data obtained from user interactions is saved. To learn more about the features implemented by each database and their importance in the project, they are presented separately below.

5.4.5.1 MongoDB

MongoDB Storage is used to save course data extracted from the Moodle website using the Emotion Recognition module. In the same way, it also stores the names of the students registered in the EAES, together with their identifier. Given the different features and purposes it has been necessary to create two collections in the database: courses and users.

The courses collection contains all the information related to the courses, from their name to the topics they cover, including evaluation activities and lessons. Once this information is stored, the teacher can update it to enable emotion recognition tools for each activity, as well as to add new activities or topics. On the other hand, the users collection is used by the EAES to get the identifier that is needed to index the data in Elasticsearch when students perform academic activities.

List 5.2 shows an example of the JSON schema that follows the course information stored in MongoDB. The stored course is called *"PYLB"*, and is composed of several topics. The first of them is entitled *"Introduction and Printing"* and has enabled the emotion recognition for the *"Introduction"* test.

Listing 5.2: JSON course data schema.

```
{
  id: ObjectId("507f1f77bcf86cd799439011"),
  "course": "PYLB",
  "topics": {
      "Introduction and Printing": {
          "tests": {
              "Introduction":{
                  "enabled":true
              }, ...
}
```

5.4.5.2 Elasticsearch

Elasticsearch Storage is used to store the collected data through the interaction of the students with the Moodle platform, and more specifically with the Emotion Recognition module. This data contains all the information about the students mood, personal information, and grades. This is how the information saved can be accesed to make visualizations or to perform a Learning Analytics process as we have seen.

The way data is stored in Elasticsearch is as follows. First, the information is divided into indexes. In our case, the name of the indexes has the following format, *"user-idnumber"* (e.g. user-12345678), so there is an index for each student. Within the indexes, the data are stored in documents, which contain fields where the information is finally stored. These documents are created or updated depending on the purpose of the information they contain. For example, the document that contains information of each student grades is updated as the tests are corrected. On the other hand, the document that contains the emotions self-reported by a student is created every day, and it is stored along with self-report documents from other dates.

In Elasticsearch it is convenient to specify a mapping for the indexes that are going to be created. A mapping is the process of defining how a document, and the fields it contains, are stored and indexed. In our project we will use the mapping shown in List 5.3.

Listing 5.3: Elasticsearch Mapping defined for the data ingestion of the EAES.

```
{
    "mappings":{
        "_doc":{
          "properties":{
                "type": { "type": "keyword" },
                "idnumber":{"type": "keyword"},
                "sex":{ "type": "text" },
                "age": { "type": "integer" },
                "date": { "type": "date" },
                "location": {"type": "geo_point"},
                "course": { "type": "keyword" },
                "topic": { "type": "text" },
                "test": { "type": "text" },
                "grade": { "type": "float" },
                "data_videogazer": { "type": "object" },
                "duration": { "type": "integer" },
                "attention_average": { "type": "float" },
                "data_emocapturer": { "type": "object" },
                "anger_average": { "type": "float" },
                "sadness_average": { "type": "float" },
                "surprise_average": { "type": "float" },
                "happiness_average": { "type": "float" },
                "eda_level":{ "type": "integer" }
            }
    ...
}
```

The mapping shows all possible fields that can be used in our indexes. However, not all fields are used at once. For the purpose of defining different kinds of document, Elasticsearch provides the *“type”* field. The different types defined in our project are detailed below:

- **Personal.** Includes student’s personal information, such as name, age, gender, or location. It also includes the grades obtained in each course, topic and test.
- **Self-Report.** Includes the emotions reported by the students, their intensity, and the date on which they were collected.
- **Video.** Includes the average attention, per second and per video, the video duration and the exact date and time in which video lesson has been watched.
- **Test.** Includes the emotions intensity per second, the emotions average, the test duration and the exact date and time in which test has been performed.

5.5 Visualization Module

The visualization module is a vital part for achieving the objectives proposed in our project, as it is in charge of showing the data collected by EAES. This data must be shown in an accessible and easy to understand way, without the need of previous knowledge by the users. To make this possible, Kibana has been selected as an implementation tool for these visualizations. Among its many advantages is its ease of integration with Elasticsearch.

Kibana provides a web-based environment, on which we can implement our visualizations. To do this, it is necessary to define an index-pattern, which will serve Kibana as a guide to add Elasticsearch indexes with the same name pattern. In our case, the index-patern is *“user-*”*. Through it, Kibana can get an idea of the type of data that these indexes are going to contain, since it accesses the mapping explained in the previous section. It will also serve to automatically add so many indexes that adjust to this pattern, without performing any manual operation.

The range of visualizations that can be carried out with Kibana is wide, as shown in Figure 5.7. From all of these options, those that best fit the data obtained as well as the objectives of the project have been selected. The visualizations implemented are summarized below:

- **Area.** Shows the evolution of a variable through the coloring of the resulting areas. Project uses: average self-report emotions per time.

- **Data Table.** Shows specific statistics compared to others. Project uses: average grades, maximum grades and minimum grades per topic.

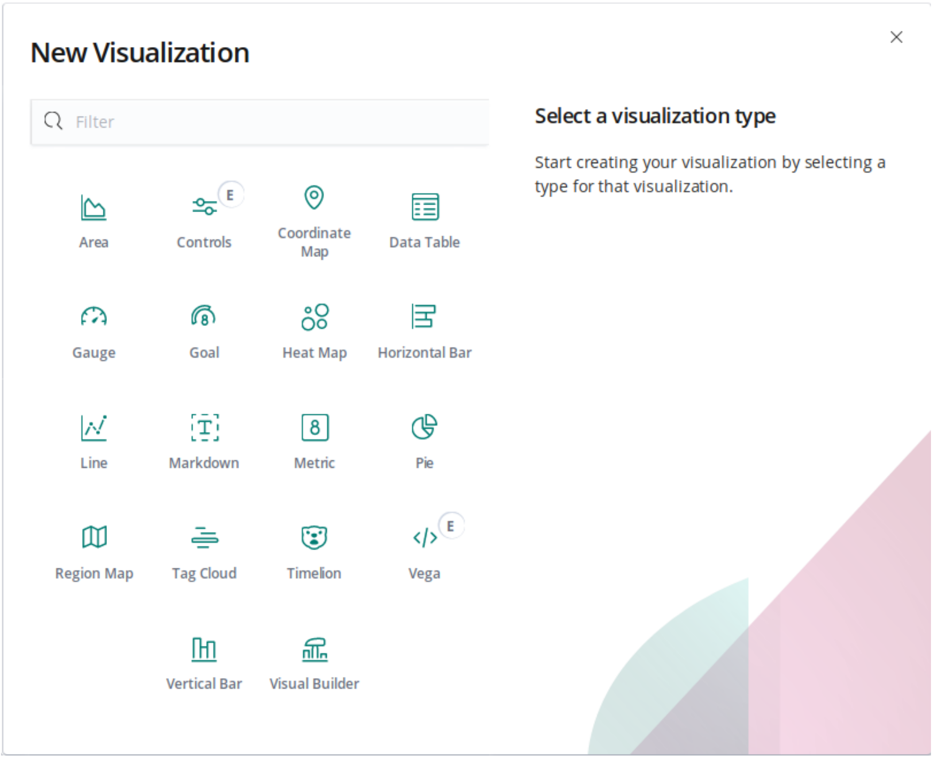

Figure 5.7: Kibana visualization components

- **Gauge.** Shows an unique measure, usually a percentage. Project uses: number of students enrolled.

- **Goal.** Shows the percentage of times a goal has been achieved. Project uses: success rate.

- **Heatmap.** Shows the number of times a value is found within defined ranges. Project uses: grade levels heatmap.

- **Bar.** Shows the number of times a feature appears, or its mean value. They can also be stacked with similar features. Project uses: average attention per video, grade average per topic, or average emotions per test. Example shown in Figure 5.8.

- **Line.** Shows the evolution of a variable through lines. It is also possible to show the evolution of several features of a same time period. Project uses: emotion evolution per minute in a test, average attention per second, or average self-report emotions per time.

- **Pie.** Shows proportions for the different ranges into which feature value can be divided. Project uses: number of students per gender, and number of students per grade.
- **Region.** Shows the students location on a world map. Project uses: students country.
- **Tag cloud.** Shows a set of tags extracted from the information stored. Project uses: idnumber tag cloud.

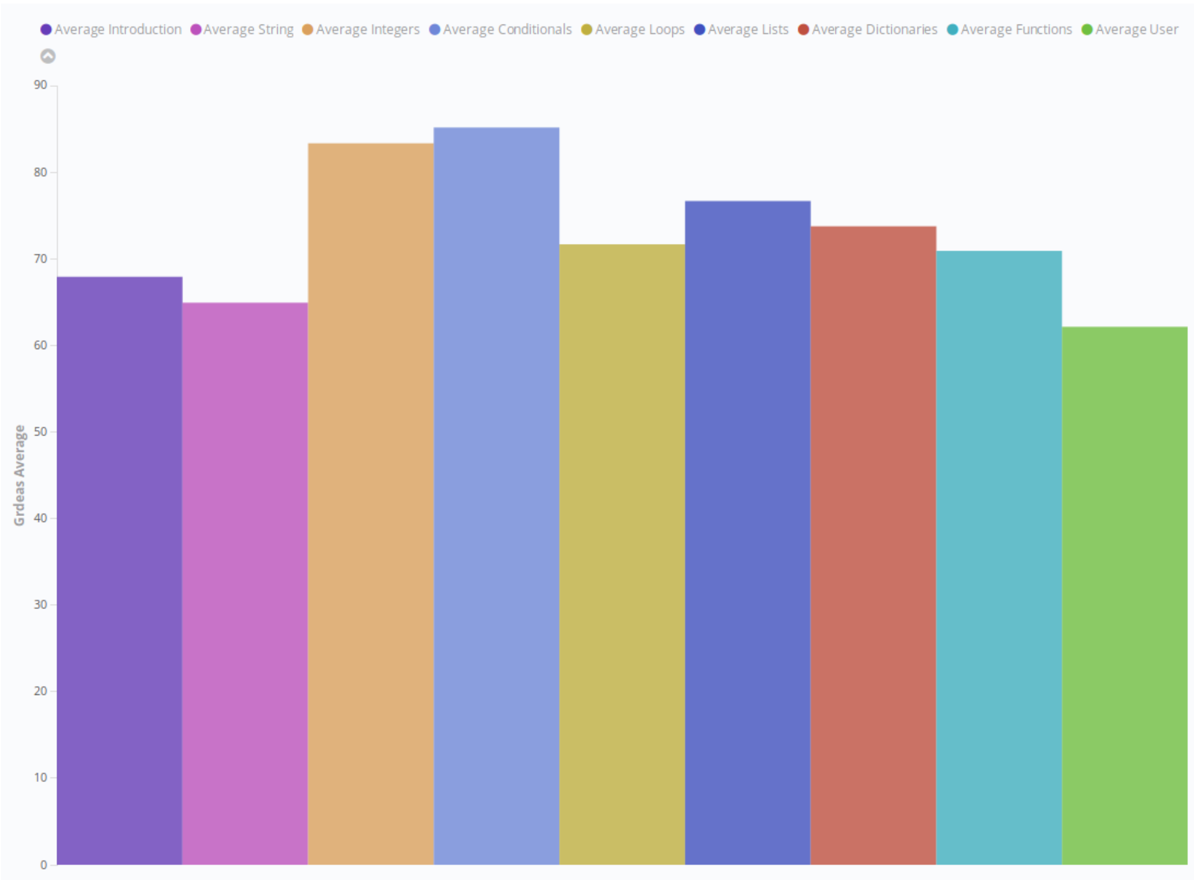

Figure 5.8: Kibana Average Grades per Topic Visualization

These visualizations are grouped into three different dashboards: two for teachers and one for students. The first one shows general data of the course, such as the grades and characteristics of the students enrolled. The second shows visualizations that correspond to the emotional evolution of the students as well as the attention they have paid to the lessons. Finally, the student dashboard includes both visualizations about their emotions, as well as the attention and grades they have obtained in each topic.

5.6 Semantic Task Automation Platform

The Semantic Task Automation platform is one of the main developments of our project, due to the improvement that the adaptation of the students' workspace in a smart way can suppose for their academic performance. This platform, called Ewetasker, has as its starting point the work of Sergio Muñoz López in 2016 [41]. In recent years, the platform has been used, modified and improved in numerous master thesis and papers [43] [42] [60] [40]. It has also been used in the scope of research projects such as Emospaces [1]. For this reason, it has been necessary to develop a new version that brings together the different features added over the years, improves the ease of use and performance, and fixes some bugs found during its intensive use.

In the next sections, the purpose and operation of the Ewetasker platform will be detailed, as well as the modules that integrate it, the technology they implement, and their function. Finally, it will be explained how its integration has been carried out within the scope of this project.

5.6.1 Ewetasker

Ewetasker is an automation platform based on semantic ECA (Event-Condition-Action) rules. It is capable of enable semantic automation rules in a smart environment allowing users to conFigure his own automation rules in an easy way. To this purpose, plugins have been developed semantically as channels to make possible integrations with services such as Twitter or Gmail and with devices such as smart lights, beacons or smartphones. These channels use EWE ontology to define their features such as events, actions, and their respective parameters.

Figure 5.9 shows the platform architecture. At a first glance, it can be seen how the server is the central part of the application, through which the rest of the modules are connected. It also shows the web application through which users access the platform, as well as the rules engine, the different storage solutions, and the devices and services that interact with the system. In the next sections we describe the operation of each module of this architecture.

[1] https://itea3.org/project/emospaces.html

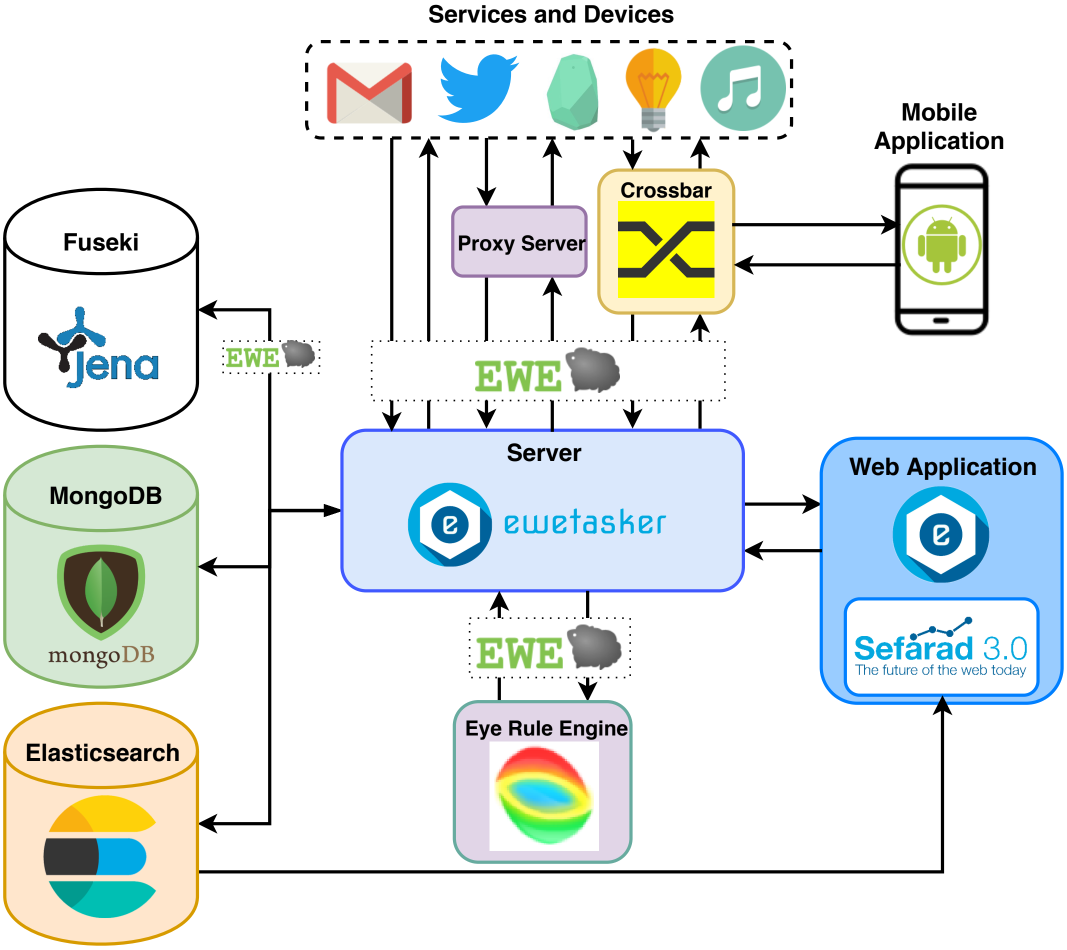

Figure 5.9: Ewetasker Architecture

5.6.1.1 Web Application

The web application connects users to the platform. It implements a web interface developed in React that allows users to create automation rules in a simple and intuitive way. To learn more about the possibilities offered by this application, the views it provides are detailed:

- **Channels View.** Shown in Figure 5.10. Provides a list of available channels as well as the events and actions they define.

- **Devices/Services View.** Provides a tool for defining and storing device and service parameters for ease of later use in rule creation..

- **Rules View.** Provides a list of created rules and the user they belong to, and a rule editor, based on drag and drop icons and actions.

- **Dashboard View.** Provides a dashboard for administrators in which usage statistics visualizations are shown. The tool used for the implementation of these visualizations is Sefarad, which is based on Polymer Web Components.

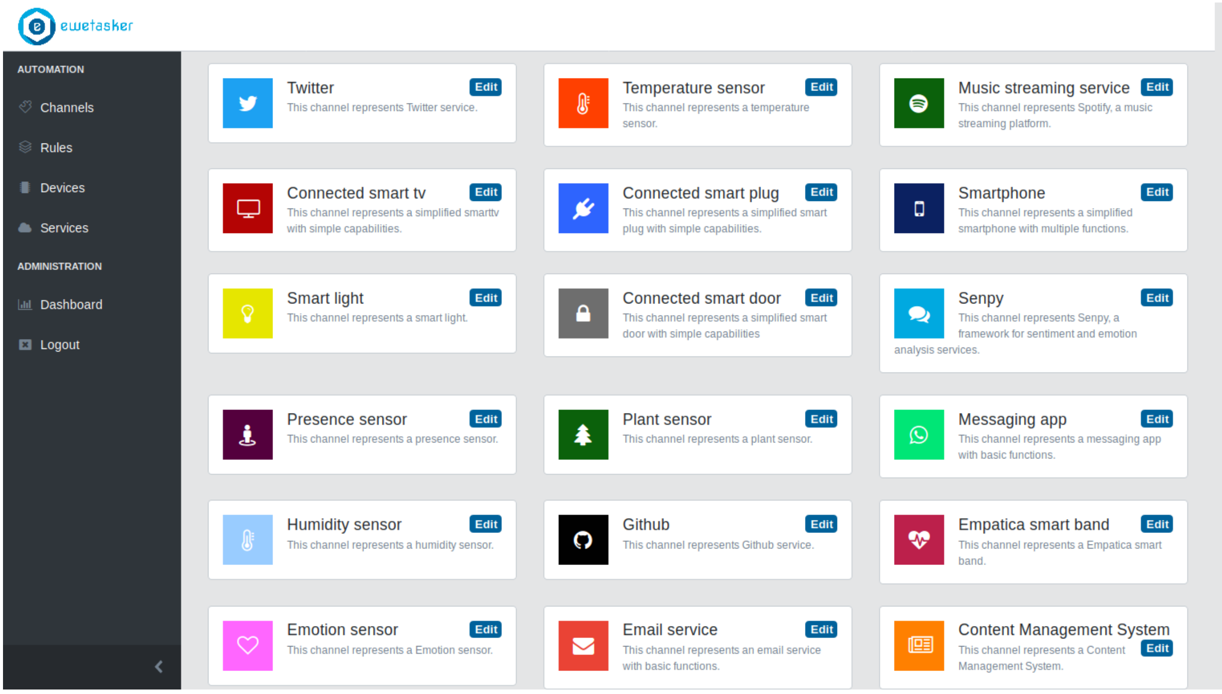

Figure 5.10: Ewetasker Web Application Interface

5.6.1.2 Server

The server is the main part of the platform. It implements a Flask API in Python that the rest of the modules use to make requests and communicate with each other. These requests hide behind a logic and a set of functions that allow server the correct integration with the rest of modules.

First, it defines three models to interact with each database: Elasticsearch, Fuseki and MongoDB. Secondly, it implements a sub-module in charge of evaluating the events it receives. To do this, it forwards the events it receives, along with the rules that are stored, to the rules engine. Finally, it receives the result of the evaluation carried out by the rule engine and performs the corresponding actions. For this purpose, it implements a sub-module called Actuator, which is in charge of sending the actions to be performed to the different devices or services.

5.6.1.3 Storage

The storage system implements three different solutions depending on the function of the data it stores. The purpose of each database is briefly summarized below.

- **Fuseki.** Stores the definition of channels, rules and users semantically via N3 using EWE ontology. This data is accessible through the SPARQL queries defined in the corresponding model implemented in the server.
- **MongoDB.** Stores the usernames and passwords of the automation platform as well as tokens and personal keys needed to use services such as Twitter or Gmail.
- **Elasticsearch.** Stores the platform usage data, so that they can be accessible for Sefarad visualization.

5.6.1.4 Eye Rule Engine

The Eye Rule Engine is the module that lies the automation platform potential. It implements an EYE server, presented in Section 2.7.3. This server provides an Euler Yap Engine reasoner that executes the inferences of the ontological model. This reasoner receives the events and rules, and after processing them, returns the actions to be performed.

5.6.1.5 Proxy Server & Crossbar Plugin

The Proxy Server and Crossbar Plugin are two modules that enhance the functionalities and capabilities of Ewetasker. However, due to their different nature they are explained separately for a better understanding of their purpose.

Proxy Server. The main objective of this module is to connect to channels in a smart environment outside the network where Ewetasker is deployed. In this way, it is possible to send and receive information from intelligent devices located in other local networks to generate events and carry out actions in which they are involved. Nowadays, this architecture is widely used in the implementation of controllers for devices such as smart lights, smart thermostats or smart plugs.

Within the scope of this project, the Proxy Server has been adapted to the channel definition implemented by the new version of the Ewetasker platform. Furthermore, an improvement has been added to the sub-module in charge of executing actions in smart lights of the Philips Hue [2] type. This improvement gives users the possibility to change the colour of the bulbs, adapting the RGB colour values, defined in the automation rules, to the CIE RGB colour space [64] used by Philips.

Crossbar Plugin. This module, developed by Pablo Viñals Lage [34] and based on the Crossbar.io platform [12], has three main functions. First, it performs a protocol conversion, specifically, it converts WAMP [3] and MQTT [4] protocol messages to the HTTP protocol used by Ewetasker. This notoriously broadens the range of devices that can be connected to the automation platform both to send events and to receive actions. Secondly, it converts the messages of these devices from JSON format to semantic format, for its correct functioning with the rule engine. Finally, it is able to listen to a greater number of events simultaneously, which increases the number of devices and services connected at the same time.

In this project, the Crossbar Plugin has been adapted to the new version of Ewetasker. To this end, the semantic converter has been adapted as well as its connection to the platform. In addition, it has been necessary to implement a new event verifier for the channel that integrates the automation platform with the EAES.

5.6.1.6 Mobile Application

This mobile application, based on the work of Antonio Férnandez Llamas [22], is developed in Android, and it aims to be a source of events and to provide several actuators for the automation platform.

In this project, this application has been adapted to the new version of Ewetasker. To do this, a channel called *"Smartphone"* has been created, which encompasses all the events and actions available on the phone. The format of events and actions has also been modified to be understandable by the Crossbar Plugin, because the mobile application is one of the devices that generates more events per second. In addition, it will be used in the study cases through its integration with the Empatica wristband.

[2] https://www2.meethue.com/en-us
[3] https://wamp-proto.org/
[4] http://mqtt.org/

5.6.2 Emotion Aware E-learning System Integration

As mentioned in Section 1, Ewetasker's integration with EAES aims to improve students' mood and increase their comfort in the workspace in which they are carrying out the academic activities. In order to achieve this, it has been necessary to implement semantically a channel that specifies the events that can happen in the learning platform as well as the parameters that define them. These events are generated based on user interactions or through changes and notifications in the Moodle course.

A summary version of the channel definition for EAES following the EWE specification is shown in List 5.4.

Listing 5.4: Emotion Aware E-learning System channel definition.

```
# Channel definition
ewe:EmotionAwareElearningSystem a owl:Class ;
    rdfs:label "Emotion Aware E-learning System" ;
    rdfs:comment "This channel represents Emotion Aware E-learning
        System for Moodle platform." ;
    ewe:generatesEvent ewe:EmotionDetected ;
    ewe:hasCategory ewe-service:Service ;
    rdfs:subClassOf ewe:Channel .
# Events definition
ewe:EmotionDetected a owl:Class ;
    rdfs:label "Student emotion detected by Emotion Recognition Module"
         ;
    ewe:hasInputParameter ewe:Emotion ;
    ewe:hasOutputParameter ewe:EmotionIntensity ;
    rdfs:subClassOf ewe:Event ;
    rdfs:domain ewe:EmotionAwareElearningSystem .
# Parameters definition
ewe:Emotion a owl:Class ;
    rdfs:label "Emotion" ;
    rdfs:comment "This parameter represents the specific emotion
        detected." ;
    rdf:datatype xsd:string ;
    rdfs:subClassOf ewe:Parameter ;
    rdfs:subClassOf onyx:Emotion ;
    ewe:operation string:equalIgnoringCase ;
    rdfs:domain ewe:EmotionAwareElearningSystem .
```

CHAPTER 6

Case study

In this chapter we are going to describe the process we have followed to implement the study cases of Academic and Emotional Dashboards for teachers, Emotional Dashboard for students, Emotional Learning Analysis, and Smart Automation for students and teachers. For this purpose we will go over them to show the main features.

6.1 Introduction

Once the technologies, motivations, and architecture of the project are known, it is necessary to explain how the system works in a real environment in which both teachers and students interact with it. The context in which these interactions take place is an online programming course. This course, entitled *"Basic Python for Beginners"* is available on the Moodle.net [1] platform and was created by Christian Thompson [2]. This course has been chosen mainly because it adapts easily to the use cases that we will see later. More specifically, it offers a set of video lessons and topic-based questionnaires, which will be used together with the Videogazer and EmoTestCapturer modules respectively. In this way, we have a base on which to deploy our Emotion Aware E-learning System.

In order to evaluate the usefulness and soundness of the system, four case studies have been defined: Academic and Emotional Dashboards for teachers, Emotional Dashboard for students, Emotional Learning Analysis, and Smart Automation for students and teachers. In them, the main features of the system have been tested: recognition of emotions and attention; data collection, processing, and storage; data visualization; data analysis based on Machine Learning techniques; and automation of a smart environment.

It should be remembered that the main goal of this project, and consequently of its case studies, is to improve the mood and comfort of students who use an e-learning platform to broaden their knowledge and skills. In order to do this, first of all, both teachers and students will be actively involved in the knowledge of the emotions they experience during the course through the use of dashboards. In this way, adaptations of academic activities can be made as well as motivating students to a change of attitude that allows them to improve their academic performance. On the second place, there will be a more exhaustive analysis of the data collected by the system through the application of Machine Learning techniques, which will show relationships between the data impossible to see directly through the dashboards but with the same or greater usefulness. Lastly, the capabilities of the modules developed for the recognition of emotions and attention will be used together with the task automation platform to improve the comfort of students while they are studying.

This chapter is going to detail the operation of each proposed case study, and is going to summarize the conclusions drawn from its implementation.

[1] https://new.moodle.net/
[2] http://www.christianthompson.com/

6.2 Academic and Emotional Dashboards for teachers

This use case is meant to remove communication barriers between teachers and students in distance learning environments. To this end, as already mentioned, an online course has been created for Python learning on the Moodle e-learning platform. This course has served us as a virtual scenario in which students and teachers interact through the proposed academic activities. In order to better understand the role played by each actor in this case study, its functions are detailed below:

- ***Student.*** Its goal is to make the most of the lessons that make up the course. For this purpose, the student will carry out three types of activities: theory study, video lessons, and questionnaires. In these last two activities the student's attention, stress, and mood will be monitored. In addition self-reported emotions tests will be performed during the course. The collected data will be shown to the teacher in a dashboard that will help him to better understand how the student's learning experience is and to make adaptations to improve his academic results.
- ***Teacher.*** The objective of the teacher is to know, as in face-to-face courses, how the student's learning experience is being during the course. In this way, he can make adaptations in the methodology, content, or activities that improve the student mood, and consequently his academic results.

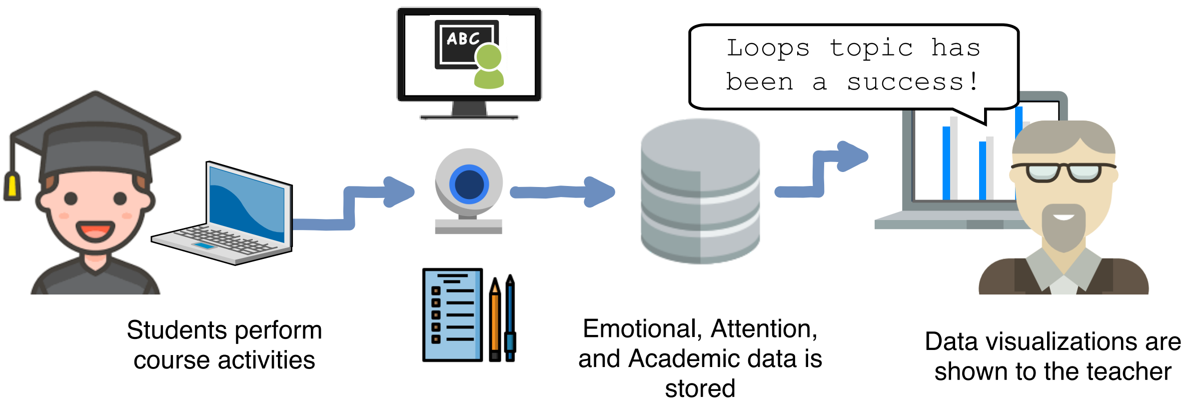

Figure 6.1: Teacher's Case Study

For the design of this case of study, we have thought the actions that can be developed between the moment that students log in for the first time in the course, until the teacher looks the results in the developed dashboard, once the course is finished. To get an idea of the main actions that take place in this case study we have Figure 6.1.

More in detail, the actions start when the student accesses to the course for the first time. At that time, the student can see the classic Moodle course overview along with the widget designed in this project. This widget will ask the student to register in the Emotion Aware E-learning System. To do this, you will need to enter personal data such as age, gender, nationality and identifier number, as shown in Figure 6.2. This data will be sent to the system that will create an index in Elasticsearch for future data storage. In addition, the user name will be registered together with his identifier in MongoDB in such a way that the system knows that the user is registered for future accesses. To make it easier to register a large number of students at the same time, an option has been implemented (Figure 5.3) in the teacher widget view by uploading a csv file with similar characteristics to the one used to add participants in Moodle.

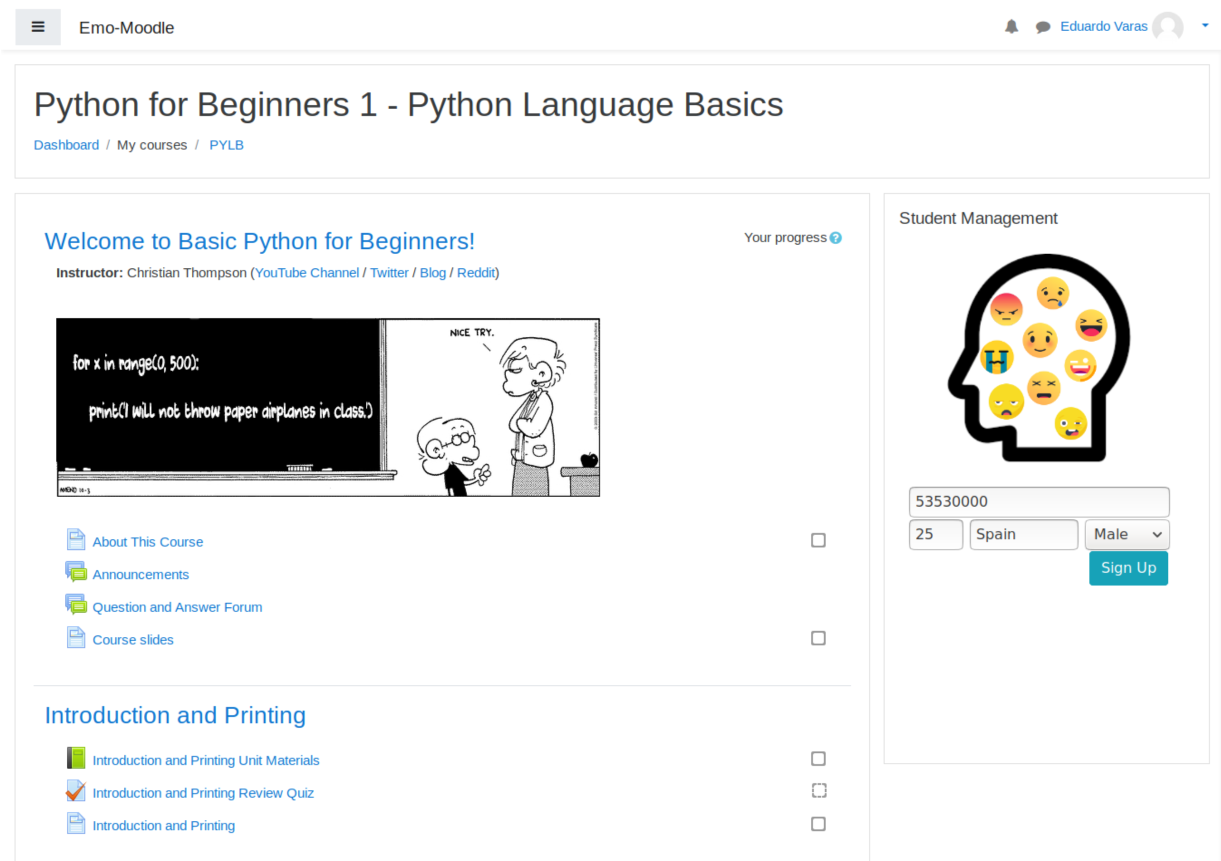

Figure 6.2: Students' Start Course View

Once the student is registered in the system, he can start to carry out the course activities. These activities are divided into topics and include a video lesson and a quiz to evaluate the knowledge. The video lessons, stored in Youtube platform, use the Videogazer module, shown in Figure 5.4, to measure the attention of the students in relation to the topic and to the current video time.

When the concepts presented in the video lessons have been assimilated and studied, students must take a test for each topic. These tests are not available throughout the course, teacher decides when they can be taken. To this end, a data extractor has been implemented in the Emotion Widget module, which allows teachers to store the questionnaires to be carried out, together with the subject and the course to which they belong. The availability of these tests along with the EmoTestCapturer module is activated through the teacher widget view. The availability information of each questionnaire, stored in MongoDB, can also be consulted from the student widget view, as shown in Figure 5.3.

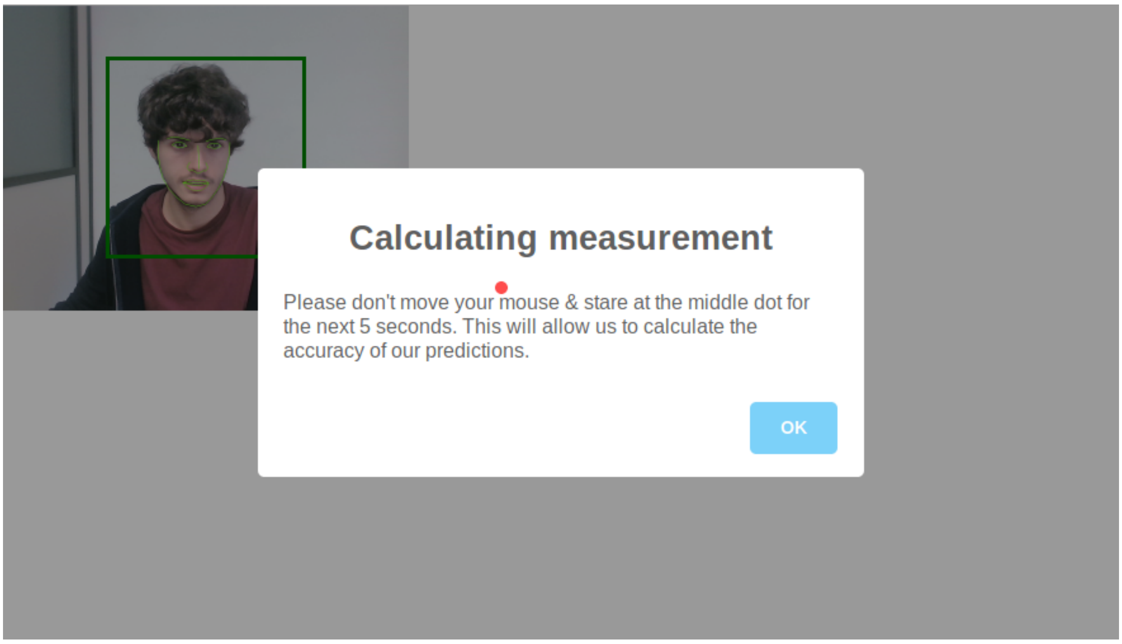

Figure 6.3: Videolesson example empowered by Videogazer module

In these questionnaires, student must answer a series of questions related to the topic studied, while the emotions and stress level experienced are captured by the EmoTestCapturer module and the Empatica smartband respectively. The operation of the EmoTestCapturer can be seen in Figure 6.4.

At the same time that the activities of the course are being carried out, the student can register the emotions that he experiences in his daily life through the Self-reporter module, shown previously in Figure 5.5. In the same way, teacher can obtain the grades that the students have obtained at any time of the course through the Moodle platform. This platform allows teachers to download the results in csv files. On the other hand, the teacher widget view allows teacher to upload the results to the system, so that they can be processed and stored in Elasticsearch, and later visualized.

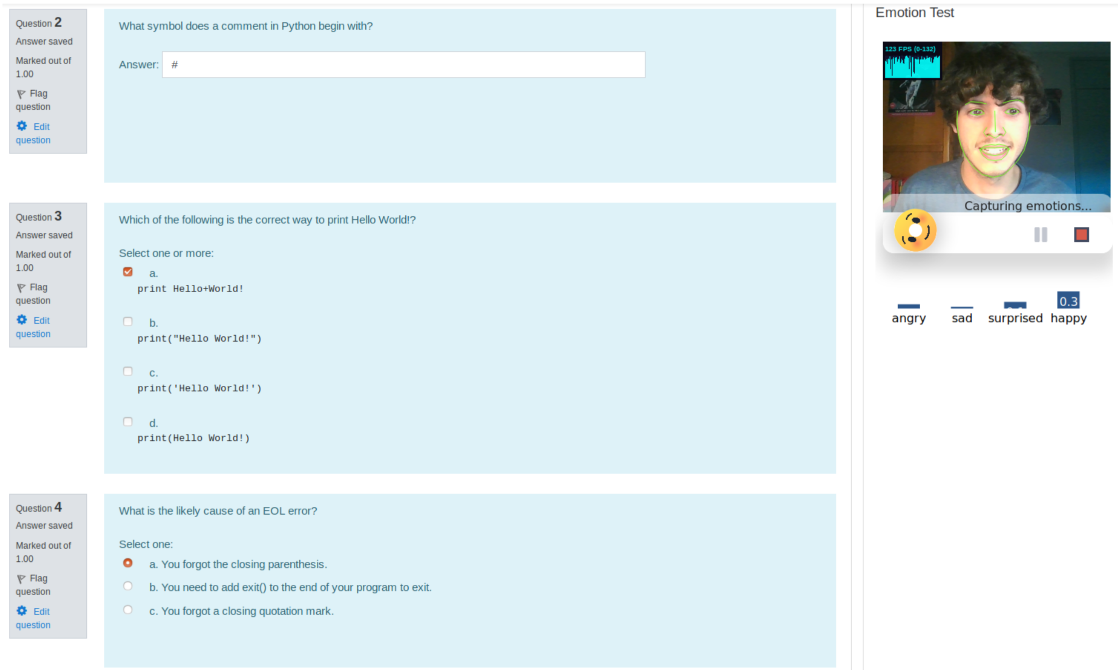

Figure 6.4: Questionnaire example empowered by EmoTestCapturer module

Finally, the teacher can access the visualizations of the data obtained through all these processes. These visualizations are divided into two dashboards: one more focused on the academic results and personal information of the group of students enrolled in the course, and another more oriented to show the emotions that students have experienced in relation to the academic activities carried out. These two dashboards are shown and detailed below.

6.2.1 Teacher's Academic Dashboard

The Teacher's Academic Dashboard is designed to show the personal and academic information of students in a fast, simple and intuitive way. To do so, it is based mainly on personal information provided by students through the Moodle platform and the Emotion Widget. The dashboard overview is shown in Figure 6.5.

The information shown includes personal features such as sex, age, nationality, as well as academic data, such as the grades obtained in each questionnaire of the course. It also includes global course data, such as the number of students enrolled or the number of students who have successfully completed the course. In order to adapt the displays to the data they show and to make them more understandable, different types of visualizations have been selected from the wide range offered by Kibana. The most outstanding visualisations are listed and detailed below.

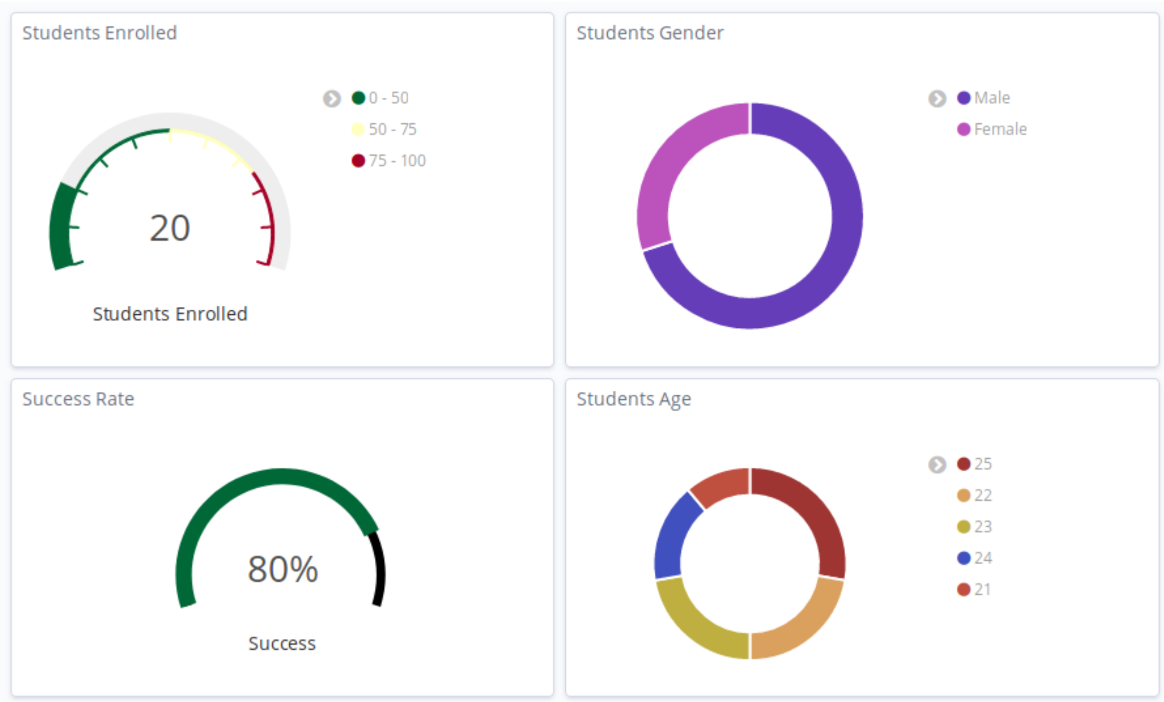

Figure 6.5: Teacher Academic Dashboard overview

Figure 6.6 shows the statistics obtained from students' grades. First, the average grade for each topic is shown. Second, the highest grade per topic is shown, and finally, the lowest grade per topic is shown.

Average Grades Table

Introduction	String	Integers	Conditionals	Loops	Lists	Dictionaries	Functions	User	Total
67.917	64.926	83.334	85.157	71.667	76.667	73.75	70.91	62.143	72.94

Max Grades Table

Introduction	String	Integers	Conditionals	Loops	Lists	Dictionaries	Functions	User	Total
100	100	100	100	100	100	100	100	100	97.4

MIn Grades Table

Introduction	String	Integers	Conditionals	Loops	Lists	Dictionaries	Functions	User	Total
19.44	9.09	33.33	50	44.44	0	25	18.18	14.29	39.03

Figure 6.6: Grades Charts for teachers

Finally, Figure 6.7 shows the percentage of students whose final grade is within predefined ranges that correspond to different grade levels.

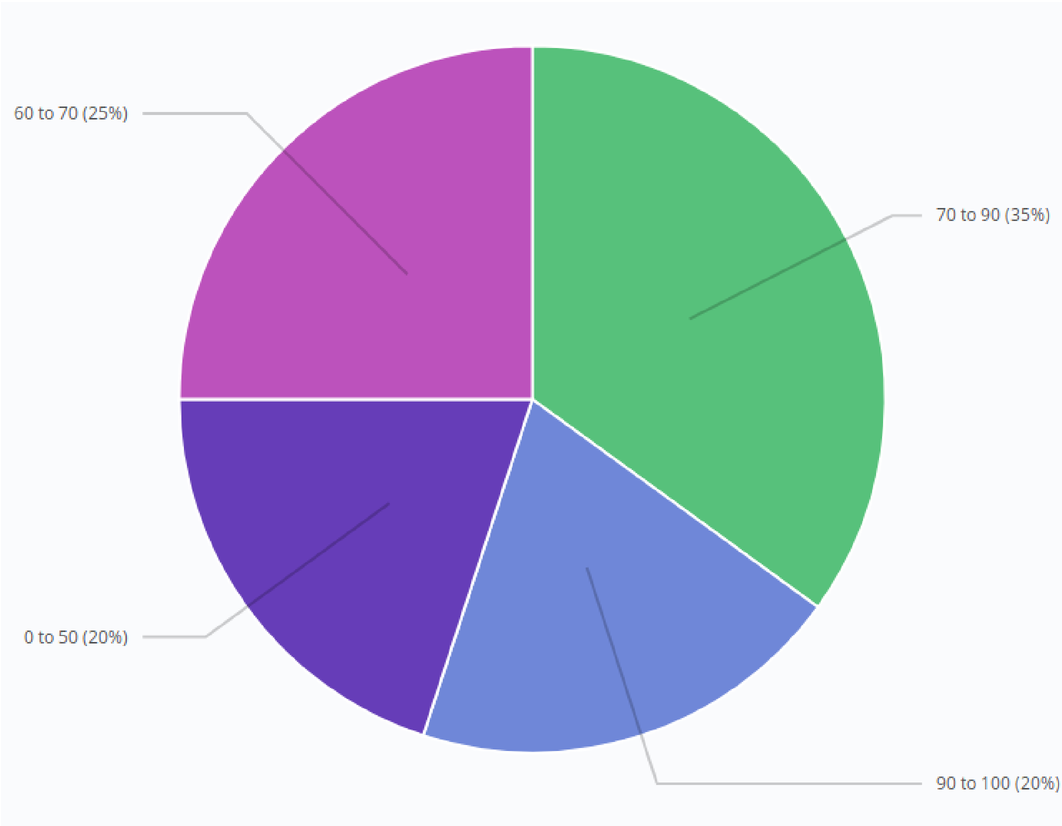

Figure 6.7: Students' Grades Range Pie Chart per frequency

6.2.2 Teacher's Emotional Dashboard

The Teacher's Emotional Dashboard is designed to show information about the emotions that students experience during the different activities of the course. This information is correlated with the grades obtained on each topic in such a way as to show how each emotion affects the learning outcomes obtained. In the same way, the data of the attention paid to the video lessons and their influence on the grades are shown, as well as a histogram of the emotions reported by the students throughout the course. The most outstanding visualisations are listed and detailed below.

The visualization shown in Figure 6.8, allows the teacher to know the average of attention that the students have paid to the video lesson of the topic of Introduction. It can be seen how the attention in the first seconds of the video is high, while as time goes by it varies until it drops sharply at the end.

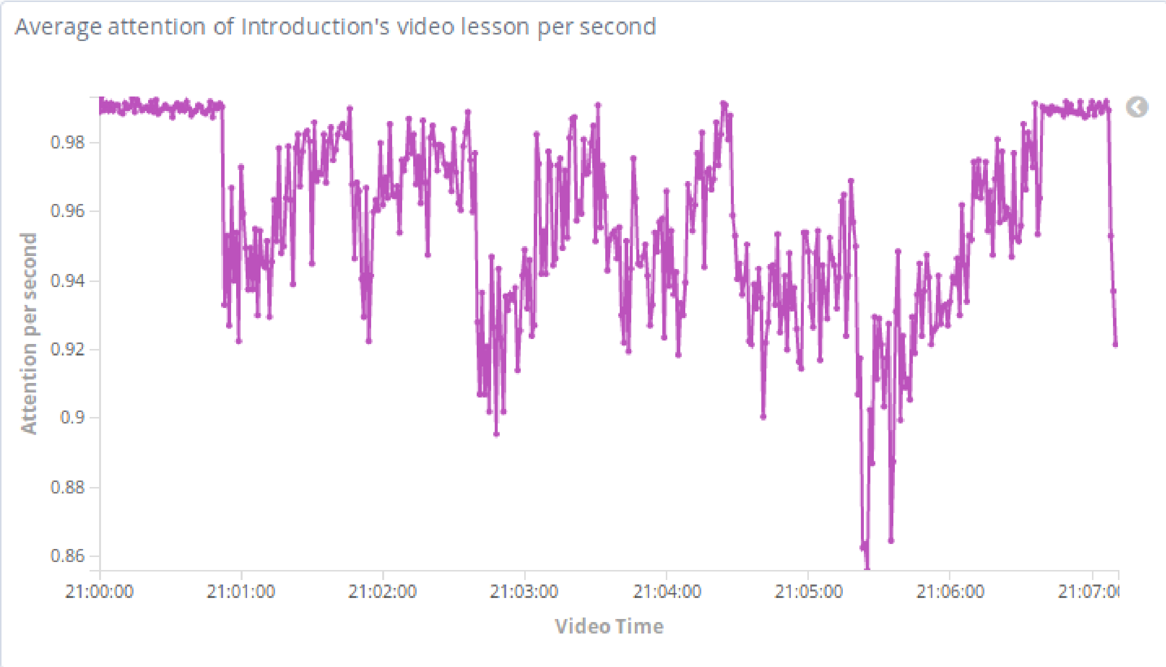

Figure 6.8: Example of a student's average attention of a video lesson per second

Figure 6.9 shows the average of each of the emotions that the students have experienced while taking the different questionnaires available in the course.

Lastly, Figure 6.10 shows a *"tag cloud"* visualization, which allows the teacher to quickly update the visualizations of this dashboard to show only the data corresponding to the student with the selected name.

Through these visualizations the teacher can better understand the characteristics of the group of students who are participating in the course. This allows him to adapt the contents to the type of students who have greater difficulties, taking into account a great variety of factors such as emotion, attention, age or nationality, in such a way that the communication barriers of an e-learning environment are overcome and the students successfully pass the course. In the same way, these visualizations are interactive and allow the teacher to filter the data by the user to whom they belong thanks to the Name Tag Cloud. In this way teacher can compare the performance of a student with respect to the group, as well as know all its information immediately.

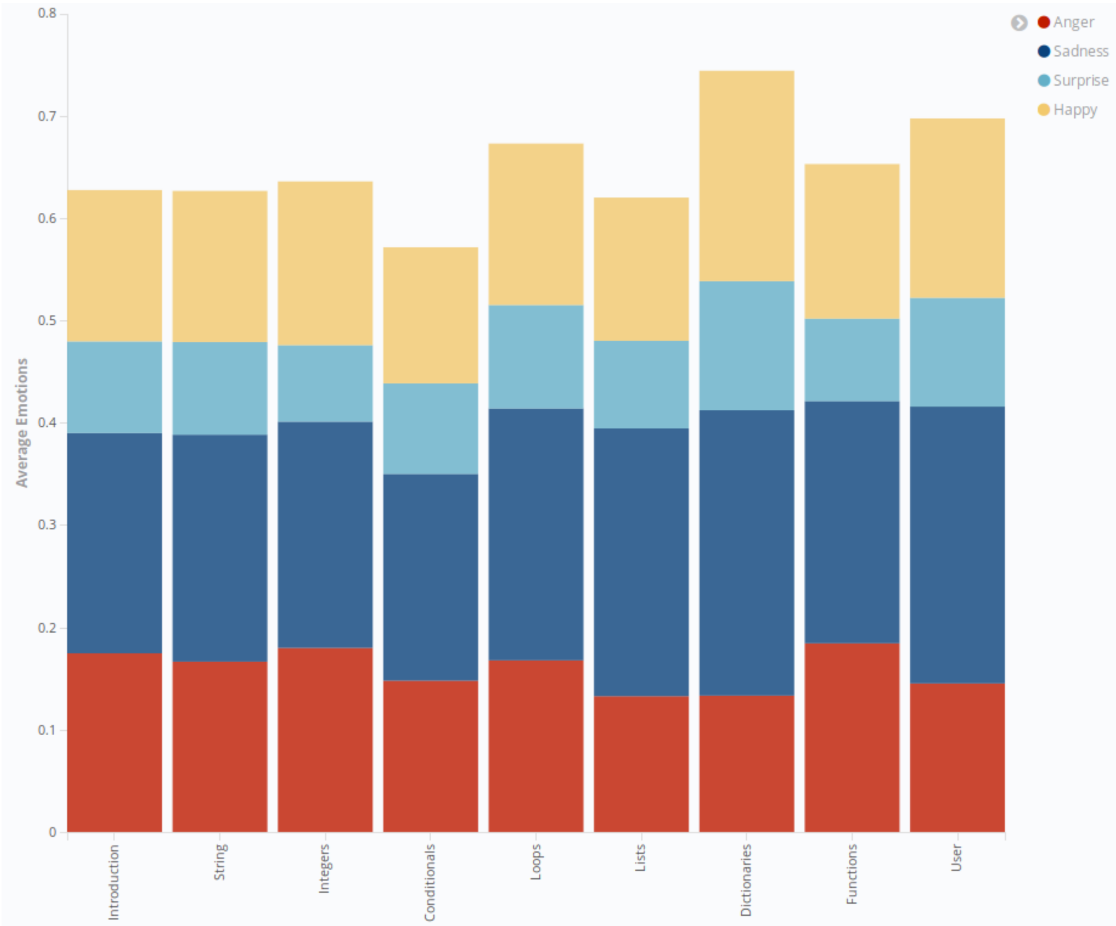

Figure 6.9: Average of emotions captured per test

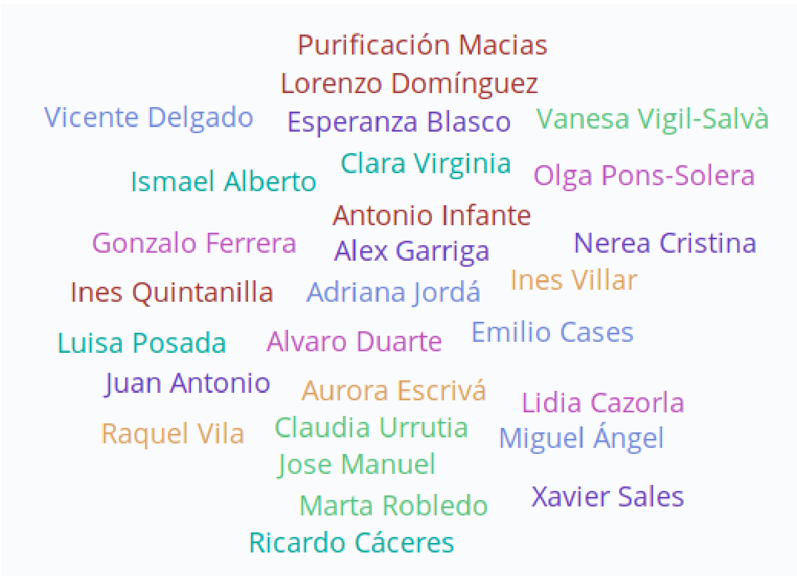

Figure 6.10: Students' Name Tag Cloud

6.3 Emotional Dashboard for students

This case study includes much of the interactions that occurred in the previous one, so we will focus primarily on the visualization part. However, the most important actions that take place in this study case are summarized in Figure 6.11.

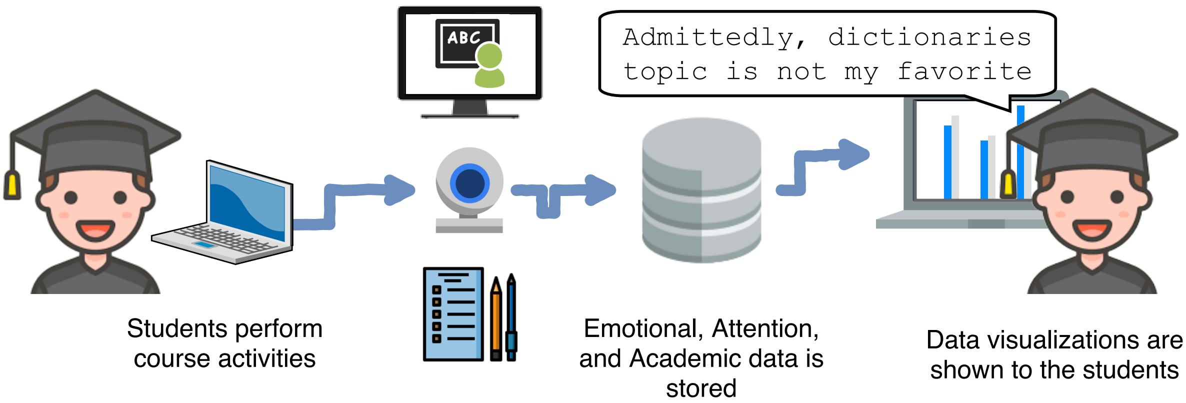

Figure 6.11: Student's Case Study

The Student's Emotional Dashboard is designed to show in a simple way, i.e. without any previous knowledge about visualizations, the evolution of the student's emotional state along with the grades obtained during the course. To this end, a series of visualizations similar to the previous ones have been implemented. The most relevant are shown below.

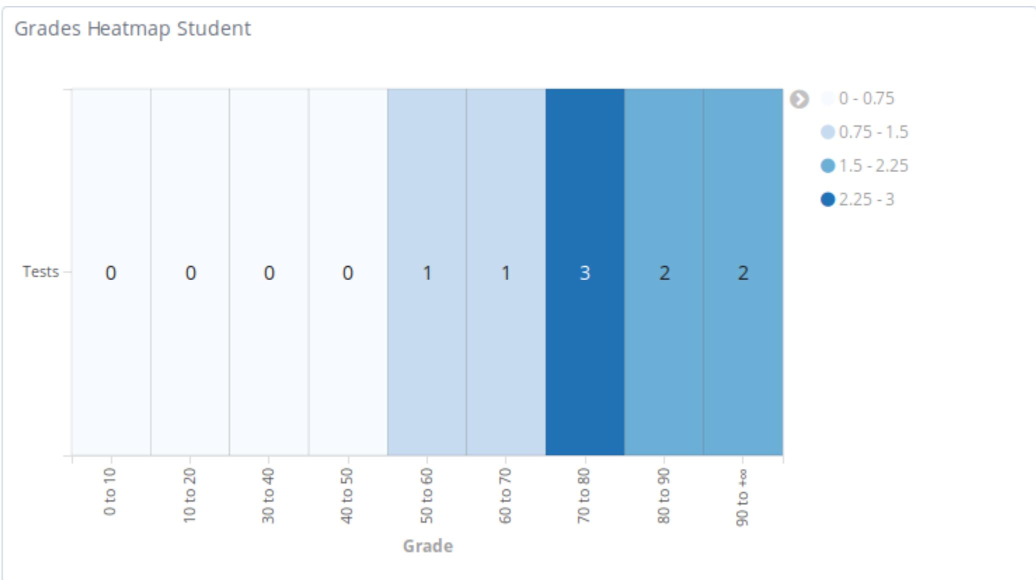

Figure 6.12: Student's Heatmap of Grades per frequency

Figure 6.12 consists on a heat map visualization showing the number of times the student has scored within established ranges. This allows the student to know the level of achievement that he is having in the course.

Secondly, Figure 6.13 shows the evolution of the emotions that the student has experienced during the time that he has been carrying out a questionnaire. The color code, used in this visualization and in others in which emotions appear, is inspired in D Borth et al. work [7], and it is the following one: anger, red; sadness, dark blue; surprise, light blue; and happiness, yellow. In this example it can be seen how the emotion that the student experiences with greater intensity is sadness, followed by surprise. It can also be seen how in the middle of the test the student changes his mood for a few seconds in which the intensity of happiness increases drastically.

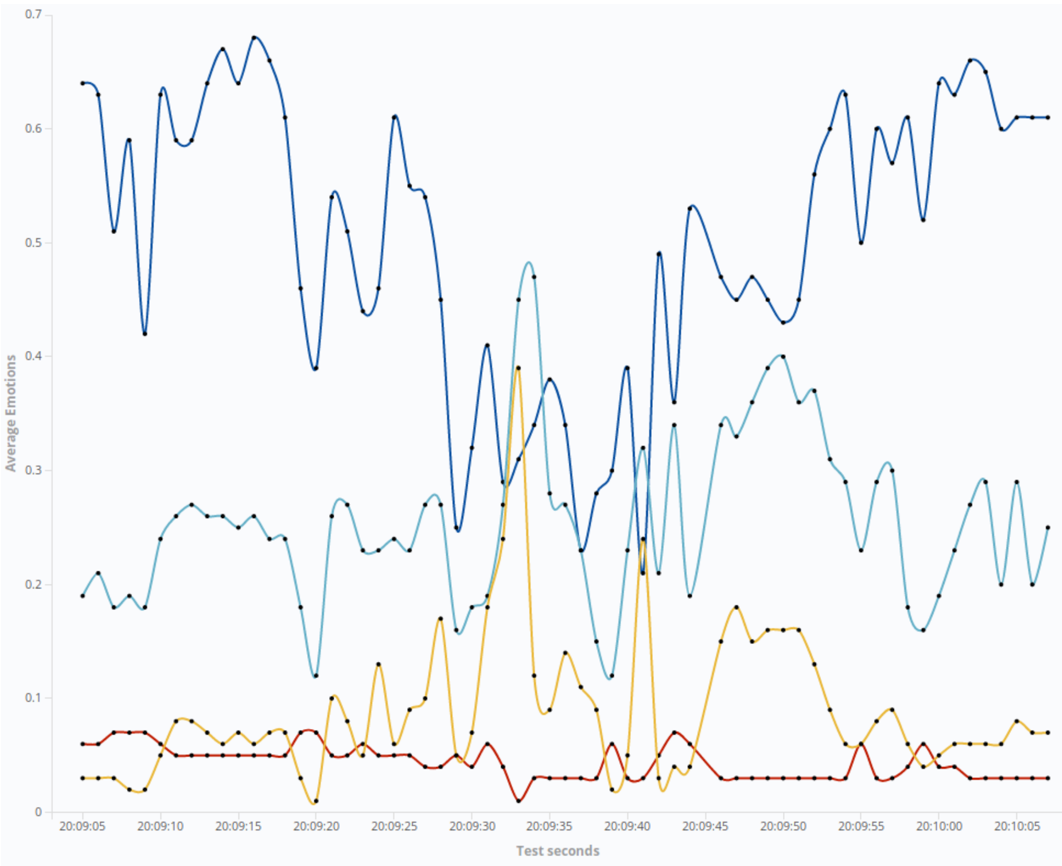

Figure 6.13: Example of the average captured emotions in a test per second

Finally, Figure 6.14 shows how the feelings expressed by the student have evolved during the weeks of the course. In this example it can be seen how the emotion that is most intensely reported is happiness. In the same way we can also see weeks of the course in which this emotion reaches a higher level than the one recorded when the course started, at which time the students tend to be happier. It can also be seen how surprise is a frequently reported emotion and how negative emotions have small values in comparison.

This dashboard allows students to perform an introspection exercise, in which they can learn how emotions affect their performance, and consequently, look for solutions to make the most of the course.

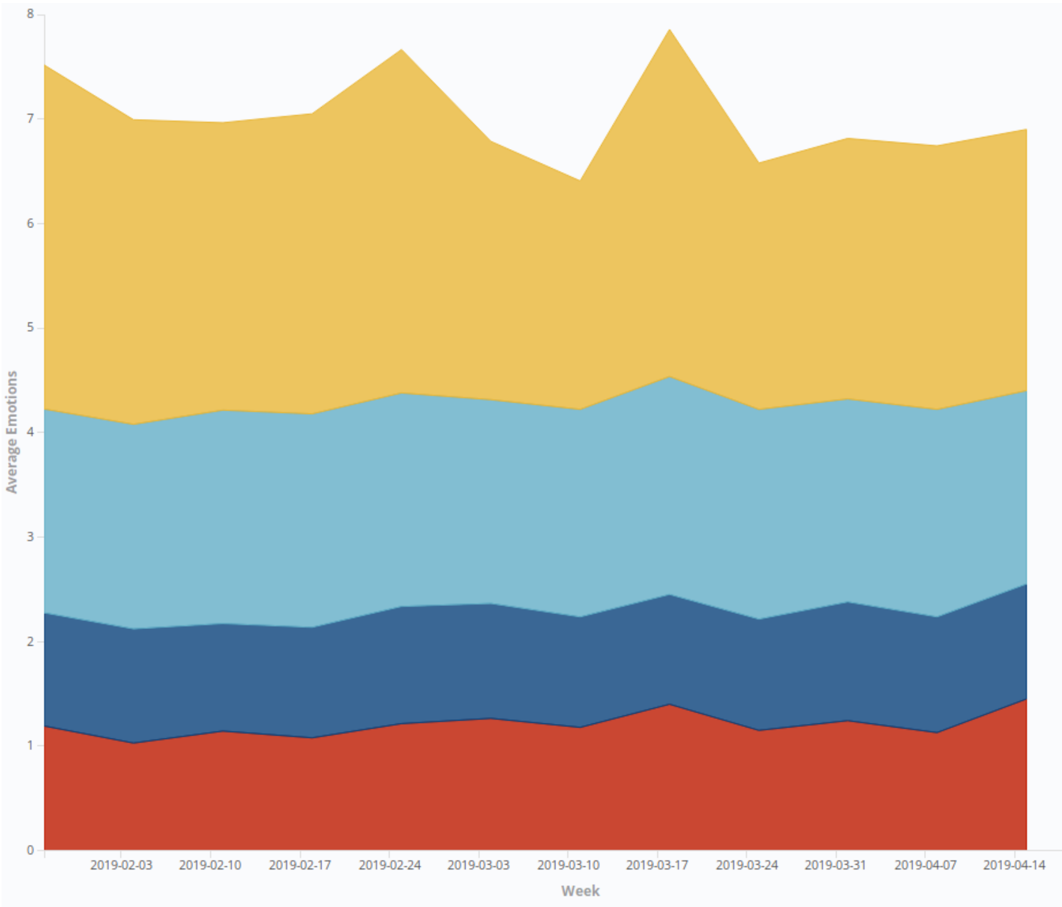

Figure 6.14: Emotions self-reported by students shown per week

6.4 Emotional Learning Analysis

This case study is based on the use of Machine Learning techniques to analyse the data stored in the Emotion Aware E-learning System (EAES) in order to extract new and valuable information that cannot be obtained through ordinary study and visualisation techniques. These techniques will allow the teacher to know in detail how factors such as sex, age or emotions influence the academic performance of students as well as to obtain a classification of them from emotions using the K-Means algorithm.

For the development of this case study, the Python course presented previously has been used as the scenario, and a total of thirty students of the Escuela Técnica Superior de Ingenieros de Telecomunicación of the Universidad Politécnica de Madrid have participated. The data collected correspond only to the emotions experienced during the completion of the questionnaires and the grades obtained by the students, due to the temporary impossibility of using all the capabilities offered by the system developed in this project.

In order to better understand the role played by each actor in this case study, its functions are detailed below:

- ***Student.*** Its goal is to get the highest grade in the course. To do this, the student will make the questionnaires for each topic while the system will measure the emotions he experiences as well as the time spent on each test.
- ***Teacher.*** The main objective of the teacher is to know the influence that certain emotions have on the grades obtained, in order to be able to adapt the methodology of the course, and consequently improve the academic performance of the students. With this purpose in mind, the teacher will use the EAES to obtain a report with the results of this analysis.

This case study begins once all the questionnaires have been completed by the students. Then, the teacher, through the interface provided by the Emotion Widget, orders the EAES to perform the analysis tasks. This analysis results in a set of charts and tables. We proceed to explain and detail the main ones.

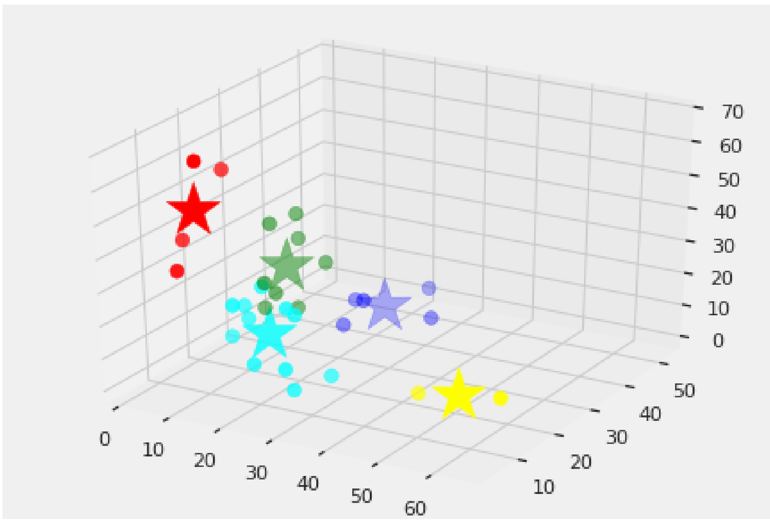

Figure 6.15: K-Means Clustering 3D Classification

The first, as it could not be otherwise, is based on the data of the emotions that the students have experienced during the questionnaires. To do this, it was necessary to create four new features (anger, sadness, surprise, and happiness). These features represent the average of the emotions experienced by each student and will also serve as a basis for other visualizations.

The visualization shown in Figure 6.15 is the result of applying the K-Mean classification algorithm to our dataset. In this analysis we have used as dimensions, three of the previously calculated ones, specifically, the three that have appeared with greater intensity (anger, sadness, and happiness), and as a study variable, the course final grade. As it is a dataset with few samples, the value of k, which defines the number of groups into which samples can be divided, is not very high. This value was calculated from the elbow curve graph, commonly used in this algorithm, and whose optimal result was k=5. The result of this analysis shows how students who have passed through the course can be divided into five large groups according to the emotions experienced and the score obtained. However, this graph does not show the relationship between each emotion and the feature under study.

To be able to observe this relationship, it will be necessary to display a two-dimensional chart with the projections from our 3D graph as shown in Figure 6.16. In this graph we can see how most students experience anger with a low intensity, and how this intensity is not directly related to the results obtained. In spite of this, it is possible to appreciate trends such as that students with an anger intensity between 10 and 30 obtain worse results than students with a lower intensity. It can also be seen the existence of two outliers with levels of anger intensity above the average, but without a direct relationship with the performance.

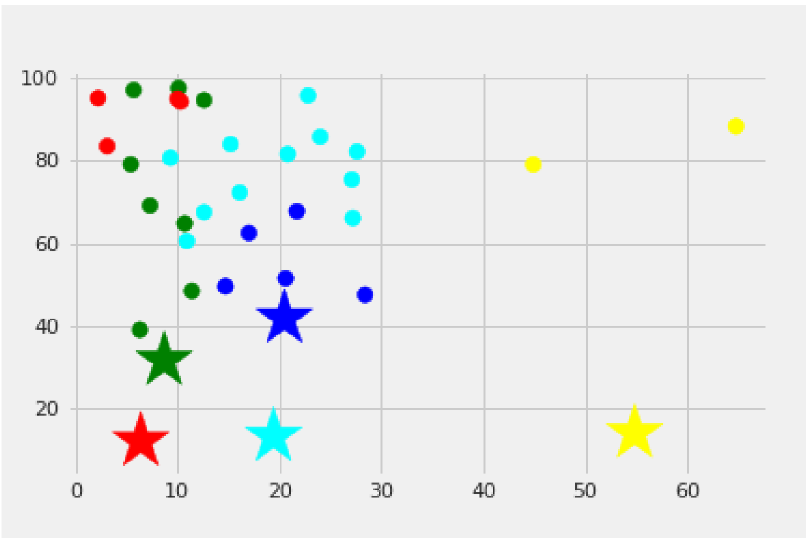

Figure 6.16: K-Means Clustering Classification per Anger intensity

Figure 6.17 shows a table that contains ten features with the highest correlation with the final grade. Logically, the result of this study shows that the characteristics with the highest correlation are the test scores. However, some tests seem to be more important than others, as opposed to the fact that all tests weigh the same to get the final score. This table can help the teacher to know which topics act as indicators of success as well as to know which ones do not challenge the students because they get similar scores. It is possible to see how the tests that act as success indicators are the *"User"* and the *"Loops"* ones.

	Final Grade
Final Grade	1.000000
User	0.903047
Loops	0.893988
Functions	0.813569
Conditionals	0.802878
String	0.797093
Integers	0.678739
Introduction	0.649629
Dictionaries	0.627619
Loops-duration	0.592389

Figure 6.17: Top ten correlations with the final grade

The visualization shown in Figure 6.18 consists of a table with the questionnaires that have obtained similar results according to both the score and the intensity of each of the emotions. Through it, the teacher can know which tests arouse similar emotions among the students and which tests share the same level of difficulty. Similarly, the teacher is capable of knowing which questionnaires have taken a similar amount of time to be completed. This example shows how tests with similar scores are the *"Loops"* and the *"Functions"* ones. It can also be seen, among others, that the *"Introduction"* and *"String"* tests produce a similar intensity for happiness and surprise emotions.

	grade	anger	duration	happiness	sadness	surprise
topic_A	Loops	Lists	User	Introduction	Integers	String
topic_B	Functions	Dictionaries	Lists	String	String	Introduction
value	0.757	0.05	1.696	0	0.05	0.1

Figure 6.18: Tests most similar per feature

Figure 6.19, shows a comparison between the top five students and the bottom five. Through it, the teacher can see how the intensity of each emotion affects the academic performance of students. In more detail, it can be observed how sadness is the emotion that appears with more intensity in each of the groups. However, this emotion most intensely affects students with lower grades. Another fact to highlight is how students with better grades are recognized by the system as happier. Finally, the anger and surprise measurements do not provide relevant results.

In order to know which questionnaire has been the favourite for the students based on the emotions experienced, four visualisations have been generated, as shown in Figure 6.20. These graphs show how the intensity of each emotion has evolved in each of the tests. In this example, it can be clearly seen how the test that has registered the greatest happiness has been the Dictionaries one.

Finally, Figure 6.21 shows a comparison of the results obtained between the students who have spent the most time answering the questionnaires and those who have spent the least. In this graph it can be seen again how sadness is the emotion that detects the system with greater intensity in both groups, so it is not a differential factor. However, emotions such as happiness and anger are dominant in each group respectively. Logically, the results show that students who devote more time to tests score higher.

To conclude, the analysis of the dataset resulting from the interaction of the students with the learning platform allows the teacher to discover new information that can be useful for the improvement of the course. In this case study, the analysis has relied on a small group of students, but if this system is extrapolated to a scenario with a larger number of users and all the capabilities implemented in that project are used, the scope can be much greater.

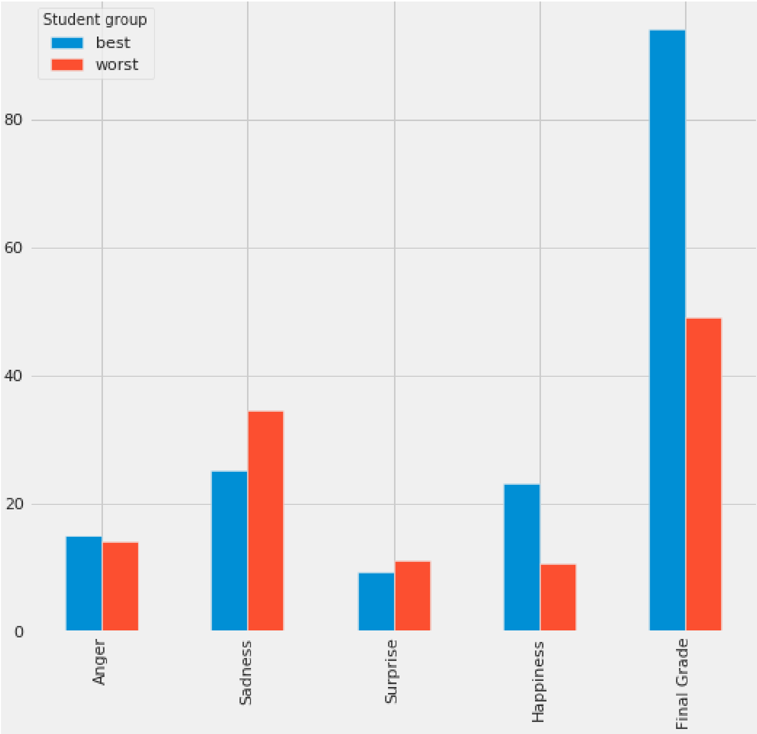

Figure 6.19: Five best vs worst students chart

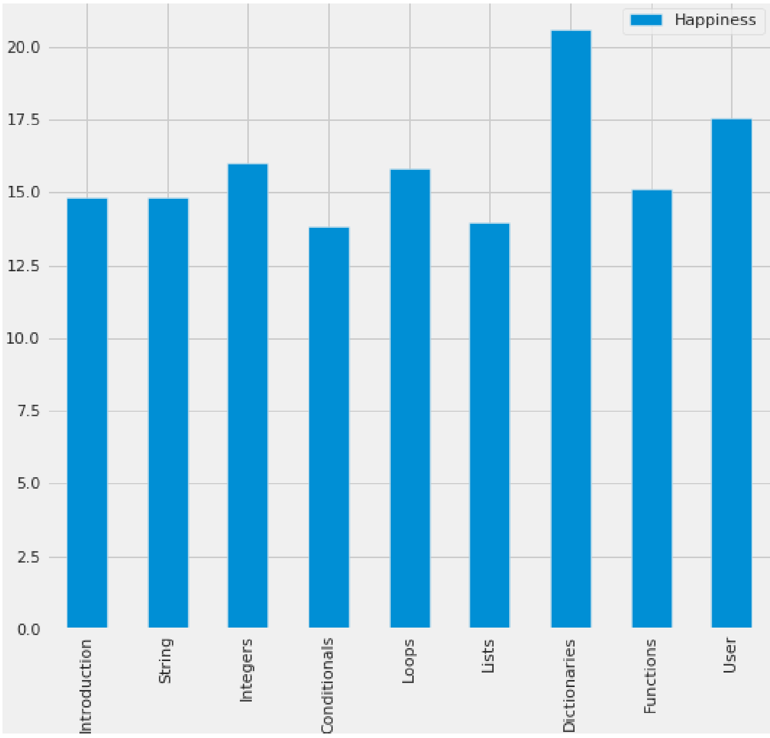

Figure 6.20: Happiness intensity evolution per test

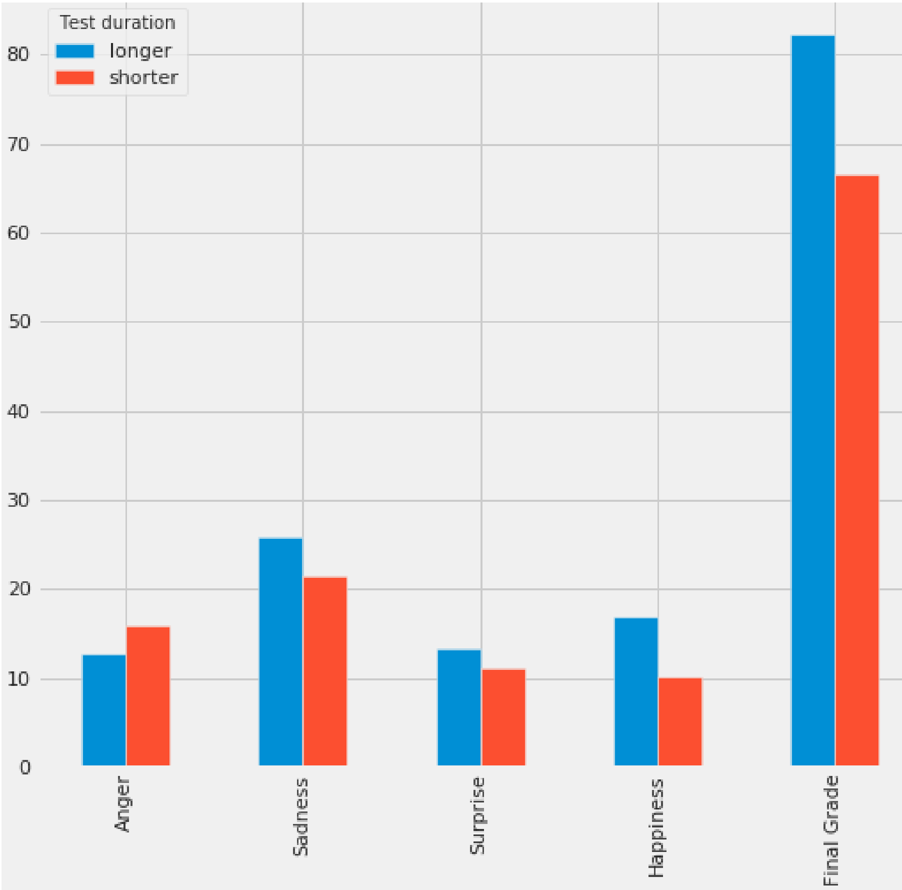

Figure 6.21: Impact of test duration on emotions captured and on students' grades

6.5 Smart Automation for students and teachers

This last case study is based, like the previous ones, on the use of attention and emotion capturing modules. However, in this case also comes into play the Semantic Task Automation platform presented in previous sections: Ewetasker. The main objective is to improve the comfort and mood of students while they are studying, through the use of smart devices and task automation. For this purpose, the scenarios shown in Figure 6.22 have been developed.

In order to carry out the proposed scenarios, it is necessary for the actors involved in them (students and teachers) to create automation rules with Ewetasker. For this reason, the following section will explain how to use the Semantic Task Automation platform, in such a way that it serves as a starting point for the case studies that will be detailed later.

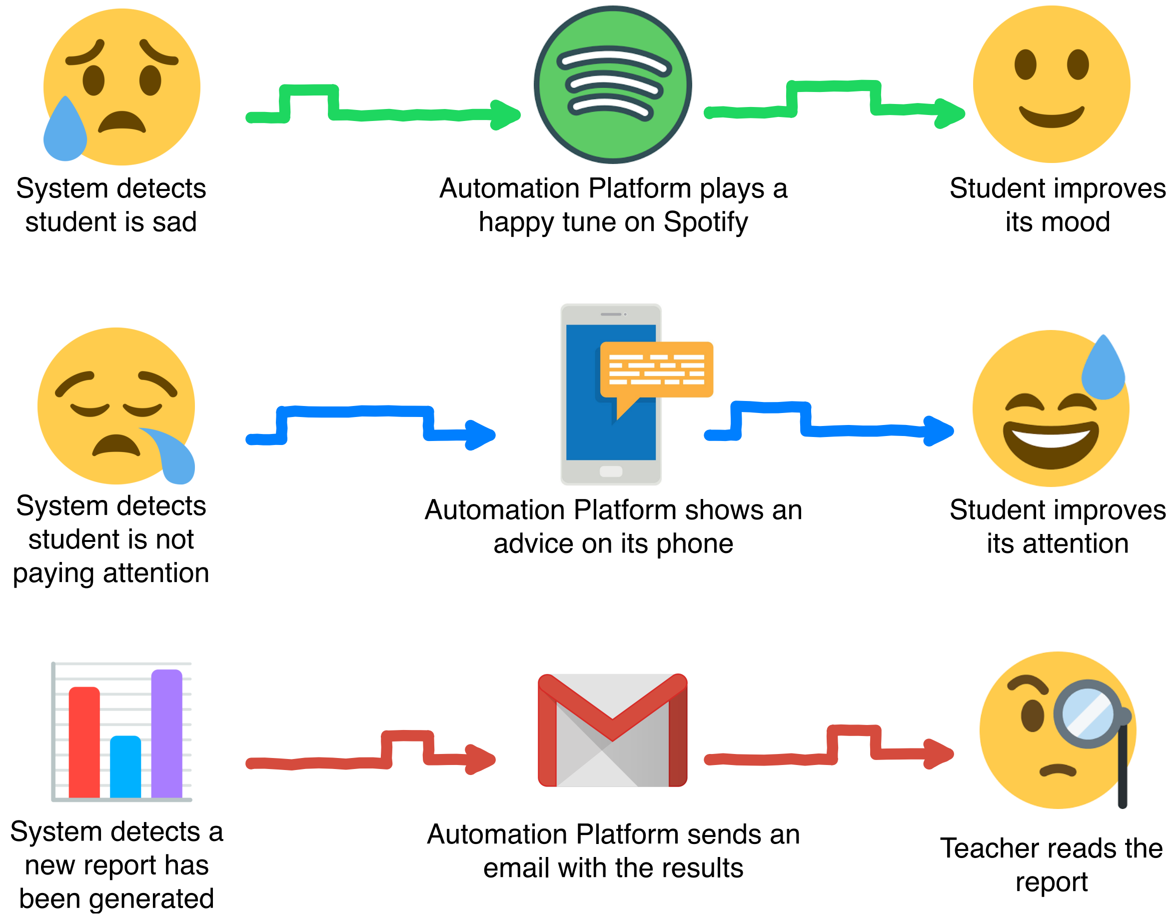

Figure 6.22: Smart Automation Case Study

6.5.1 Creating Smart Automation Rules

The process of creating automation rules begins once users have been registered on the Ewetasker System. The home page shows a list of available channels, including the Emotion Aware E-learning platform. In order to create rules, user must register the devices or services to be used. This can be achieved in the devices or services tab respectively. However, in this case study, only the registration of the learning platform channel is detailed in order to be as brief as possible.

On this page, shown in Figure 6.23, the user must import the EAES channel for the specific use it will be given. To do this, it is necessary to complete a form that includes the name and description of the service, as well as the course parameter that will be used in the creation of rules and events. In this case we will define the course parameter by its initials (PYLB).

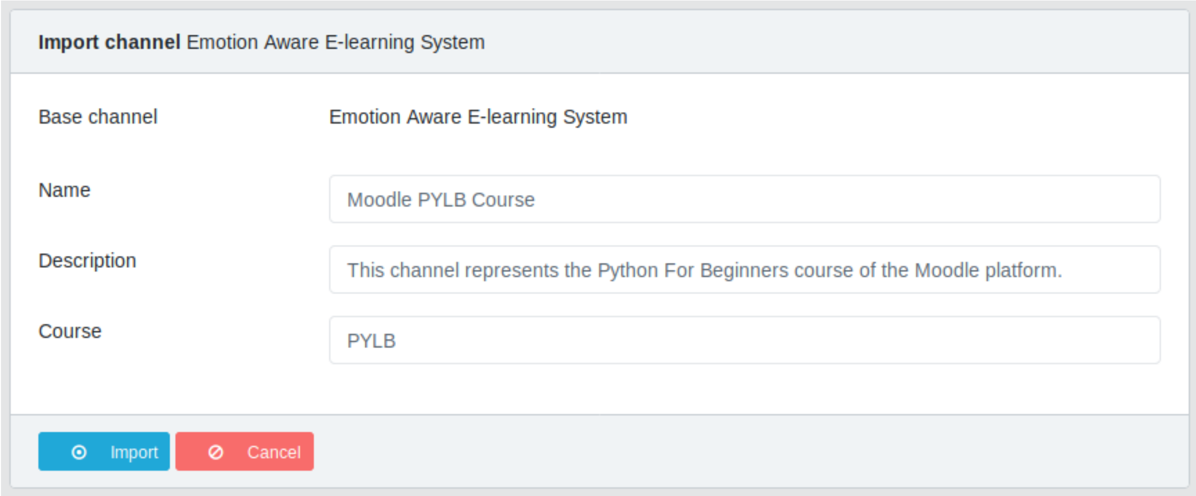

Figure 6.23: Ewetasker Import Services View

Once this is done, the user should go to the Rules page, shown in 6.24, which displays all the rules created and the option to create new ones. When the user clicks on the option to create a new rule, a list appears with all the available channels and a container in which the channels that intervene in the rule will be placed. The user must select the events and actions of the rule, by dragging and dropping them into the container. Each time an event or action is dropped in the container, a modal view appears, which allows users to define more specific parameters of the rule such as attention intensisty or emotion experienced. Finally, user must press the *"submit"* button to save the rule he has just created.

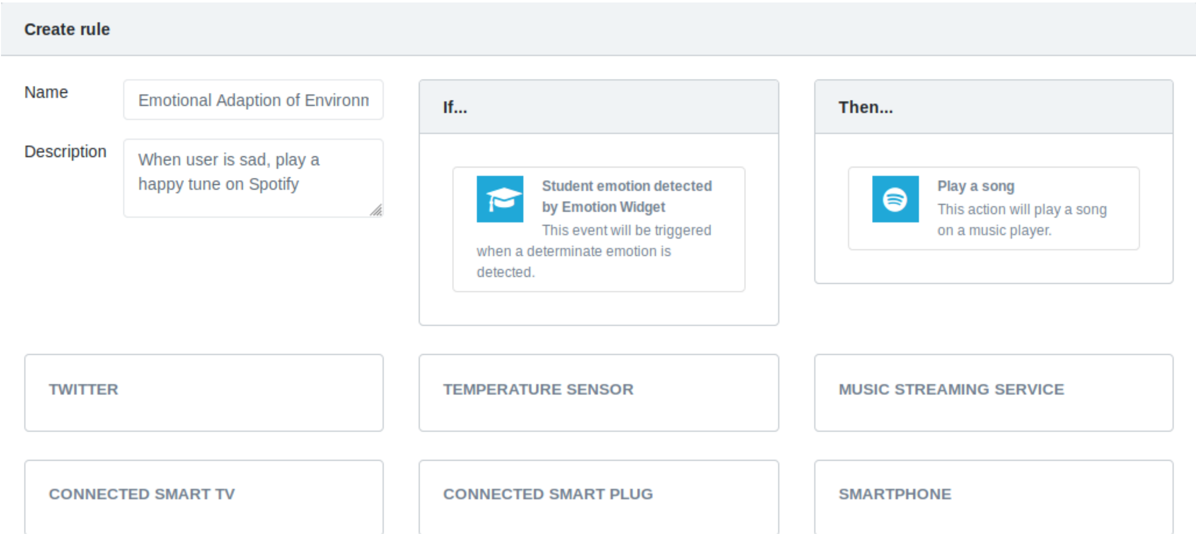

Figure 6.24: Ewetasker Rules Creation View

As we have seen, the process of creating automation rules is simple and fast, which facilitates their application in the different case studies that we will see below.

6.5.2 Emotion Automation Case Study

This use case goal is to adapt the environment to the emotions experienced by a student when he is studying through the use of smart devices present in his workplace. With this purpose, two modules implemented in EAES will be used: the emotion capturer and the event trigger. The first one will measure the emotions of the student, and the second one will send every minute an event to the automation platform with the intensity measured of each emotion. The only actor involved in this case is the student.

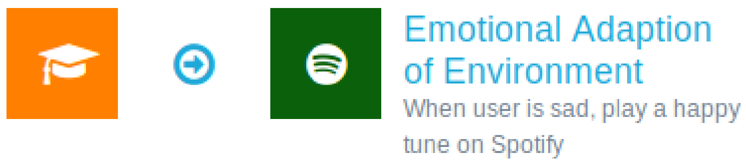

Figure 6.25: Emotion Automation Rule

The use case starts with the student creating automations through Ewetasker as seen in Section 6.5.1. Within these automations, he creates a rule, shown in Figure 6.25, that plays cheerful music on Spotify, if EAES recognises that he is sad when doing course activities. Similarly, student can create other rules that involve other devices or services, such as adapting the temperature of the room or changing its brightness.

In this way, when student is studying the course slides, the work environment will adapt to his mood, improving his comfort and, consequently, his performance. Figure 6.26 shows an example of the implementation of this case study.

6.5.3 Attention Automation Case Study

The aim of this case study is to improve the attention of students when they are studying the course contents through the e-learning platform. To do this, it will be necessary to use the attention-recognition and event trigger modules. The first will be in charge of measuring the attention paid per minute, and the second will be in charge of sending this information as an event to the automation platform. As in the previous case, the main actor is the student.

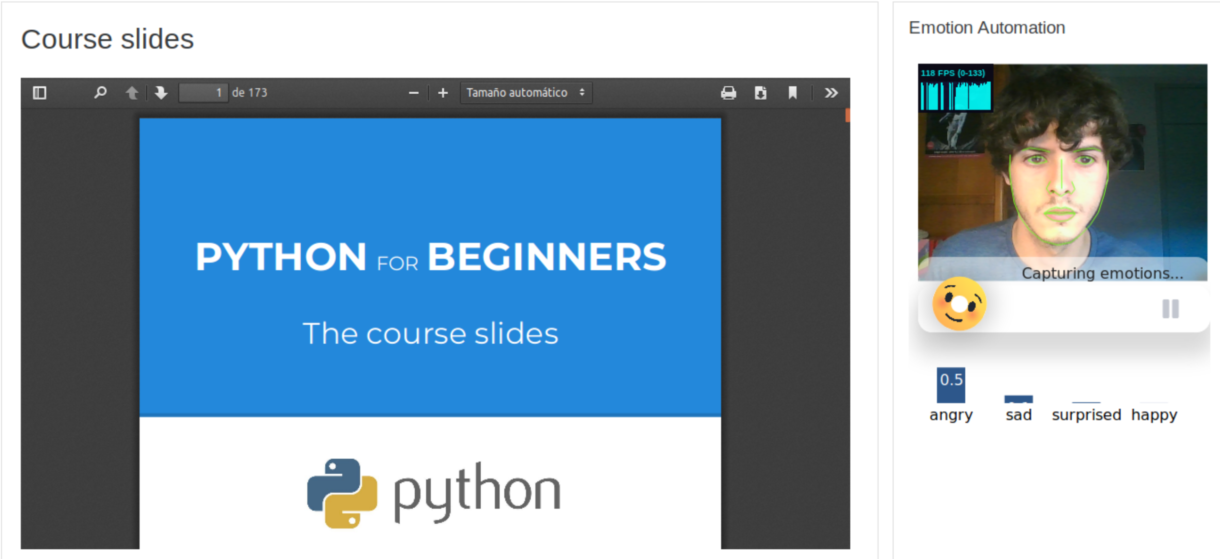

Figure 6.26: Emotion Automation Example

The first action that takes place in this use case is that of the student creating an automation rule in Ewetasker. This rule, shown in Figure 6.27, defines that if the attention paid to the slides is less than a certain threshold, a notification is sent to the phone alerting the student of his behaviour. However, this is only one of the possible automations that can be created for this use case, being able to warn the user through other types of actions, such as light signals with the desk lamp or sound signals with the music player.

Figure 6.27: Attention Automation Rule

Through these automations the student can improve his concentration, optimizing the study time and obtaining better academic results.

6.5.4 Report Automation Case Study

The aim of this case study is to facilitate exploratory analysis of the course data presented in Section 6.4. This data analysis, when we are faced with large datasets, becomes a process that can take from the order of minutes to hours. Waiting until the end of this process can discourage teachers from using these analysis techniques, leaving them in disuse or undervalued. For this reason, and taking advantage of the capabilities of the automation platform, a warning has been implemented to inform teachers of the completion of the analysis. The only actor involved in this case is the teacher.

The case study begins when the teacher creates an automation rule in Ewetasker. This rule, shown in Figure 6.28, defines that when an analysis task finishes, an email message with the results will be sent to the teacher. Once created the automation, the teacher performs an analysis of the data through the EAES interface. It is then, when the process of extraction, pre-processing, and data analysis, detailed in previous chapters, begins.

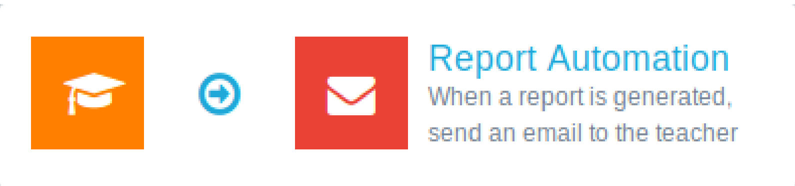

Figure 6.28: Report Automation Rule

The number of resulting files is considerable, which complicates the task of downloading and viewing by the teacher. To deal with this problem, at the end of the analysis tasks the files are compressed in zip format. When this process finishes, the EAES sends an event to Ewetasker, informing that the analysis of the user data has finished. This event will activate the created rule, and finally, an email will be sent to the teacher with the generated report.

Thanks to this automation, the teacher can carry out analysis tasks and obtain the results in his personal mail in a few simple steps.

6.6 Conclusions

In this chapter, we have presented the case studies we have designed to improve the academic performance of students participating in an online course through the study of their emotions.

To this end, firstly, the use of visualization tools has been proposed to broaden the knowledge of both teachers and students about the interactions that occur during the course activities. These interactions include data pertaining to emotions, attention and course grades.

Secondly, the use of these visualizations as tools for self-reflection and self-knowledge for students has been motivated. In this way, students can know the impact of their emotions and attention on the score obtained, being able to change their attitude in the future to achieve better results.

Thirdly, an analysis of the data has been carried out using ML techniques, which allow teachers to extend and contrast the information shown on the dashboards, as well as to infer and discover new relationships and data.

Finally, we have developed three cases of joint use of the e-learning system together with the task automation platform. In each case we have explained how the interactions between the different actors take place, with what purpose they take place, and what actions they trigger.

CHAPTER 7

Conclusions

In this chapter we will describe the conclusions extracted from this master thesis, problems, achievements and thoughts about future work.

7.1 Conclusion

To conclude this master thesis document, we are going to recapitulate about the implemented system. We have developed a system for online learning platforms, specifically designed for Moodle, which allows us to capture data about emotions, the attention paid by students, and the grades obtained. The system in question, called Emotion Aware E-learning System, aims to improve the academic performance of students from the data obtained. This system consists of a web interface that implements the data capture modules, a server that contains all the logic of the system as well as the data storage and analysis modules, and a visualization module that implements different dashboards. In addition, a Semantic Automation Platform, called Ewetasker, has been developed for joint use with this system. This platform also includes a mobile application to expand its capabilities.

The development of this project has been motivated by two currents of research that have great weight today. Firstly, emotion-aware systems applied to online learning environments. These systems are able to recognize emotions and know the mood of students. Through this information, teachers can adapt the lessons to improve the academic results of their students.

Second, the use of ML techniques to explore academic data, widely known as Learning Analytics, allows teachers to extract information hidden in the data, such as relationships, anomalous cases, or trends. In this way, the usefulness of the data is increased, and consequently, the results of the actions carried out by the teachers are improved.

Moreover, the use of captured data for workspace adaptation adds to society's growing interest in smart devices. These adaptations, as we have seen in other research projects, provide immediate benefits in the comfort of users, which has a positive impact on their performance. For this reason, the use of a task automation platform has been considered of interest to students.

To conclude, the project is developed in such a way as to facilitate its deployment and implementation in multiple environments. Each module described in Chapter 5 is run in Docker containers, providing security, error isolation, and facilitating a possible deployment in the cloud.

7.2 Achieved Goals

The achieved goals for this project are the following:

- We have designed and implemented a system capable of detecting the mood of students in an online course during the performance of different activities.
- We have designed and implemented a visualization system to display the data collected by the emotion-aware system along with academic data.
- We have designed and implemented an analytical learning system to explore the data, the state of mind, and academic results obtained on the different activities carried out in the course.
- We have integrated our system in a smart educational environment through a semantic task automation platform.
- We tested the system in a real environment with a group of thirty university students.

7.3 Future work

In this section, the possible new features or improvements that could be done to the project will be explained.

- Improve the integration of the widgets developed with the web interface of the learning platform. This would facilitate their use by students and teachers, increasing the quantity and quality of data collected.
- Add new data sources, or improve the skills of the emotion-recognizers used.
- Test the capabilities of the system on a broader set of students. In this way the results obtained in analysis performed with ML techniques would be improved and prediction algorithms could be applied.
- Create new intelligent automations, taking into account the environment in which the system is deployed and the needs or suggestions of students and teachers.
- Improve the usefulness of the visualizations displayed to the users by adding an explanation of the data shown.

APPENDIX A

Impact of the project

E-learning platforms play a key role in today's education systems, whether in face-to-face courses or in distance environments. For this reason, a system that could have a significant impact on the academic performance of students could also have a major impact on the economy and daily life of a country. On the other hand, different educational institutions, both public and private, need to be at the forefront of learning methods and techniques in order not to lag behind in the race to provide the best possible learning environment for their students.

In this appendix, we are going to talk about the possible social, economic and environmental impact that this project could have in Sections A.1, A.2 and A.3, respectively. We will also give the possible ethical and professional implications of such project in Section A.4.

A.1 Social Impact

The social impact of this project can be measured in terms of service quality and user experience. These terms, extrapolated to the learning context, imply an improvement in academic results and a learning experience adapted to the needs of students. Similarly, if we look at a more specific environment within the academic context, such as e-learning, we will see that this project also helps to overcome the communication barriers that appear using this type of learning platforms.

Another issue to consider is the impact on users' privacy. In this project we use data from different sources such as webcam or physiological sensors among others. For the successful operation of the system, these data are collected associated with a student identifier through personal devices, so it would be necessary to have the consent of users based on current legislation on data protection. In addition, anonymization operations could be carried out on the data to be used in Machine Learning, as it is not necessary that data remain linked to an identity.

A.2 Economic Impact

The aim of this project is to improve the experience and learning outcomes in e-learning platforms, so that different agents can be positively, and consequently, economically affected by its development.

Firstly, students are the first to make the most of the money and time invested in their learning. Secondly, teachers could significantly reduce the time spent searching for methodologies and solutions that improve their students' academic results. Thirdly, public and private entities, which implement e-learning platforms to offer learning services, would see their popularity and benefits increase due to the improvement of their students' academic results. Finally, the society can opt for an improvement of the economy in general if workers skills are improved.

A.3 Environmental Impact

The development of this project and the subsequent implementation of the resulting system do not have a direct impact on the environment. However, if this case of use is extrapolated to a much larger scenario with numerous users, the deployment and execution of the subsystems presented will require a considerable amount of computing resources.

In order to address this problem, this system could be deployed as a cloud service. These services use computing resources that can be dynamically adapted in response to demand. The resources that implement these services in the cloud have a much higher consumption compared to a personal computer. However, by means of this technology, it is possible to deploy several services at the same time using the same resource, reducing the total amount of energy consumed.

A.4 Ethical and Professional Implications

The ethical implications of this project are related to data collection as explained in A.1.

Data collection should always be carried out with users consent once they are informed of the purpose. Within this data we distinguish two types. Data obtained from the e-learning platform to which teachers already have access and new data relating to students mood. The terms of use, when using free software platforms such as Moodle, are defined by the entities that implement these platforms in accordance with the legislation of each country and the treatment that will be made of the data. Our system only collects, displays and makes predictions based on information that was already mostly accessible in the traditional platforms for the subjects involved (teachers and students).

The main risk associated with the collection of these data is that they could be accessed by an external party overcoming security gates. However, the risk is minimal since these platforms deal with data of greater relevance than those used in this project, and consequently, they implement security methods with high reliability.

The aim of this project is to improve the experience and learning outcomes in e-learning platforms, whereby the main beneficiaries will be teachers and students.

APPENDIX B

Economic budget.

This master thesis consists of the Design and Development of an Emotion-aware Learning Analytics system based on Machine Learning Techniques and Semantic Task Automation. In this process, we have incurred in some costs considering material and human resources.

In this appendix, we are going to resume the possible costs involved in this development. Firstly, in Section B.1 we are going to calculate the human resources needed for this master thesis' design and development. In addition, in Section B.2 the costs of the material resources needed for this project are going to be described. Finally, the licenses used in this project are going to be described on Section B.3

B.1 Human resources

In this section, we will take into account the time employed in the designing, developing and testing this system. We will give an approximation based on the average salary of a Telecommunication Engineer, to find the cost of the development of the project.

The estimate of the working time used to carry out this project has been calculated on the basis of ECTS credits [1]. A master's thesis consists of 30 ECTS credits, each representing 25 to 30 hours of work. This makes a total of 900 hours of work, or four months of full-time work. If we consider a gross salary of 1500 euros per month, the cost of the project amounts to 6000 euros.

This cost includes the design, development and testing tasks for the creation of the system. The cost of system maintenance and operation is not considered very high.

B.2 Material resources

The following material resources have been used to carry out this project. Firstly, a personal computer in which the design, development and testing tasks can be carried out. Secondly, an E4 bracelet from Empatica, to monitor physilogical signals in real time. And finally, a cloud computing environment or a dedicated cluster in which to deploy the final system to be accessible by users.

The personal computer on which the project has been developed costs approximately 700 euros by mid 2019. The technical characteristics are as follows:

- **CPU:** Intel Core i5 2.7GHz
- **Memory:** 8GB RAM DDR4
- **Hard Disk:** 250GB SSD

Empatica's E4 bracelet [2] costs approximately 1500 euros by mid 2019.

[1] https://ec.europa.eu/education/resources/european-credit-transfer-accumulation-system_en
[2] https://store.empatica.com/products/e4-wristband

Finally, it is necessary to have a cloud service or VPS service to deploy the different modules of the system, including storage modules (MongoDB, Elasticsearch and Fuseki) and web servers. The price of these services depends on the traffic and load used. Similarly, prices vary depending on the provider. For this project, a cost of 50 euros per month has been estimated in order to comply with the minimum system requirements.

B.3 Licenses

The software used in the development of the project is open-source software.

APPENDIX C

Installation and deployment

In this appendix, we are going to describe the steps to deploy and maintain the system using docker containers with docker-compose.

C.1 Requirements

This installation guide is oriented for Linux machines capable of running Docker engine. Since docker is a technology that can be used in other operating systems, this guide can serve as an example for all of them. The prerequisites are docker and git tools. In order to install Docker, you can follow the steps founded in the official website [1] and install git with the following command.

```
$ sudo apt install git
```

Once prerequisites have been met, the installation can begin. This installation is divided into two parts: Emotion Aware E-learning System and Ewetasker.

C.2 Running Emotion Aware E-learning System with docker-compose

The installation of this system includes all the modules that have been used in this project except the semantic task automation platform. To do this, it is necessary to download the Github repository and navigate to the cloned folder:

```
$ git clone https://github.com/enriquetolby/emotion-moodle
$ cd emotion-moodle
```

Once this is done, we can deploy our system using docker-compose. Docker-compose is a tool for defining and running multi-container Docker applications.

```
$ docker-compose up --build
```

The modules or subsystems that make up the complete system are deployed separately into docker containers. These containers communicate with each other by creating a docker-network. Below is a list of the modules that are deployed in this system:

- **Moodle:** This container deploys Moodle e-learning platform. This service is available on http://localhost:5151/.

[1] https://docs.docker.com/install/linux/docker-ce/ubuntu/

- **Mariadb:** This container implements a MariaDB database required for the operation of Moodle.
- **Emotion Moodle:** This container includes the Emotion Widget web interface that implements the data capture submodules. This service is available on http://localhost:5151/.
- **Emotion Api:** This container includes a REST API that communicates the different modules with the system logic. This service is available on http://localhost:3001/.
- **Mongo:** This container implements a MongoDB database required for the operation of Emotion Recognition module. This service is available on http://localhost:17017/.
- **Elasticsearch:** This container implements a Elasticsearch database in order to ingest the data obtained by Emotion Recognition module. This service is available on http://localhost:29200/.
- **Kibana:** This container deploys the Kibana service that allows you to create visualizations based on Elasticsearch data. This service is available on http://localhost:5601/.

The docker-compose file is a YAML file defining services, networks and volumes. Our file is detailed below:

```
version: '3'
services:
  moodle:
    image: 'bitnami/moodle:latest'
    environment:
      - MARIADB_HOST=mariadb
      - MARIADB_PORT_NUMBER=3306
      - MOODLE_DATABASE_USER=bn_moodle
      - MOODLE_DATABASE_NAME=bitnami_moodle
      - ALLOW_EMPTY_PASSWORD=yes
    labels:
      kompose.service.type: nodeport
    ports:
      - '5580:80'
      - '443:443'
    volumes:
      - 'moodle_data:/bitnami'
    depends_on:
      - mariadb

  mariadb:
```

```
    image: 'bitnami/mariadb:latest'
    environment:
      - MARIADB_USER=bn_moodle
      - MARIADB_DATABASE=bitnami_moodle
      - ALLOW_EMPTY_PASSWORD=yes
    volumes:
      - 'mariadb_data:/bitnami'

  emotion-moodle:
    container_name: emotion-moodle
    build:
      context: ./emotionmoodle
      args:
        PORT: 3000
    ports:
      - "5151:3000"
    volumes:
    - ./emotionmoodle:/usr/src/app
    networks:
      - emo_network

  emotion-api:
    container_name: emotion-api
    build:
      context: ./emotionapi
      args:
        PORT: 3000
    ports:
      - "3001:3000"
    volumes:
      - ./emotionapi:/usr/src/app
    networks:
      - emo_network
    depends_on:
      - mongo

  mongo:
    image: mongo:latest
    container_name: emomongodb
    ports:
      - "17017:27017"
    volumes:
      - emomongodata:/data/db
    networks:
      - emo_network
```

```
  moodleelasticsearch:
    image: "docker.elastic.co/elasticsearch/elasticsearch:6.7.0"
    container_name: moodleelasticsearch
    ulimits:
      memlock:
        soft: -1
        hard: -1
    environment:
      - cluster.name=docker-cluster
      - bootstrap.memory_lock=false
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
      - "xpack.security.enabled=false"
      - "http.cors.enabled=true"
      - 'http.cors.allow-origin=*'
    volumes:
      - esmoodledata:/usr/share/elasticsearch/data/
    ports:
      - 29200:9200
      - 29300:9300
    networks:
      - emo_network

  kibana:
    container_name: kibana
    build:
      context: kibana/
      args:
        ELK_VERSION: 6.7.0
    volumes:
      - ./kibana/config/:/usr/share/kibana/config:ro
    ports:
      - "5601:5601"
    networks:
      - emo_network
    depends_on:
      - moodleelasticsearch
networks:
  emo_network:
    driver: bridge
volumes:
  emomongodata:
  esmoodledata:
  mariadb_data:
  moodle_data:
```

Moodle, MariaDB, MongoDB, Elasticsearch, and Kibana images are downloaded from the Docker Hub [2] platform. This platform provides an extensive catalogue of docker images. In addition, these images are updated and maintained by their own developers. However, in this project, it has been necessary to build specific images for each of the modules developed: Emotion Moodle and Emotion Api.

Emotion Moodle image is based on the node:10-alpine base image and necessary dependencies and code are added over it. The Dockerfile for building this image is detailed below:

```
FROM node:10-alpine

ARG PORT
ENV PORT $PORT
EXPOSE $PORT
WORKDIR /usr/src/app
COPY package*.json ./
RUN npm install
COPY . ./

CMD [ "npm", "start" ]
```

Emotion Api image is based on the node:10-alpine base image and necessary dependencies and code are added over it. The resulting Dockerfile is this:

```
FROM node:10-alpine

ARG PORT
ENV PORT $PORT
EXPOSE $PORT
WORKDIR /usr/src/app
ADD requirements.txt /usr/src/app/
COPY package*.json ./
RUN apk update
RUN apk --no-cache --update-cache add gcc gfortran python python-dev py
    -pip build-base wget freetype-dev libpng-dev openblas-dev
    wkhtmltopdf
RUN ln -s /usr/include/locale.h /usr/include/xlocale.h
RUN pip install -r /usr/src/app/requirements.txt
RUN  npm install -g nodemon && npm install
COPY . ./

CMD ["nodemon", "-L", "emotion_api.js"]
```

[2] https://hub.docker.com/

C.3 Running Ewetasker with docker-compose

The installation of this system includes all the modules that make up the semantic task automation platform. However, this platform is divided into two parts: Ewetasker Webclient and Ewetasker Server. For a correct installation of the platform the two parts are shown separately.

C.3.1 Ewetasker Server

As with the previous installation, it is necessary to download the GitLab repository and navigate to the cloned folder:

```
$ git clone https://lab.gsi.upm.es/ewe/ewetasker_server
$ cd ewetasker_server
```

Once this is done, we can deploy our system using docker-compose.

```
$ docker-compose up --build
```

Below is a list of the modules that are deployed in this system:

- **Ewetasker Server:** This container deploys the server where the platform logic is located. This service is available on http://localhost:5050/.
- **Crossbar:** This container deploys the Crossbar proxy to manage the events triggered to the platform. This service is available on http://localhost:8082/ for HTTP protocol, on http://localhost:8081/ for WAMP protocol and on http://localhost:1883/ for MQTT protocol.
- **Eye Server:** This container EYE reasoning engine. This service is available on http://localhost:4040/.
- **Fuseki:** This container implements a Fuseki database required for the storage of semantic definitions.
- **Mongo:** This container implements a MongoDB database required for the operation of Ewetasker Webclient. This service is available on http://localhost:27027/.

- **Elasticsearch:** This container implements a Elasticsearch database in order to ingest the data obtained from platform interactions. This service is available on http://localhost:19200/.

The docker-compose file is detailed below:

```
version: '2'
services:
  api:
    build: ./ewetasker
    container_name: api
    ports:
      - "5050:5000"
    networks:
      - ewetasker
    environment:
      - ES_ENDPOINT=elasticsearch
      - ES_ENDPOINT_PORT=9200
      - SPARQL_URL=fuseki:3030/ewetasker
      - MONGODB_URL=mongo:27017
      - EYE_URL=eye:8000
      - TWITTER_TOKEN=token
      - TWITTER_SECRET=secret
      - GMAIL_CLIENT_ID=token
      - GMAIL_CLIENT_SECRET=secret
      - API_REDIRECT_URI=api
      - SPOTIFY_CLIENT_ID=id
      - SPOTIFY_CLIENT_SECRET=secret
      - EWE_PEM=ewetasker.pem
    depends_on:
      - fuseki
      - mongo
      - elasticsearch
    command: ["./wait-for-it.sh", "elasticsearch:9200", "--", "python",
         "app.py"]

  fuseki:
    image: stain/jena-fuseki
    container_name: fuseki
    ports:
      - "3030:3030"
    volumes:
      - fuseki:/fuseki
    environment:
      - ADMIN_PASSWORD="ewefuseki"
```

```
    networks:
      - ewetasker

  mongo:
    image: mongo:latest
    container_name: ewemongodb
    ports:
      - "27017:27017"
    volumes:
      - mongodbdata:/data/db
    environment:
      - MONGODB_USER="ewemongo"
      - MONGODB_PASS="gsimongodb2018"
    networks:
      - ewetasker

  eye:
    image: bdevloed/eyeserver
    container_name: eye
    ports:
      - "4040:8000"
    networks:
      - ewetasker
  elasticsearch:
    image: "docker.elastic.co/elasticsearch/elasticsearch:5.5.2"
    container_name: elasticsearch
    ulimits:
      memlock:
        soft: -1
        hard: -1
    environment:
      - cluster.name=docker-cluster
      - bootstrap.memory_lock=false
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
      - "xpack.security.enabled=false"
      - "http.cors.enabled=true"
      - 'http.cors.allow-origin=*'
    volumes:
      - esdata:/usr/share/elasticsearch/data/
    ports:
      - 19200:9200
      - 19300:9300
    networks:
      - ewetasker
```

```
    crossbar:
      build: ./ewe-crossbar
      container_name: crossbar
      environment:
        - API=api:5000
      ports:
        - "8081:8081"
        - "8082:8082"
        - "1883:1883"
      depends_on:
        - api
      networks:
        - ewetasker
      entrypoint: []
      command: >
         bash -c "cd ./ && cd /crossbar &&  ./wait-for-it.sh api:5000 --
             crossbar start --cbdir /crossbar/.crossbar"

  networks:
    ewetasker:
      driver: bridge
  volumes:
    fuseki:
    esdata:
    mongodbdata:
```

Eye, Fuseki, MongoDB,and Elasticsearch images are downloaded from the Docker Hub platform. As same as before, in this project, it has been necessary to build specific images for each of the modules developed: Ewetasker Server and Crossbar.

Ewetasker Server image is based on the Python 3.6 base image and necessary dependencies and code are added over it. The Dockerfile for building this image is detailed below:

```
FROM python:3.6

RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app
ADD requirements.txt /usr/src/app/
RUN pip install -r /usr/src/app/requirements.txt
ADD . /usr/src/app
RUN openssl genrsa -out ewetasker.pem 2048

CMD ["python", "app.py"]
```

Crossbar image is based on the Crossbar base image and necessary dependencies and code are added over it. The resulting Dockerfile is this:

```
FROM crossbario/crossbar

USER root
RUN pip install -U pytest
RUN pip install objectpath
RUN pip install voluptuous
RUN pip install pytz
COPY ./crossbar /crossbar
RUN chown -R crossbar:crossbar /crossbar
RUN apt-get update && apt-get install bash
ENTRYPOINT ["crossbar"]

CMD ["start", "--cbdir", "/crossbar/.crossbar"]
```

C.3.2 Ewetasker Webclient

First, it is necessary to download the GitLab repository and navigate to the cloned folder:

```
$ git clone https://lab.gsi.upm.es/ewe/ewetasker_webclient
$ cd ewetasker_webclient
```

Once this is done, we can deploy our system using docker-compose.

```
$ docker-compose up --build
```

Below is a list of the modules that are deployed in this system:

- **Ewetasker Webclient:** This container implements a web application for Ewetasker platform. This service is available on http://localhost:5000/.
- **Sefarad:** This visualization environment provides a dashboard for Ewetasker. This service is available on http://localhost:19200/.

The docker-compose file is detailed below:

```
version: '3'
services:
  webui:
    container_name: ewewebclient
    build:
      context: ./
    ports:
      - "5000:5000"
  dashboard_admin:
    container_name: dashboard_admin
    build: ./admin-dashboard
    ports:
     - "7070:8080"
    environment:
      - ES_ENDPOINT_EXTERNAL=http://localhost:7070
    volumes:
    - ./admin-dashboard:/usr/src/app
```

In this part, it has also been necessary to build specific images for each of the modules developed: Ewetasker Webclient and Sefarad.

Ewetasker Webclient image is based on the Node 10.11.0 base image and necessary dependencies and code are added over it. The Dockerfile for building this image is detailed below:

```
FROM node:10.11.0

RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app
ADD . /usr/src/app
RUN yarn
RUN yarn build
RUN yarn global add serve

CMD serve -s build
```

Sefarad image is based on the Node 7.10 base image and custom widgets and dashboard are installed over it. The resulting Dockerfile is this:

```
FROM node:7.10.0

RUN apt-get update && apt-get install -y gettext
ENV NODE_PATH=/tmp/node_modules APP_NAME=admin-dashboard
RUN npm install -g http-server bower
RUN npm install -g polymer-cli@1.1.0
WORKDIR /usr/src/app/
RUN mkdir -p /etc/ssl/certs
COPY lab.cluster.gsi.dit.upm.es.crt /usr/local/share/ca-certificates/
COPY ca.crt /usr/local/share/ca-certificates
RUN update-ca-certificates
RUN cat /usr/local/share/ca-certificates/lab.cluster.gsi.dit.upm.es.crt
     >> /etc/ssl/certs/ca-certificates.crt
COPY tweet-chart.cluster.gsi.dit.upm.es /
RUN cat /tweet-chart.cluster.gsi.dit.upm.es >> /etc/ssl/certs/ca-
    certificates.crt
ADD bower.json /usr/src/app/bower.json
RUN bower link --allow-root
RUN bower install --allow-root  && mv bower_components ..
ADD . /usr/src/app

CMD ["/usr/src/app/init.sh"]
```

APPENDIX D

EWE Channel Definition

In this appendix, we are going to define semantically a channel for the Emotion Aware E-learning System developed in this project.

D.1 Emotion Aware E-learning System Channel

As mentioned in Section 4, in order to integrate the Ewetasker semantic automation platform with the developed system for e-learning platforms, it has been necessary to create a channel using the EWE ontology. This channel is shown below:

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ewe: <http://gsi.dit.upm.es/ontologies/ewe/ns/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://example.org/ex#> .
@prefix dogont: <http://elite.polito.it/ontologies/dogont.owl#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix ewe-service:<http://gsi.dit.upm.es/ontologies/ewe-service/ns/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix onyx: <http://gsi.dit.upm.es/ontologies/onyx/ns/> .
@prefix math: <http://www.w3.org/2000/10/swap/math#> .
@prefix string: <http://www.w3.org/2000/10/swap/string#> .
@base  <http://gsi.dit.upm.es/ontologies/ewe/ns/> .




###################################
# Channel definition
###################################
ewe:EmotionAwareElearningSystem a owl:Class ;
    rdfs:label "Emotion Aware E-learning System" ;
    rdfs:comment "This channel represents Emotion Aware E-learning
        System for Moodle platform." ;
    foaf:logo "fa fa-graduation-cap" ;
    dbo:colour dbo:Orange ;
    ewe:generatesEvent ewe:EmotionDetected ;
    ewe:generatesEvent ewe:EmotionReported ;
    ewe:generatesEvent ewe:AttentionDetected ;
    ewe:generatesEvent ewe:ReportGeneratedElearning ;
    ewe:hasCategory ewe-service:Service ;
    rdfs:subClassOf ewe:Channel .
```

```
###################################
# Events definition
###################################
ewe:EmotionDetected a owl:Class ;
    rdfs:label "Student emotion detected by Emotion Recognition Module"
         ;
    rdfs:comment "This event will be triggered when a determinate
        emotion is detected." ;
    ewe:hasOutputParameter ewe:Emotion ;
    ewe:hasInputParameter ewe:Course ;
    ewe:hasOutputParameter ewe:EmotionIntensity ;
    rdfs:subClassOf ewe:Event ;
    rdfs:domain ewe:EmotionAwareElearningSystem .

ewe:EmotionReported a owl:Class ;
    rdfs:label "Emotion reported by student" ;
    rdfs:comment "This event will be triggered when a determinate
        emotion is reported." ;
    ewe:hasOutputParameter ewe:Emotion ;
    ewe:hasInputParameter ewe:Course ;
    ewe:hasOutputParameter ewe:EmotionIntensity ;
    rdfs:subClassOf ewe:Event ;
    rdfs:domain ewe:EmotionAwareElearningSystem .

ewe:AttentionDetected a owl:Class ;
    rdfs:label "Student attention detected by Emotion Recognition
        Module" ;
    rdfs:comment "This event will be triggered when a determinate
        attention intensity is detected." ;
    ewe:hasOutputParameter ewe:AttentionIntensity ;
    ewe:hasInputParameter ewe:Course ;
    rdfs:subClassOf ewe:Event ;
    rdfs:domain ewe:EmotionAwareElearningSystem .

ewe:ReportGeneratedElearning a owl:Class ;
    rdfs:label "Course report generated with ML techniques" ;
    rdfs:comment "This event will be triggered when a course report was
         generated." ;
    ewe:hasInputParameter ewe:ElearningCourse ;
    rdfs:subClassOf ewe:Event ;
    rdfs:domain ewe:EmotionAwareElearningSystem .
```

```
####################################
# Parameters definition
####################################
ewe:Emotion a owl:Class ;
    rdfs:label "Emotion" ;
    rdfs:comment "This parameter represents the specific emotion
        detected." ;
    rdf:datatype xsd:string ;
    rdfs:subClassOf ewe:Parameter ;
    rdfs:subClassOf onyx:Emotion ;
    ewe:operation string:equalIgnoringCase ;
    rdfs:domain ewe:EmotionAwareElearningSystem .

ewe:Course a owl:Class ;
    rdfs:label "Course" ;
    rdfs:comment "This parameter represents the e-learning course." ;
    rdf:datatype xsd:string ;
    rdfs:subClassOf ewe:Parameter ;
    ewe:operation string:equalIgnoringCase ;
    rdfs:domain ewe:EmotionAwareElearningSystem .

ewe:EmotionIntensity a owl:Class ;
    rdfs:label "Emotion Intensity" ;
    rdfs:comment "This parameter represents the emotion intensity." ;
    rdf:datatype xsd:math ;
    ewe:operation math:lessThan ;
    ewe:operation math:greaterThan ;
    rdfs:subClassOf ewe:Parameter ;
    rdfs:domain ewe:EmotionAwareElearningSystem .

ewe:AttentionIntensity a owl:Class ;
    rdfs:label "Attention Intensity" ;
    rdfs:comment "This parameter represents the attention measured." ;
    rdf:datatype xsd:math ;
    ewe:operation math:lessThan ;
    ewe:operation math:greaterThan ;
    rdfs:subClassOf ewe:Parameter ;
    rdfs:domain ewe:EmotionAwareElearningSystem .
```

Bibliography

[1] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials*, 17(4):2347–2376, 2015.

[2] Kimberly E Arnold and Matthew D Pistilli. Course signals at purdue: Using learning analytics to increase student success. In *Proceedings of the 2nd international conference on learning analytics and knowledge*, pages 267–270. ACM, 2012.

[3] Yadid Ayzenberg, Javier Hernandez Rivera, and Rosalind Picard. Feel: frequent eda and event logging–a mobile social interaction stress monitoring system. In *CHI'12 extended abstracts on human factors in computing systems*, pages 2357–2362. ACM, 2012.

[4] David Beckett, Tim Berners-Lee, Eric Prud'hommeaux, and Gavin Carothers. Rdf 1.1 turtle. *World Wide Web Consortium*, 2014.

[5] Ishan Behoora and Conrad S. Tucker. Machine learning classification of design team members' body language patterns for real time emotional state detection. *Design Studies*, 39:100 – 127, 2015.

[6] Tim Berners-Lee and Dan Connolly. Notation3 (n3): A readable rdf syntax. *W3C team submission*, 28, 2011.

[7] Damian Borth, Rongrong Ji, Tao Chen, Thomas Breuel, and Shih-Fu Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 223–232. ACM, 2013.

[8] John P Campbell, Peter B DeBlois, and Diana G Oblinger. Academic analytics: A new tool for a new era. *EDUCAUSE review*, 42(4):40, 2007.

[9] Mohamed Amine Chatti, Anna Lea Dyckhoff, Ulrik Schroeder, and Hendrik Thüs. A reference model for learning analytics. *International Journal of Technology Enhanced Learning*, 4(5-6):318–331, 2013.

[10] M. Coronado. EWE Ontology Specification, 2013. http://www.gsi.dit.upm.es/ontologies/ewe.

[11] Miguel Coronado and Carlos A Iglesias. Task automation services: Automation for the masses. *IEEE Internet Computing*, 20(1):52–58, 2016.

[12] Crossbar.io. Crossbar.io, Networking for apps. https://crossbar.io/.

[13] Mongo DB. MongoDB: Open Source Document Database. https://www.mongodb.com/en.

[14] DB-Engines. DB-Engines Ranking - popularity ranking of database management systems. https://db-engines.com/en/ranking.

[15] Ben De Meester, Dörthe Arndt, Pieter Bonte, Jabran Bhatti, Wim Dereuddre, Ruben Verborgh, Femke Ongenae, Filip De Turck, Erik Mannens, and Rik Van de Walle. Event-driven rule-based reasoning using eye. In *SSN-TC/OrdRing@ ISWC*, pages 75–86, 2015.

[16] Leony Derick, Gayane Sedrakyan, Pedro J Munoz-Merino, Carlos Delgado Kloos, and Katrien Verbert. Evaluating emotion visualizations using affectvis, an affect-aware dashboard for students. *Journal of Research in Innovative Teaching & Learning*, 10(2):107–125, 2017.

[17] Elastic. Elastic Search: RESTful, Distributed Search Analytics. https://www.elastic.co/products/elasticsearch.

[18] Tanya Elias. Learning analytics. *Learning*, pages 1–22, 2011.

[19] Empatica. Real Time Physiological Signals, E4 EDA/GSR Sensor. https://www.empatica.com/en-eu/research/e4/.

[20] Mohamed Ez-Zaouia and Elise Lavoué. Emoda: a tutor oriented multimodal and contextual emotional dashboard. In *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*, pages 429–438. ACM, 2017.

[21] M Feidakis. A review of emotion-aware systems for e-learning in virtual environments. In *Formative Assessment, Learning Data Analytics and Gamification*, pages 217–242. Elsevier, 2016.

[22] Antonio Fernández Llamas. Design and implementation of a Semantic Task Automation Rule Framework for Android Devices. Master's thesis, ETSI Telecomunicación, Universidad Politécnica de Madrid, feb 2016.

[23] Antonio Fernández-Caballero, Arturo Martínez-Rodrigo, José Manuel Pastor, José Carlos Castillo, Elena Lozano-Monasor, María T. López, Roberto Zangróniz, José Miguel Latorre, and Alicia Fernández-Sotos. Smart environment architecture for emotion detection and regulation. *Journal of Biomedical Informatics*, 64:55 – 73, 2016.

[24] Luis Ferreira, Antonio Neves, Artur Pereira, Eurico Pedrosa, and Joao Cunha. Human detection and tracking using a kinect camera for an autonomous service robot. *Advances in Aritificial Intelligence-Local Proceedings, EPIA*, pages 276–288, 2013.

[25] Ángel Fidalgo-Blanco, María Luisa Sein-Echaluce, Francisco J García-Peñalvo, and Miguel Ángel Conde. Using learning analytics to improve teamwork assessment. *Computers in Human Behavior*, 47:149–156, 2015.

[26] Reza GhasemAghaei, Ali Arya, and Robert Biddle. A dashboard for affective e-learning: data visualization for monitoring online learner emotions. In *EdMedia+ Innovate Learning*, pages 1536–1543. Association for the Advancement of Computing in Education (AACE), 2016.

[27] Philip J Goldstein and Richard N Katz. *Academic analytics: The uses of management information and technology in higher education*, volume 8. Educause, 2005.

[28] Explore Group. The Most Popular Databases 2018. https://www.explore-group.com/us/blog/the-most-popular-databases-2018/bp46/.

[29] Yuvraj Gupta. *Kibana Essentials.* Packt Publishing Ltd, 2015.

[30] IBM. Ibm Watson Services: Tone Analyzer. https://www.ibm.com/watson/services/tone-analyzer/.

[31] Apache Jena. Apache Jena Fuseki. https://jena.apache.org/documentation/fuseki2/.

[32] John Kelly III and Steve Hamm. *Smart Machines: IBM Watson and the Era of Cognitive Computing.* Columbia University Press, 2013.

[33] Barry Kort, Rob Reilly, and Rosalind W Picard. An affective model of interplay between emotions and learning: Reengineering educational pedagogy-building a learning companion. In *Advanced Learning Technologies, 2001. Proceedings. IEEE International Conference on*, pages 43–46. IEEE, 2001.

[34] Pablo Viñals Lage. Design and Implementation of a Messaging Module for Smart Space Automation compatible with WAMP and MQTT Protocols. Tfg, ETSIT, Mandrid, June 2018.

[35] Dolores Lerís, Ángel Fidalgo, and M Luisa Sein-Echaluce. A comprehensive training model of the teamwork competence. *International Journal of Learning and Intellectual Capital 14*, 11(1):1–19, 2014.

[36] Carlos Marzal Romón. Analysis and Development of a Personal Dashboard for Emotion Tracking. Tfg, ETSI Telecomunicación, January 2019.

[37] Wes McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython.* " O'Reilly Media, Inc.", 2012.

[38] Microsoft. Microsoft Azure Cognitive Services: Face Api. https://docs.microsoft.com/es-es/azure/cognitive-services/face/.

[39] Joi L Moore, Camille Dickson-Deane, and Krista Galyen. e-learning, online learning, and distance learning environments: Are they the same? *The Internet and Higher Education*, 14(2):129–135, 2011.

[40] Carlos Moro García. Development of a Task Automation Platform based on Semantic Multievent-driven rules. Tfg, Universidad Politécnica de Madrid, June 2017.

[41] Sergio Muñoz López. Development of a Task Automation Platform for Beacon enabled Smart Homes. Master's thesis, Universidad Politécnica de Madrid, feb 2016.

[42] Sergio Muñoz López. Development of an Emotion Aware Ambient Intelligent System for Smart Offices. Application in a Living Lab and in a Social Simulated Scenario. Tfm, Universidad Politécnica de Madrid, July 2017.

[43] Sergio Muñoz López, Oscar Araque, J. Fernando Sánchez-Rada, and Carlos A. Iglesias. An Emotion Aware Task Automation Architecture Based on Semantic Technologies for Smart Offices. *Sensors*, 18(5):1499, May 2018.

[44] Sergio Muñoz López, Antonio Fernández, Miguel Coronado, and Carlos A. Iglesias. *Smart Office Automation based on Semantic Event-Driven Rules*, volume 21 of *Ambient Intelligence and Smart Environments*, pages 33–42. IOS Press, September 2016.

[45] Sergio Muñoz López. Development of a Task Automation Platform for Beacon Enabled Smart Homes. Master's thesis, ETSI Telecomunicación, Universidad Politécnica de Madrid, jan 2016.

[46] Donald Norris, Linda Baer, and Michael Offerman. A national agenda for action analytics. In *National Symposium on Action Analytics*, pages 21–23, 2009.

[47] Moodle Org. Moodle. https://moodle.org/?lang=en.

[48] Mohammed Ouadoud, Mohamed Yassin Chkouri, Amel Nejjari, and Kamal Eddine El Kadiri. Studying and comparing the free e-learning platforms. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pages 581–586. IEEE, 2016.

[49] Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediyana Daskalova, Jeff Huang, and James Hays. Webgazer: Scalable webcam eye tracking using user interactions. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3839–3845. AAAI, 2016.

[50] Zachary A Pardos, Ryan SJD Baker, Maria OCZ San Pedro, Sujith M Gowda, and Supreeth M Gowda. Affective states and state tests: Investigating how affect and engagement during the school year predict end-of-year learning outcomes. *Journal of Learning Analytics*, 1(1):107–128, 2014.

[51] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[52] Paul R Pintrich. A motivational science perspective on the role of student motivation in learning and teaching contexts. *Journal of educational Psychology*, 95(4):667, 2003.

[53] H. Qu and Q. Chen. Visual analytics for mooc data. *IEEE Computer Graphics and Applications*, 35(6):69–75, Nov.-Dec. 2015.

[54] Thanyalak Rattanasawad, Kanda Runapongsa Saikaew, Marut Buranarach, and Thepchai Supnithi. A review and comparison of rule languages and rule-based inference engines for the semantic web. In *2013 International Computer Science and Engineering Conference (ICSEC)*, pages 1–6. IEEE, 2013.

[55] Tarmo Robal, Yue Zhao, Christoph Lofi, and Claudia Hauff. Webcam-based attention tracking in online learning: A feasibility study. In *23rd International Conference on Intelligent User Interfaces*, pages 189–197. ACM, 2018.

[56] Samara Ruiz, Sven Charleer, Maite Urretavizcaya, Joris Klerkx, Isabel Fernández-Castro, and Erik Duval. Supporting learning by considering emotions: tracking and visualization a case study. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, pages 254–263. ACM, 2016.

[57] Samara Ruiz, Maite Urretavizcaya, Isabel Fernández-Castro, and Juan-Miguel López-Gil. Visualizing students' performance in the classroom: Towards effective f2f interaction modelling. In *Design for Teaching and Learning in a Networked World*, pages 630–633. Springer, 2015.

[58] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[59] J. Fernando Sánchez-Rada, Carlos A. Iglesias, Ignacio Corcuera-Platas, and Oscar Araque. Senpy: A Pragmatic Linked Sentiment Analysis Framework. In *Proceedings DSAA 2016 Special Track on Emotion and Sentiment in Intelligent Systems and Big Social Data Analysis (SentISData)*, pages 735–742, Montreal, Canada, October 2016. IEEE.

[60] Enrique Sánchez Tolbaños. Design and Development of a Cognitive Computing em- powered Robot Assistant for Semantic Task Automation in Smart Spaces. Tfg, Universidad Politécnica de Madrid, June 2017.

[61] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Face alignment through subspace constrained mean-shifts. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1034–1041. IEEE, 2009.

[62] Cornelia Setz, Bert Arnrich, Johannes Schumm, Roberto La Marca, Gerhard Tröster, and Ulrike Ehlert. Discriminating stress from cognitive load using a wearable eda device. *IEEE Transactions on information technology in biomedicine*, 14(2):410–417, 2010.

[63] George Siemens. What are learning analytics. *Retrieved March*, 10:2011, 2010.

[64] Thomas Smith and John Guild. The cie colorimetric standards and their use. *Transactions of the optical society*, 33(3):73, 1931.

[65] Wout Swinkels, Luc Claesen, and Haibin Shen. Camera-based real-time emotion classification using support-vector machines. 2017.

[66] Robert Sylwester. How emotions affect learning. *Educational leadership*, 52(2):60–65, 1994.

[67] W3C Team. RDF/XML Syntax Specification, 2004. https://www.w3.org/TR/REC-rdf-syntax/.

[68] W3C Team. Notation3 (N3): a readable RDF syntax, 2011. https://www.w3.org/TeamSubmission/n3/.

[69] EverSQL Tomer Shay. Most Popular Databases in 2018 according to StackOverflow Survey. https://www.eversql.com/most-popular-databases-in-2018-according-to-stackoverflow-survey/.

[70] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22, 2011.

[71] Beyond Verbal. Beyond Verbal Emotion AI. http://beyondverbal.com/api/.

[72] Sara Värlander. The role of students' emotions in formal feedback situations. *Teaching in Higher Education*, 13(2):145–156, 2008.

[73] K. Wang, N. An, B. N. Li, Y. Zhang, and L. Li. Speech emotion recognition using fourier parameters. *IEEE Transactions on Affective Computing*, 6(1):69–75, Jan 2015.

[74] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2016.

[75] Audun Mathias Øygard. clmtrackr.js - Javascript library for fitting facial models to faces in images and video. https://www.auduno.com/clmtrackr/docs/reference.html.