

# Visualizing Historical Patterns in Large Educational Datasets

*DRAFT VERSION*

Tiago Martins

INESC-ID and Instituto Superior Técnico, Universidade de Lisboa  
Lisboa, Portugal  
tiago.a.martins@tecnico.ulisboa.pt

Daniel Gonçalves

INESC-ID and Instituto Superior Técnico, Universidade de Lisboa  
Lisboa, Portugal  
daniel.goncalves@inesc-id.pt

Sandra Gama

INESC-ID and Instituto Superior Técnico, Universidade de Lisboa  
Lisboa, Portugal  
sandra.gama@tecnico.ulisboa.pt

## INTRODUCTION

Education levels have been increasing across the world. This is true for traditional learning in brick-and-mortar schools and colleges, and even more so given the rise of new technologies in distance education such as Massive Online Open Courses (MOOC), supported by learning management systems (LMS) and course management systems (CMS). The number of students has, thus increased throughout the years and with it, the need for new tools to be able to understand of effective the learning experience is.

Platforms such as CMS and LMS allow the easy storage of records related to student grades, attendance in classes and approvals in a course. Analyzing this data may allow the detection of students with problems or even issues with the content and structure of the courses themselves (Baepler 2010). The analysis of this data can not only allow a deeper understanding of eventual problems, but also helps alleviate the problems of a reality where students and professors are less knowledgeable of one another, either due to in-presence teaching massification or in on-line, distance learning settings, which make it difficult to perceive their problems and to track down their causes (May 2011). This can lead to deeper reflections about the contents being taught and the development of strategies to minimize failure and lower dropout rates

Given the amount of data that these systems collect over multiple years, analyzing it may be a very difficult task and require high cognitive effort. A possible solution that allows the understanding and identification of relevant patterns in data is Information Visualization (Desai

2014). By using interactive Information Visualization (InfoVis), one can explore different facets of the data and cope with the high complexity of very large datasets by using filtering and selection mechanisms. The use of derived measures that embody higher-level trends can provide precious different perspectives from the original information (Mazza 2009).

We propose a solution where InfoVis is used to represent large datasets of academic performance data. A major challenge in this regard is scalability. While most previous approaches, as described in the Related Work section, deal with data from a single course or degree, we want to be able to address the problem at the school level. As such, we devised a set of minimalistic views which, while sacrificing some more advanced analysis, make up for that shortcoming with the possibility for long-term analysis of all degrees and courses of an entire university-level school. An additional hurdle, the heterogeneity of curricula between degrees and even inside the same degree through time (as curricular restructuring takes place) also contribute to make this a challenging domain.

As a proof of concept, we used the data from Instituto Superior Técnico (IST), the engineering school of the University of Lisbon, encompassing fourteen years and all undergraduate and MSc level degrees. To that end, we used official information sources made available through the school's information system, based on the FenixEdu platform that provides an API for certain data to be extracted.

Given the large dataset, studying the evolution of a degree or course over the years or discovering patterns and trends in the data could still become a time-consuming process without prior data processing (Ali 2012). Thus, to make this data amenable to visualization, the entire dataset was pre-processed, cleaned and re-structured, as will be described below. The resulting dataset, with several thousand entries, was then successfully visualized by our solution, allowing interesting patterns to be found, as shown by usability and usefulness tests, performed with the help of IST's Statistics and Prospective Unit.

We, thus, developed a system that allows the analysis of the academic path of students and the evolution of degrees and courses over the years. This system may help professors, course coordinators and the School itself with assessing relevant issues that can be timely solved and improve the teaching-learning process.

In section two we explain the different works developed in the area, each exploring different mechanisms and academic data. The problems and strengths related to these works will also be discussed. In section three is presented the earlier phase of the implementation of our solution, which consists in the conception of the possible questions to which the visualization will answer and the description of the entire extraction process of the data. In the next section, we describe all the implementation phase, including the design of non-functional prototypes, the structure of the application and backend and frontend development. In the fifth section, the method used to evaluate the tool will be described. In section six the conclusion of the work is presented.

## RELATED WORK

The increase in the number of students in traditional and online education has provided a large amount of educational data with the introduction of learning systems. Analyzing student behavior and the evolution of courses and degrees has become possible, even in the face of a deluge of information.

Considering the analysis of the academic data, such as the academic path and the evolution of the courses and degrees, there was a need to develop applications that allow the teachers and the coordinators to keep improving the students' performance and overcome the academic failure. However, due to the increasing number of students in online teaching and the amount of data generated by them, it is hard to keep track of the students' difficulties and to help them overcome these issues, based on unprocessed data.

Considering this scenario MOOCViz (Dernoncourt 2013) was created, which provides visual techniques that may help the teachers to evaluate the performance of their students. The visual representations used are the choropleth and the stacked bar chart. Choropleth summarizes the certificates delivered in all countries where the students are and it is represented by a white (fewer certificates) to red (more certificates) scale. The other representation allows the user to see a more detailed information, regarding the time dedicated to the degree. In this case, it's represented the access to books, exams, problems, lectures and tutorials. This allows the teachers to get a detailed view of what kind of content is visited by each country. However, due to the lack of interaction in both idioms, it's hard to explore all the available information.

Another application (Gan 2009) was developed to follow students' paths across all the possible evaluations. Data from an English test of college students was collected and as an alternative to visual representations of bar and pie charts, parallel coordinates were used, to represent multiple dimensions. Each of those represents a different evaluation, where it is possible to track the evolution of the initial grade, until the final one. To avoid the visual confusion of the huge amount of gray lines, a range of colors (yellow to blue) was applied to distinguish the good grades from the bad ones. However, analyzing only one visual representation may not be enough, to get a detailed representation of the data, a box plot was implemented to analyze its distribution.

The solution described in (Weiand 2015) uses data of three years of the Faculdade Cenecista of Osório, which has information about the resources that students use, for instance, forums, web pages and mini-tests. To analyze data related to multiple students, a table lens was developed, so it is possible to compare all the different content accessed by each student and to represent their tendency to finish the degree. This attribute allows representing the students that might fail (green), that tend to give up (red) and those that might pass (blue). However, the colors used to represent this attribute, aren't intuitive at all since green was used to represent student failure, which might mislead the users. However, having a big number of students represented in the table lens, implies an equal number of bars, which may difficult its analysis without a proper filter mechanism. This application has also two more visual representations that are connected that work as overviews of the entire course. In the pie chart, is represented the number of students that accessed each resource and in the stacked bar chart is possible to see what accesses were made in a selected period. When the user checks some information in one of the views, the other is highlighted. When the period is changed, all the views are updated.

To allow the user to interact with the system, several types of mechanisms were developed, like the selection of each bar in the table lens to get more information about each student or even the ability to draw a rectangle to filter the dates present on the stacked bar chart. It is also possible to search for a specific student or a resource.

TelEduc (Da Silva 2007) is a Learning Management System also provides a visualization of student interaction with the system, using mostly matrixes and barcharts, with a focus on the analysis of student participation in a course's discussion forum. Unlike TelEduc, GISMO (Mazza 2004) allows the monitoring of student performance throughout an entire semester, for several courses. It uses more sophisticated visualizations such as Cleveland Dot plots, but is lacking in terms of interaction. CourseVis (Mazza 2007) also goes beyond traditional visualization techniques but, alas, idioms such as a 3D bubble chart hinder the accurate perception of the data.

Based on the analysis of the different works, we see that it is possible to extract relevant information about the visualization idioms used, such as the usage of filtering mechanisms and the selection of elements, to reduce the amount of visible data. Alas, existing work concentrates on fairly small datasets, and are not amenable to the analysis of years or decades of data for an entire school.

## **LEARNING WHAT TO VISUALIZE**

With the high amount of data available today, it is necessary to think about the most relevant and interesting questions that may be answered using it. Considering the attributes mentioned in the related work, such as approvals, grades, class attendance, content visited and others, an initial set of domain-specific questions was created, covering all the aspects considered important for a visualization of this nature. To evaluate some possible questions, we produced a questionnaire that was answered by 57 people from IST. Each question was rated on a 5-point Likert scale according to how interesting and relevant it was considered to be.

Additionally the survey contained a write-in question for users to suggest new questions. The most frequently and consistently mentioned was "Is there a correlation between gender and passing certain courses?"

All questions (suggested by us and write-ins) were then analyzed by people from IST's Statistics and Prospective Unit (SPU). This Unit is the one responsible for the analysis of educational and academic data in the school. It handles CUQ surveys, collects enrollment and academic success data, and regularly produces reports used by the school's management. This validation was important as the SPU would be one of the potential users of our solution, and have a strong and well-supported idea of what really is important. We were also able to better understand their tools and methods finding the most relevant questions that served as the basis for the development of our solution:

- Is there any correlation between in the percentage of passing students in a course with the professors teaching that course?
- What is the degree for which a common course (offered to several degrees) has had more passing students?

- What is the course with the worst approval rating of a degree?
- Is there any correlation between the percentage of passing students and the course's rating in the CUQs?
- Which degree has the highest average number of passing students?

These were the basis for the development of our solution.

## **COLLECTING EDUCATIONAL DATA**

The first step in any work that intends to allow the analysis of educational data is gathering such data in a way in which it can be manipulated and visualized in meaningful ways. As a case study, we collected a dataset from Instituto Superior Técnico, referring to all courses and degrees since 2002, made available using a developed API called FenixEdu. From it, we can access the following information: list of degrees; list of courses for each degree; list of all students enrolled in each course; professors teaching each course; ECTS (European Credit Transfer and Accumulation System) credits for each course; evaluation method and dates; class schedule; student work group rules and dates; number of students per group. Other information, such as demographic information and grades is also available, but only for authorized users, after login (Barata 2013). Given its sensitivity, that data was not considered in our system. We complemented this dataset with data from the CUQ (“Qualidade de Unidades Curriculares” or Curricular Unit Quality Questionnaire), that gives us an assessment of the perceived quality of courses and teaching for each course.

### **Data Extraction**

A Python script was created to access the FenixEdu API. Starting with the lists of degrees, we performed all requests necessary to obtain all possible data for all courses in IST since 2002, the year this system came online. The API responded with data in JSON format, stored locally and managed such as to prevent repeated requests. After analyzing the information that people would like to know, it was necessary to understand which attributes could be retrieved from the learning management system.

Data from the FenixEdu API was complemented by data from the CUQ reports. These reports stem from data entered by students in a form after each semester finishes. There, they are asked to rate both the courses they attended and the professors that taught them according to several criteria, such as perceived usefulness, teaching quality and workload, among others. More importantly, this is then used to compute an overall rating for the course and professor (for each taught course), which summarizes the student's perception of their quality. Information on the percentage of passing students to each course is also present, as is the grade distribution for that course (how many students had each possible grade in the course).

Here, we do not have access to the raw data, but IST's Statistics and Prospective Unit produces an HTML report for all courses and professors that can be accessed from the courses' page. With these reports, it's possible to obtain data after 2010. To gather this information, a Python script

was created to simulate browser navigation in IST's website, to traverse the pages for all courses and, from there access the CUQ reports. These were then scraped for the relevant information.

Overall close to 2 GBs of data were collected from both sources over the course of one week (mostly due to limit rates imposed by the online services). This data in itself, however, required some processing before it could be visualized. This processing, described below, comprised the inferring of curricular continuity in the face of degree restructuring, and complementing it with data from additional sources.

## Data Pre-Processing

One of the problems to be solved had to do with the lack of CUQ data before 2010, namely that regarding the percentage of passing students to each course. We completed that information based on FenixEdu data. Alas, we do not have grades, in the dataset, so this had to be heuristically estimated: students enrolled to one course in one year and that are never again enrolled in the same course in the future are considered to have passed, *if* they, overall, enrolled in all courses that would allow them to finish the degree (otherwise he may have simply dropped out).

This quickly led to another problem, stemming from the curricular restructuring of degrees. Often, courses cease to exist, to be replaced by others, or may continue to exist but with a new name. In order to be able to follow the same course throughout the years, we needed to establish a connection between those different versions of the same course. To estimate student drop-out rates, we had to estimate different academic paths students caught in restructurings could have followed. This was done manually, by using historical information available in the courses' web pages. The process was heuristic. In fact, many individual situations are handled on a per case basis by the courses' coordinators, so it was impossible to devise a rule that would automatically allow us to find all such cases. Having said this, most students do follow default equivalence paths and those were taken into account. Even if this fails for a small number of students, since our goal is to study the overall results and trends for courses and degrees, not individual students, we are confident our dataset embodies all relevant high-level trends.

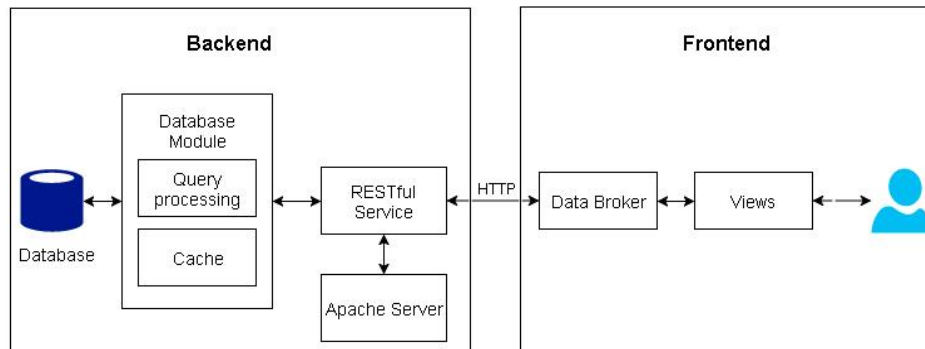
In the end, we have a dataset for 14 years of courses, degrees and professors, with data about 86 degrees and 14,656 courses. The available attributes to use are:

- Approvals, grades and evaluation of the CUQ of each course and degree (since 2010)
- Courses' ECTS credits
- Number of student enrollments in each course and degree
- Teachers' CUQ evaluations
- Average of Teachers' CUQ evaluations in each degree

## TECNICOVIS

The focus of this solution is to allow a detailed analysis of several attributes, with emphasis on patterns in the evolution of degrees and courses. Through the combination of several visual representations, this exploration becomes possible.

The implementation was divided into two components, the backend and the frontend represented in Figure 1.



*Figure 1 – Architecture*

The technologies used to implement the backend were a MySQL database, an Apache server, a data base module and a RESTful service written in Python. Regarding the frontend, the main technology used for the development of the customizable visual representations is the D3.js framework. This tool allows to manipulate elements in HTML and SVG and to assign styles in CSS, giving freedom to the development without using previously generated representations (Murray 2013).

### Backend

Regarding the large amount of data collected, all the processed information was stored in a MySQL database, which contains the tables Degrees, Courses, and Teachers. In each table, there is the respective numerical identifier, the name, acronym, the CUQ of professors and courses, grades, year, term, degree type and students number.

To make all these data accessible to the application, a module responsible for the processing of the queries was developed and a RESTful service that will allow the application to make the requests. The module that connects to the database has a cache mechanism that allows storing the data of the last twenty requests for each connected IP address, which proved sufficient for good performance. Regarding the RESTful service, several endpoints were created to retrieve the data, namely lists of degrees and courses, lists of courses in a particular degree, lists of professors for a particular course, and detailed information for each degree, course and professor. This will be invoked as needed by the frontend, as the user expands the visualization and more detailed information is required.





Figure 2 – Filtering mechanism

In a first iteration, the matrix was developed with all the courses represented on it. However, as mentioned above, we have information for 14,656 courses, making this approach unfeasible. Thus, only degrees are represented in the initial state. To see the courses, the user must click on a degree making the visualization expand to reveal the courses taught in it. Clicking again, will collapse and hide all the corresponding courses. Only when expanding a degree will the backend be requested for detailed information for the courses therein.

The cells referring to the degrees encode the average of the approvals of all courses, to keep the view consistent. To allow a quick comparison between degrees or courses, an extra cell was added to the visualization, immediately to the right of the degree's (or courses') name, colored according to the average of all semesters' approvals.

Although the degree expansion mechanism allows the drastic reduction of the number of visible rows in the matrix, analyzing a specific row may be a hard task, amidst all others. To alleviate that problem, a filtering mechanism was developed, which consists of using a selection rectangle as also depicted in Figure 2: selecting, through click-and-drag, a subset of degrees, will cause all others to disappear, allowing the analyst to focus on that particular selection. Furthermore, if instead of degrees the user selects courses, the visualization will display only degrees to which those courses are offered, and, inside them, the aforementioned courses (Figure 3). Courses and degrees can also be searched for using a search box. Current filtering constraints are displayed in the form of pills.

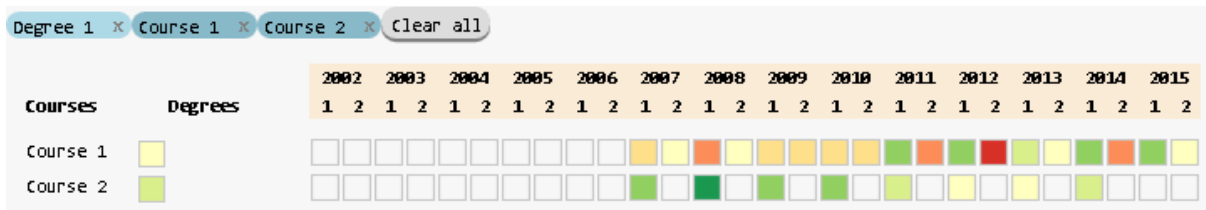


Figure 3 – Filtering

To acquire more information regarding each cell of the matrix, a mouse over mechanism was implemented where it is possible to view the exact value of the approvals in that semester when the cursor overlaps the desired cell. A highlight mechanism was also implemented in each cell. The cell's border is changed from gray to black, and the row and column are highlighted in blue. This makes it easier to read the year, semester, degree and course selected (Figure 4).

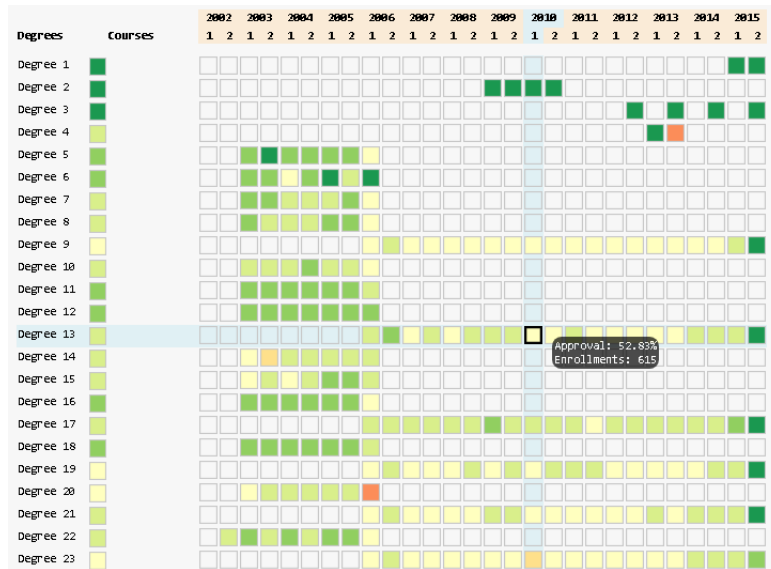


Figure 4 – Cell highlighting

Another interaction mechanism consists in the possibility of dragging a degree or course into a new position in the matrix, to allow the direct comparison between two rows. Finally, a pop-up menu allows the choice of attribute represented in the heat map: percentage of passing students (default); mean student grades and CUQ ratings.

### Scatterplot

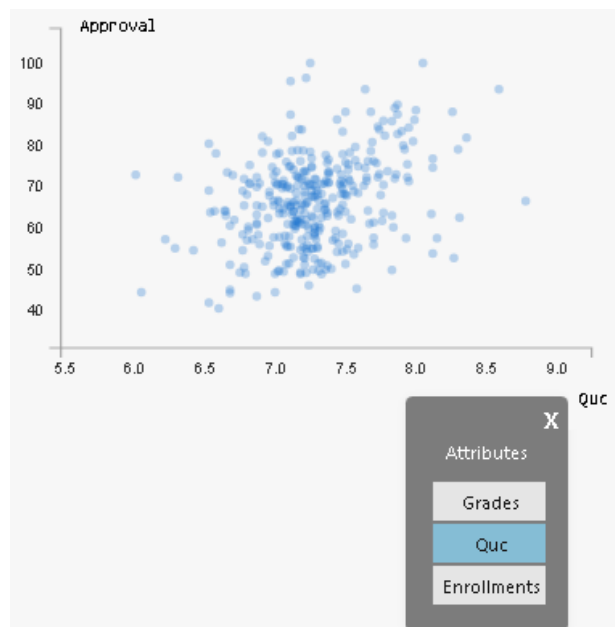


Figure 5 – Scatterplot

The matrix view does not allow the analysis of possible correlations between the several attributes. To that end, we created a scatterplot (Figure 5). The scatterplot is located to the right of the matrix and displays one dot for each course in the current selection. To minimize occlusion problems, those circles have only 30% opacity. Thus, we can also estimate how concentrated certain values are by looking for differences in saturation: more overlapping dots will result in more saturated areas. The axes automatically scale for the range of attribute values to display. The scatterplot supports zooming and panning for a detailed inspection of its contents. Furthermore, clicking on individual dots allows the selection of the corresponding courses. Either the horizontal or vertical axis can display any of the three attributes mentioned in the previous paragraph.

## Line chart and Sparkline

To represent the teachers' CUQ evaluation, it was necessary to implement a third representation. Initially, we explored the possibility to display multiple sparklines in the area below the scatterplot. However, for scalability, we integrated them into the matrix instead (Figure 6). They provide a summary of an expanded degree, where it is possible to visualize the mean of the CUQ ratings for all professors in all courses in this degree over the years. Besides being possible to analyze the slopes in this line, points were added to help with its interpretation, and once more hovering the mouse over those points will yield additional information. An attribute selection mechanism identical to the scatter plot was also added. The user can choose to analyze the teachers' CUQ and the total number of enrollments in this degree.



*Figure 6 – Sparkline*

While these sparklines allow the analysis of a single degree, comparison across degrees can be hard if they are too far apart. This may happen when a degree is expanded, increasing the number of rows in the visualization (one for each course). To alleviate this problem, a drag icon has been added, which allows moving the sparkline to the area below the scatterplot, turning it into a line chart. Several sparklines can be dragged there, resulting in different lines in the chart, thus facilitating comparisons (Figure 7).

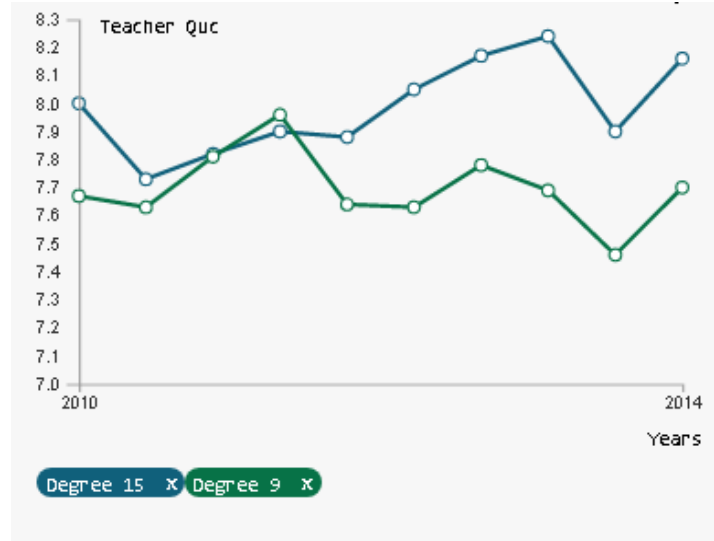


Figure 7 – Line chart

It was necessary to assign a color to each of the lines, so they could be distinguished. Whenever a degree is expanded and the sparkline is drawn, a random color is assigned to that degree that is recorded in an object, to ensure that there are no duplicate colors. However, to ensure they are visually distinct a formula was used to calculate the distance between two colors. Initially, it was considered applying the Euclidean distance to each color present in the object with the RGB color space. However, since RGB isn't perceptually uniform, results were sub-optimal. Thus, we opted for the use of CIE L\*a\*b\* instead. The first formula developed to calculate the distance between two colors is the CIELAB Delta, which consists of applying the Euclidean distance to two arrays, with the following equation:

$$\Delta E = \sqrt{(L_2^* - L_1^*)^2 + (a_2^* - a_1^*)^2 + (b_2^* - b_1^*)^2}$$

However, to improve the above formula, the factors  $K_L$ ,  $K_C$ ,  $K_H$  (factors of perception of the background conditions),  $S_L$ ,  $S_C$  and  $S_H$  (linear functions of the chromatic differences) were attributed. The values used for  $K_L$ ,  $K_C$ ,  $K_H$ ,  $S_L$  are one, while  $S_C$  and  $S_H$  have the value  $1 + 0.045 * (a_2 + a_1)$  and  $1 + 0.015 * (b_2 + b_1)$ , respectively (Plataniotis 2000). The formula used to implement the difference between colors is:

$$\Delta E_{ab94}^* = \sqrt{\frac{(L_2^* - L_1^*)^2}{K_L S_L} + \frac{(a_2^* - a_1^*)^2}{K_C S_C} + \frac{(b_2^* - b_1^*)^2}{K_H S_H}}$$

To use this equation, it was necessary to convert the RGB colors to the color space L \* a \* b \* and it is defined that values for  $\Delta E$  greater than five, allow distinguishing if two colors are different (Plataniotis 2000). To identify the rows that have been dragged into the comparison idiom, pills with the name of the selected item and the corresponding color are added. It is only possible to draw a total of five lines, to avoid a high cognitive effort in the analysis of the graph. Like in the

sparkline, it is also possible to select the attribute and use the mouse over mechanism in each circle.

## Bar Chart

The idiom chosen to represent, when expanding a course, the detailed information related to professors teaching that course, was a small multiples bar chart, where the height of the bar encodes the value of the professor's CUQ rating, as represented in Figure 8. Color was added as second encoding to improve understandability. To keep consistency with the matrix, the same color scale was used for the values of the evaluations of each professor. This view also allows using mouse over in each bar, to obtain more information.

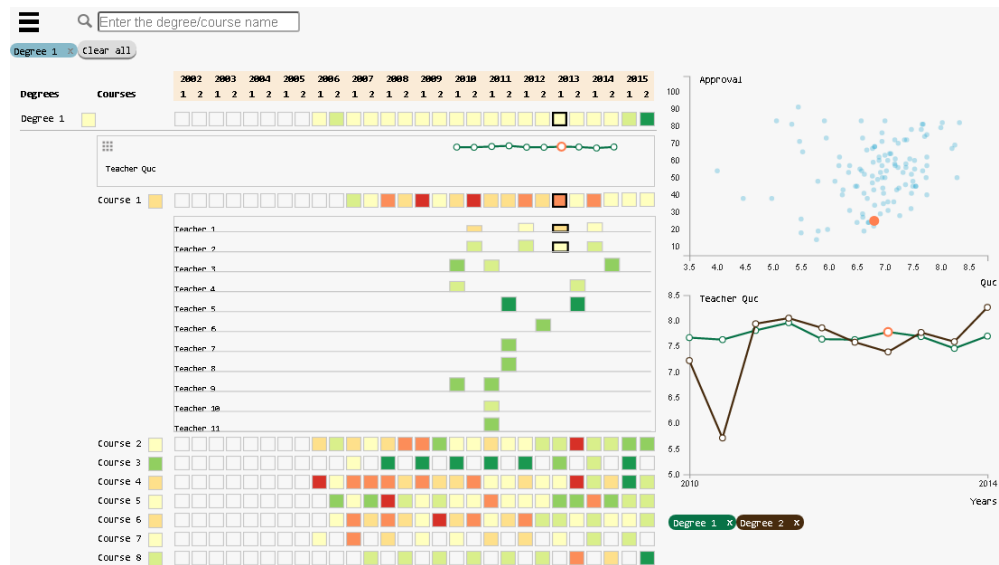


Figure 8 – Selection of an instance of a degree and a course

## Interconnectivity between the visual representations

To make possible to identify the items selected from the implemented views, a requirement was previously established that consisted of highlighting all selections. To do this, when the user overlaps the cursor in each element of the different views, it is reflected in the others. In addition, a selection mechanism has been developed that allows isolating the loaded instance with the left mouse button (Figure 8).

## EVALUATION

To evaluate the application, two types of tests were performed. The first was a usability test with regular users, to understand if the system is easy to use and understand. The second tests took a form of a case study carried out with the Statistics and Prospective Unit of IST, where technicians were given the freedom to explore the entire application. Here, we tried to assess the usefulness of the system, and validity of the patterns and conclusions it can yield.

## Usability Test

Experimental tests were conducted in a controlled environment. Twenty users participated, one at a time, in the tests. The material used to perform the test was a computer to access the application. Each test had a thirty minute duration and started with a brief demonstration of the visualization and interaction mechanisms. Before starting the tasks, a trial period of about five minutes was carried out, in order to allow user adaptation to the tool and minimize biases due to a first interaction with the system.

For each task, we collected the number of errors; task completion time; and task completion. After completing all the tasks, each user answered to a satisfaction form (System Usability Scale), to evaluate the usability of the developed system (Brooke 2013). To explore the different mechanisms, the following tasks were performed:

1. *What was the degree with the highest approval average in all courses in 2008?*
2. *What is the degree where the AL course had more approvals?*
3. *What is the course with the worst grades average over the years of the LEMat degree?*
4. *Has there been any improvement in ACED course approvals in the LERC degree after a change of teachers?*
5. *Does the LEQ teacher's average CUQ tend to increase according to the percentage of approvals?*
6. *Of all graduations, which have more than 4000 enrollments and grades greater than 13.*
7. *Although LEIC-A has approximately twice as many LEIC-T students, do the courses' grades have big changes?*
8. *Has the number of teachers in the PEST course in LEMAT influenced the percentage of approvals?*

These tasks stem from those deemed relevant during the initial questionnaire (and for which the necessary data was available). They were also chosen so that we could assess if the users understood all the different developed idioms.

## Result Analysis

Based on the data in Table 1, it is possible to verify that both Task 6 and 7 obtained a high standard deviation in the time used to perform them. They also have a longer average time than the rest. The main discrepancy is due to the lack of answer at the end of the four minutes of the task. However, it is also verified that task 5 has an equal number of errors because most users could finish it, but with wrong answers.

Tasks	Time spent (seconds)		Number of errors	
	Average	St. Dev	Average	St. Dev
Task 1	17.25	17.19	0.05	0.22
Task 2	29.8	13.07	0	0
Task 3	28.95	7.91	0.2	0.41
Task 4	36.8	11.21	0	0
Task 5	38.8	9.43	0.55	1.10
Task 6	105.5	64.65	1.05	1.85
Task 7	55,95	49.84	0.25	0.91
Task 8	32.1	9.20	0	0

*Table 1 – Task completion time and number of errors*

During the execution of Tasks 6 and 7, users who could not complete them, had some troubles while perceiving how to change the attributes in question. Although a visual hint was implemented when the mouse cursor is moved to a label, the users did not realize that the axis label is also a button and could be clicked. In Task 6, many users made the mistake of searching in the search bar for the type of the degree and only then realized that this mechanism only makes it possible to search for acronyms.

Despite these errors, only one task has an average of over 1 error, which shows that most users understood the developed application. As for time, it should be noted that only one task exceeded 60 seconds, which reveals that users needed less than a minute to perform a task that could take much longer by analyzing multiple graphs or tables without interactive mechanisms.

To prove this scenario for a broader population, statistical calculations were performed, in order to perceive the confidence interval relative to the mean. For this, a confidence interval of 95% was used. We got, for the different tasks, the following intervals: IC\_T1 = [9.21; 25.29]; IC\_T2 = [23.68, 35.92]; IC\_T3 = [25.25; 32.65]; IC\_T4 = [31.55; 42.05]; IC\_T5 = [34.39; 43.21]; IC\_T6 = [75.24; 135.76]; IC\_T7 = [32.63; 79.27]; IC\_T8 = [27.79, 36.41]. Given these intervals, it is possible to see that for 95% confidence the population mean for the first Task would not exceed 25 seconds, the second, third and eighth do not exceed 37 seconds, the fourth and fifth do not exceed 45 seconds, the sixth does not exceed 136 and the seventh 80 seconds.

As for the number of errors, the intervals are: IC\_T1 = [-0.05; 0.15]; IC\_T2 = 0; IC\_T3 = [0.01; 0.39]; IC\_T4 = 0; IC\_T5 = [0.04; 1.06]; IC\_T6 = [0.18, 1.92]; IC\_T7 = [-0.18; 0.68]; IC\_T8 = 0. Based on these intervals, it is possible to see that both task 1, 3 and 7 will have at most a value close to one error, while task 5 and 6 will have values close to two. As to the second, fourth and eighth it is not possible to calculate the interval because no errors have been made.

After performing all tasks, the users answered a satisfaction SUS questionnaire. The final SUS score was 75.63. According to the SUS score scale, this is a value that is situated on rank B, which indicates a system with good usability (Brooke 2013). This way, it is possible to reckon that, based on the answers, the system has a good classification in terms of usability.

## **Case study**

After performing usability tests some critical corrections to the interface were made. In this way, it is possible to carry out the case study, with the modified attribute selection and the attribute name of the CUQ. The major change, in this regard, was to implement two additional visual hints

that consist in drawing a bounding box around the axis label to alert the user that is in the presence of a button. Additionally, when the cursor overlaps the label, it is filled with color blue.

The case study was carried out with the head of the Statistics and Prospective Unit. Before starting the interaction with the system, all the developed mechanisms were explained. Then the user was given total freedom to explore the tool, for a period of 15 minutes.

Some suggestions and remarks were made. The most relevant consisted in grouping degrees according to their type: pre-Bologna, post-Bologna, specialization degrees, master's degrees and integrated master's degrees. Currently, as they are all together, the analysis by type of degree becomes a complicated task to accomplish. Another suggestion was to change the teachers' CUQ to a scale from red to blue, as it is used in the reports developed by the Unit.

Despite these suggestions, the Head of Unit was very interested during the interaction with the application, looking for scenarios for which the Unit already had produced an analysis and verified that, in fact, it was possible to see this information in the system. Using concrete scenarios from a course in a degree where the person already knew that there were weak approval ratings, she used the filtering mechanism to limit the set of results and expanded the course to analyze the level of approvals over the years. Then the relation between approvals and grades on the scatter plot was established, selecting the attribute on the axis to confirm that it is indeed possible to easily analyze the information in a multi-year time component. An interactive tool that allows analyzing different attributes, such as approvals, CUQ evaluations, grades and number of enrollments, can give a huge contribution to the continuous reformulation of all curricular units, to improve the whole learning process.

## **Discussion**

Regarding the purpose of the work developed and the requirements for implementing the application, it was intended that the tool could provide information related to the evolution of degrees and courses, considering a temporal component.

Given the suggested modifications, some of them have already been implemented, such as the provision of a visual aid for the selection of attributes and the change of the attribute name for the CUQ evaluations of the courses. For future implementation, it may be necessary to group all the degrees according to their type, to modify the color scale of the CUQ of the teachers and to allow to filter by degree type in the search bar. Following this, it is possible to improve all the interaction with the system, in order to make it more perceptible to the users.

Despite the many interaction mechanisms implemented, the participants, mostly, did not forget any of them and applied them whenever necessary, revealing that the initial demonstration of the system was sufficient for its use and that the application is intuitive and easy to learn. Furthermore, it coped adequately with a fairly large dataset, spanning more than a decade of academic data for an entire college.

## **CONCLUSIONS AND FUTURE WORK**

With the increase of students in the academic area, it is important to follow their path and to analyze the adaptation of the courses and the degrees in relation to the current teaching. By collecting data, it is possible to monitor all these issues with the aid of interactive visualizations. Based on this context, data were extracted from Instituto Superior Técnico, which served as a



basis to develop TécnicoVis, which allows analyzing the evolution of degrees and courses over the years.

In the first phase, several works were analyzed, where most don't allow curricular units monitorization. The problems and strengths in these works were also detected and it was based on this analysis that it was possible to choose the most relevant interaction mechanisms used in the application. Subsequently, it was necessary to proceed with the conception of the questions whose answers were intended to be obtained through the interaction with the application, followed by data extraction. After that, non-functional representations that met the initial aim were created. With the defined visualization scheme, the development phase of the system was started, which includes the implementation of the backend and frontend.

In the case of visualizations, we used different types of views with alternative encodings. Regarding the matrix and the multiple bar graphs, the color range was used according to the western conventional code and for the sparklines, scatter plot and line charts, position was used, providing alternative ways to analyze the different types of data.

To evaluate the application, usability tests and a case study were performed, to ensure that it is easy to interact with the application through the implemented mechanisms that allow the exploration of the whole data and to highlight important patterns. Most of the tasks did not take over sixty seconds and with an average value of one error per task, which proves to be a satisfactory result and it is possible to conclude that the users perceived all the different mechanisms. Regarding the case study, it is important to note that given the context in which it is inserted, relevant patterns were identified during the interaction with TécnicoVis.

This way, it is possible to verify that the developed tool allows teachers to perceive and identify certain problems of the different degrees and courses and to act as an aid to improve teaching methods and consequently, to reduce course failure.

With usability tests, we realized that there are still some potential improvements to be made. Although some have already been implemented, it is still necessary to consider changing the colors of the CUQs to a scale from red to blue, to allow the search bar filtering mechanism to search by type of degree and group all its types. Since the current color scale isn't color blind safe, it will be necessary to include a scale that fits in this context in a future iteration of the system.

Besides all these changes, it would be interesting to have access to extra data that would allow focusing in detail on the path of the student, namely having access to the courses' grades, tests and exam grades, high school grade and the gender of the students. In this way, it would be possible not only to analyze the evolution of degrees, courses and the more superficial analysis to the students (average grades) but also to analyze their situation in a more detailed and concise view. With access to this information, it would be possible to identify potential students with difficulties in certain courses and to develop educational content, in order to allow the student to achieve his potential throughout his academic career.

## **ACKNOWLEDGMENTS**

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.

## REFERENCES

- Ali, L., Hatala, M., Gašević, D., & Jovanović, J. (2012). A qualitative evaluation of evolution of a learning analytics tool. *Computers & Education*, 58(1), 470–489.
- Baepler, P., & Murdoch, C. J. (2010). Academic Analytics and Data Mining in Higher Education. *International Journal for the Scholarship of Teaching and Learning*, 4(2), 1–9.
- Barata, R., Silva, S., Martinho, D., Cruz, L., & Guerra, L. (2013). Open APIs in Information Systems for Higher Education.
- Brooke, J. (2013). SUS : A Retrospective. *Journal of Usability Studies*, 8(2), 29–40.
- Da Silva, C. G., & Da Rocha, H. V. (2007). Learning Management Systems' database exploration by means of Information Visualization-based query tools. In *Proceedings - The 7th IEEE International Conference on Advanced Learning Technologies, ICALT 2007* (pp. 543–545)
- Dernoncourt, F., Taylor, C., O'Reilly, U.-M., Veeramachaneni, K., Wu, S., Do, C., & Halawa, S. (2013). MoocViz: A Large Scale, Open Access, Collaborative, Data Analytics Platform for MOOCs. *Nips '13*, (July 2015).
- Desai, A., Mian, M., Hazel, D., Teredesai, A., & Benner, G. (2014). Data visualization in educational datasets using a rule-based inference system. In *Proceedings - 2014 IEEE International Congress on Big Data, BigData Congress 2014* (pp. 462–469).
- Gan, X., Liu, K., & Liu, P. (2009). Visual analysis of college students' scores in English test. In *Proceedings of 2009 4th International Conference on Computer Science and Education, ICCSE 2009* (pp. 1816–1819). (1)School of Foreign Language, Wuhan University of Science and Technology.
- Gao, Y. (2010). A study on mass higher education in the world-based on comparative perspectives. In *Education and Management Technology (ICEMT), 2010 International Conference on* (pp. 528–530).
- May, M., George, S., & Prévôt, P. (2011). TrAVis to enhance online tutoring and learning activities: Real-time visualization of students tracking data. *Interactive Technology and Smart Education*, 8, 52–69.
- Mazza, R. (2009). *Introduction to Information Visualization*. October, 1(978-1-84800-218-0), 139.
- Mazza, R. (2009). *Introduction to Information Visualization*. October, 1(978-1-84800-218-0), 139. <http://doi.org/10.1007/978-1-84800-219-7>
- Mazza, R., & Dimitrova, V. (2007). CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. *International Journal of Human Computer Studies*, 65(2), 125–139.

Murray, S. (2013). *Interactive Data Visualization for the Web, An Introduction to Designing with D3*. Sebastopol: O'Reilly Media.

Plataniotis, K. N., & Venetsanopoulos, A. N. (2000). *Color Image Processing and Applications*. New York, NY, USA: Springer-Verlag New York, Inc.

Weiland, A., & Manssour, I. H. (2015). Towards visual analysis techniques for monitoring students of distance education courses. 6th Workshop on Visual Analytics, Information Visualization and Scientific Visualization, At Mercure Salvador Rio Vermelho Hotel. Salvador, 4, 5.