Co-funded by the Erasmus+ Programme of the European Union

LALA

Building Capacity to Use Learning Analytics to Improve Higher Education in Latin America

(586120-EPP-1-2017-1-ES-EPPKA2-CBHE-JP)

WPD3.O.4

Design of Learning Analytics Tools

Version 1.0

12/04/2019

Authors: Margarita Ortiz[1], Alberto Jimenez[1], Ricardo Maya[1], Pedro J. Muñoz-Merino[2], Pedro Manuel Moreno-Marcos[2], Jon Imaz Marín[2], Carlos Delgado Kloos[2], Miguel Ángel Zúñiga Prieto[3], Marlon Ulloa[3], Ronald Pérez[4], Mar Pérez-Sanagustín[4], Valeria Henríquez[5], Julio Guerra[5], Rafael Ferreira Leite de Mello[6], Tom Broos[7], Martijn Millecamp[7].

[1] Escuela Superior Politécnica del Litoral, ESPOL
[2] Universidad Carlos III de Madrid, UC3M
[3] Universidad de Cuenca
[4] Pontificia Universidad Católica de Chile
[5] Universidad Austral de Chile
[6] University of Edinburgh
[7] Katholieke Universiteit Leuven

Work funded by the LALA project (concession No. 586120-EPP-1-2017-1-ES-EPPKA2-CBHE-JP). This project has been funded with the support of the European Commission. This communication reflects only the opinions of the authors, and the Commission and the Agency are not responsible for the use that may be made of the information contained therein.

Table of Contents

1. Introduction

In the process of adopting learning analytics, the next step once a baseline has been raised concerning the reality and need of the institution of higher education is to design a tool that meets the requirements that the target group establishes (teachers, students, institutional leaders).

In the design process, the target group should not only be considered as the primary source of information but should be involved throughout the tool design process to provide constant feedback to researchers, programmers, and designers. Among the methodologies recommended in this phase are agile methodologies such as Scrum or kambam (Kniberg & Skarin 2010). However, if necessary, adaptations should be made to meet the needs of the designers.

In the context of this project, the objective of this deliverable is to describe how tools for learning analytics developed in the European context have been adapted to the four Latin American institutions that are regular project partners. Fundamentally, two tools have been adapted: a counseling tool consisting of visualization panels to support decision making when deciding which subjects to take; in addition to another to automatically support students' work in online learning contexts, and another tool about early warning of dropout.

This deliverable is connected to the LALA Framework https://www.lalaproject.org/es/deliverables/ because it is the next step, once it has been compiled through the LALA canvas, interviews, focus groups and surveys, the needs of the different educational actors. Thus, this document includes the design of learning analytics tools adapted to the needs of four Latin American institutions, based on the definition of the "LALA framework" deliverable, which is a previous result in the project.

In the process of designing and implementing the learning analytics tools of the LALA project, two parts are distinguished: the backend architecture and the frontend of the tools. The backend mentions how a generic database was designed so that any Higher Education Institution (HEI) can adapt it to its needs. Then, it describes how each university adapts that generic base. On the other hand, in the frontend, each university mentions the process of design and adaptation of a tool for academic advising and early warning of academic desertion, as proposed in the project. Although each university had autonomy with this phase, all of them focus on constant interaction with the target group and its connection with the LALA framework. The source code of all the developed tools is available in the following link for the use and adaptation of any institution https://git.cti.espol.edu.ec/LALA-Project-EN. Additionally, other analytical tools are included, adapted by the universities. Finally, the conclusion highlights the contributions of this deliverable and next steps to follow.

2. Backend: Generic architecture for the dropout early warning system and counseling tool

2.1 Generic LALA database

We decided to create a generic database and a generic architecture, based on the common needs of institutions in Latin America. The idea behind is for any Latin American institution that wants to use learning analytics could start from a base that can be easily adapted to each institutional need. This generic LALA base applies to both the counseling tool and the early dropout prediction tool.

2.1.1 Database design

Next, the generic database is described. Figure 2.1.1 shows an Entity-Relationship diagram of it. Moreover, its tables with its fields and description are shown.

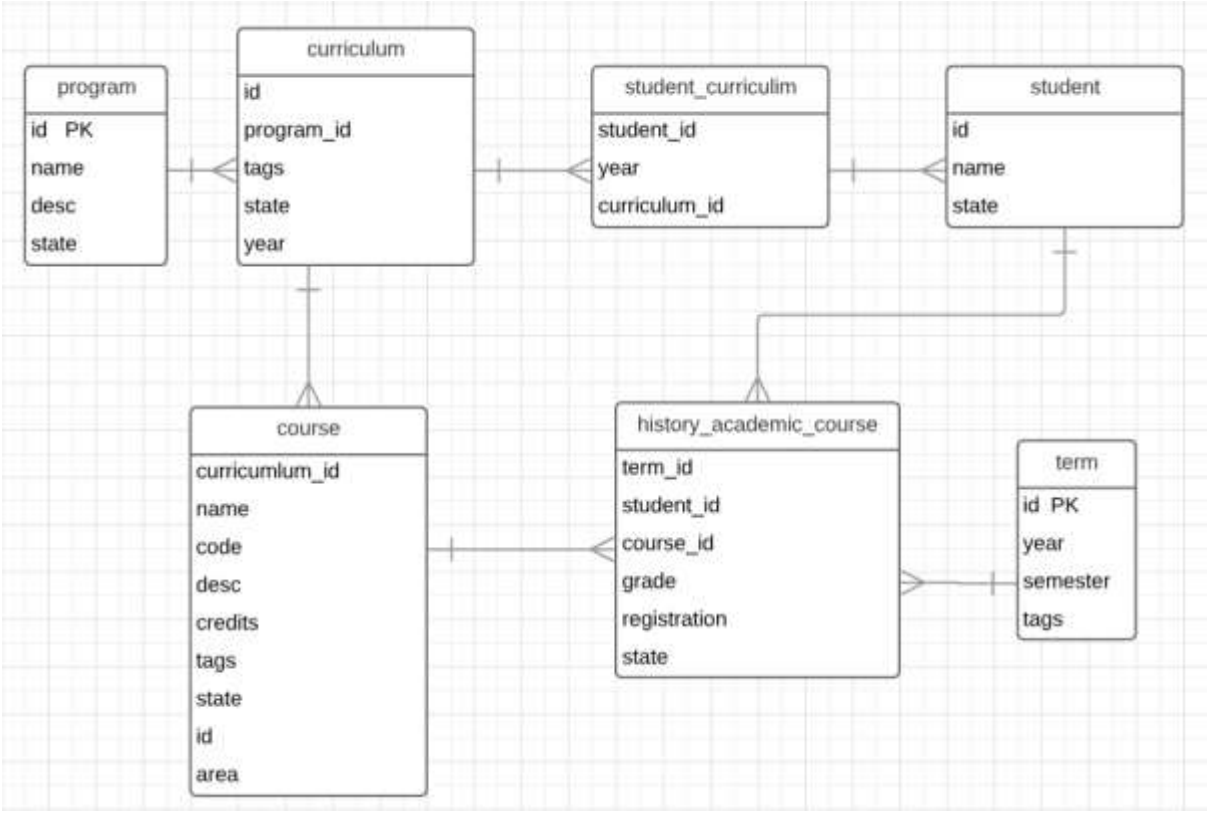


Figure 2.1.1 Generic Database

Table: Program
Description: Indicates the career or program dictated by the university
Fields:

Name	Datatype	Description
id	bigint	Autoincremental, field used as primary key.
name	text	Indicates the name
desc	text	Indicates the description
state	text	Indicates the status can be: active or inactive (for a career that is no longer dictated at present)

Table: Curriculum
Description: Indicates the meshes for the different careers or programs dictated by the university
Fields:

Name	Datatype	Description
id	bigint	Autoincremental, field used as primary key.
program_id	bigint	Field used as foreign key of the program table
year	int	Indicates the year of creation
tags	text	Indicates the labels associated with the mesh
state	text	Indicates the status can be: active or deactivated (for meshes that are no longer dictated at present)

Table: Course
Description: Indicates the courses associated with the different meshes of the races or programs dictated by the university
Fields:

Name	Datatype	Description
id	bigint	Autoincremental, field used as primary key.
curriculum_id	bigint	Field used as foreign key of the curriculum table
name	text	Indicates the name
code	text	Indicates the code
desc	text	Indicates a brief description
credits	int	Indicates the credits associated with the course

area	text	Indicates the area to which it belongs. Example: Basic Training, Vocational Training, Itinerary, Free Choice
tags	text	Indicates the labels associated with the course
state	text	Indicates the state can be: active or inactive (for courses that are no longer dictated at present)

Table: Term
Description: Indicates the academic term dictated by the University
Fields:

Name	**Datatype**	**Description**
id	bigint	Autoincremental, field used as primary key
year	bigint	Indicates the year
semester	text	Indicates the semester
tags	text	Indicates the labels associated with the term

Table: Student
Description: Indicate the students registered by the University
Fields:

Name	**Datatype**	**Description**
id	bigint	Autoincremental, field used as primary key
name	text	Indicates the year
state	text	Indicates the status can be: active or graduate or graduated, etc. state -> active or inactive. Data that can be used for statistics but not to present in the system (a student that will not be analyzed with the visualization panel but its performance will serve to take statistics, example the historical average in a subject) anonid is the student's anonymized code

Table: History Academic Course
Description: Indica el historial académico de los estudiantes en la Universidad
Fields:

Name	Datatype	Description
student_id	bigint	Field used as the foreign key of the student table
course_id	bigint	Field used as foreign key of the course table
term_id	bigint	Field used as foreign key of the term table
grade	double	Indicates the grade obtained
registration	text	Indicates the status of the record. Ex: Registered, Validated, Accredited
state	text	Indicates the state. Ex: Approved, Failed, Canceled

Table: Student Curriculum
Description: Indicate the students registered by the University
Fields:

Datatype	Datatype	Description
student_id	text	Field used as the foreign key of the student table
year	bigint	Indicates the year of entry of the student to the mesh
curriculum_id	text	Field used as foreign key of the curriculum table

2.1.2 General Architecture

For the consumption and data load of the main application, the creation of an API REST was chosen with an architecture as shown in figure 2.1.2, with the purpose that in future time not only the counseling and warning systems can use the academic data, but future applications developed by universities can obtain this data without being linked to a particular application. Thus, avoiding connections to the database by application, and centralizing all data access to the developed API. This proposed architecture is the most viable in Latin America since the majority of IT departments of each institution restrict their data access to certain systems, this being a limitation when developing independent applications that require the same academic data.

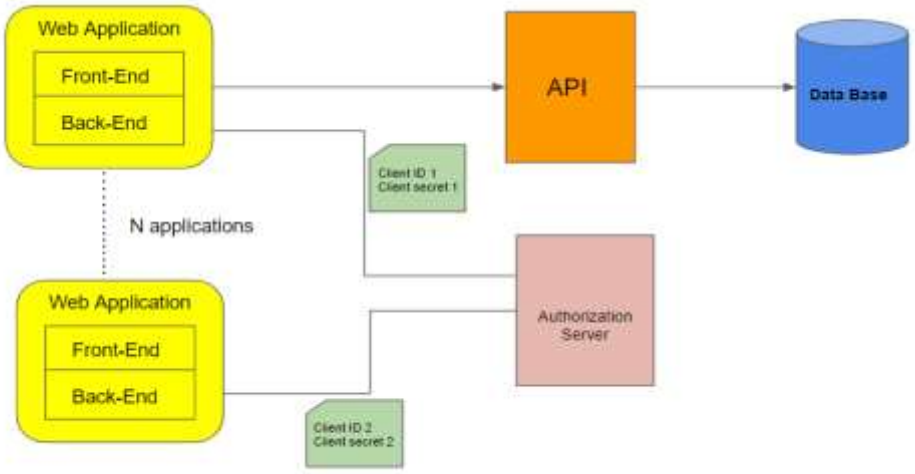


Figure 2.1.2 General Architecture of the system

The API Rest was developed using NodeJS technologies with an implementation of two layers of abstraction: the model and the controller, together with configuration files for the connection to the database and information necessary for the connection to the authorization server. However, the implementation of an API implies that there is a security and authentication protocol to avoid unauthorized applications or users having access to the data. For this, the OAuth 2.0 protocol was used, which defines the flow of authorization protocols according to the application's implementation design, thus allowing access to the data in a secure manner.

This protocol involves several roles for its use, such as:

- **Authorization server:** The server in charge of granting the data access token.
- **Owner of the resource:** It is the user who will have access to the data, thus providing their credentials for verification.
- **The client:** The application responsible for making a request to the authorization server using the credentials provided by the owner of the resource, and then making a request to the server of the resource with the access token provided.
- **Resource server:** It is the server in charge of validating the access token and returning the data in case the validation happens, in our case the developed API.

Due to the large amount of work necessary for the development of an authorization system that implements the protocols defined by OAuth 2.0, one that already implements them was chosen that was Okta.

Okta is a cloud service which allows user management, authorization, and authentication (which already implements the OAuth 2.0 protocol) for easy integration with different applications. This service gives the ability to the user who owns a developer account, to create and configure an authorization server capable of granting an access token to the application, as well as to create credentials to the applications that will interact with it. Because of this, a user with an administrator role must be in charge of creating a developer account, creating

the necessary credentials for each application that will consume the API, deploying an authorization server and configuring the correct authorization flow.

OAuth 2.0 also defines several types of authorization flows, but the one used for the implementation of the API is the "Client Credentials Grant Type", this because the authorization process between the 2 applications (Counseling System and API Rest) is a process in background, transparent to the user who interacts with the main system.

How the API validates the credentials is using an authorization token sent in each request, this access token is of the JWT (JSON Web Token) type, which allows being validated by itself. It consists of 3 parts, all of them encoded separately in base64: the first, known as header is information about the algorithm to be used by the server to generate the signature, for the case of this implementation is RS256 which is an asymmetric encryption algorithm already implemented in the Okat authorization server, then there is the payload that contains information about the user linked to the token, and finally the signature coded with the header and the payload using the private key of the server.

How the API validates the token is by using the public key of the authorization server, thus validating that the three parts of the token are corresponding to each other, that is, there is no type of alteration in the provided token.

2.2 Adaptations to the generic database

Although a generic base was created, the needs of each university are individual. That is why adaptations are needed. Next, the different adaptations to the generic LALA base that each university made are presented.

2.2.1 Adaptations in Escuela Superior Politécnica del Litoral (ESPOL)

2.2.1.1 Design

The following tables were added to include necessary information in the counseling dashboard

Table: Course Available
Description: Indicate the courses available to the student during the registration period
Fields:

Name	Datatype	Description
student_id	text	Field used as the foreign key of the student table
course_code	text	Indicate the code
course_name	text	Indicate the name

course_area	text	Indicates the area to which it belongs. Example: Basic Training, Vocational Training, Itinerary, Free Option
course_level	text	Indicates the level to which the course belongs in the mesh

Table: Course Counseling
Description: Indican los cursos sugeridos realizados por el consejero durante los términos
Name:

Name	Datatype	Description
student_id	text	Field used as the foreign key of the student table
course_code	bigint	Indicates the year of entry of the student to the mesh
course_name	text	Field used as a key for the term of the table
term_id	bigint	Field used as a key for the term of the table
teacher_name	text	Indicate the name of the counselor (in the case of ESPOL of the teacher)

Table: Wellness
Description: Indican registros de la asistencia en Bienestar Estudiantil
Name:

Name	Datatype	Description
student_id	text	Field used as the foreign key of the student table
term_id	bigint	Field used as foreign key of the term table
incidence_type	text	Indicates the type of incident
case_type	text	Indicate the type of case
summary	text	Indicates the attendance summary
designated_person	text	Indicates the person designated for follow-up
created_at	date	Indicates the date the creation of the record
incident_date	date	Indicates the date of assistance

2.2.1.2 Architecture

Because the developed API was thought in such a way that it is a centralized source of data, it was not necessary to create a different system, so the reader can refer to section 2.1.2 General architecture, which covers the architecture used for the creation of the API.

2.2.2 Adaptations in Universidad Austral de Chile (UACh)

2.2.2.1 Design

To meet the specific needs of the context of the Universidad Austral de Chile UACh, the following tables were added:

Table: Student_statistic_by_term
Description: Indicates the situation of the student in each semester studied
Fields:

Name	Data type	Description
student_id	text	Field used as the foreign key of the student table
year	bigint	Primary key composite., Year of statistics
termAvg	text	Average semester grades
accAvg	text	GPA accumulated
state	text	Situation with which the student has finished that semester year
semester	text	Statistical composite primary semester key

Table: Student_dropout
Description: Indicates the progress of the student in high school and the plans to finish it
Fields:

Name	Data type	Description
student_id	text	Field used as the foreign key of the student table
bach_completed_courses	bigint	Number of completed baccalaureate courses
bach_total_courses	bigint	Total number of baccalaureate courses
estimated_terms_to_complete_bach	text	Prediction of semesters to complete the

		baccalaureate
estimated_probability	bigint	Probability of finishing the baccalaureate

Table: group_course_academic_by_term
Description: Indicates the distribution of grades for a given course and semester
Fields:

Name	Data type	Description
distribution	text	Distribution values
distRange	text	Distribution ranges
StudentCount	bigint	Total number of students in the distribution
failRate	bigint	Failure rate of
droRate	bigint	Dropout rate within the distribution
CountByDelay	bigint	Number of students who took the course outside of the predicted semester according to the curriculum
code	text	Field used as a foreign key of the course table using the code field
type	bigint	Type of distribution to generate
year	bigint	Year of distribution
semester	bigint	Semester of the distribution

Table : group_course_academic_by_cohort
Description: Indicates the distribution of grades for a given course and cohort
Fields:

Name	Data type	Description
distribution	text	Distribution values
distRange	text	Distribution ranges
StudentCount	bigint	Total number of students in the distribution
failRate	bigint	Failure rate of
droRate	bigint	Dropout rate within the distribution
CountByDelay	bigint	Number of students who took the course outside of

		the predicted semester according to the curriculum
course_id	bigint	Field used as a foreign key of the course table , using the PK
year	bigint	Year of the cohort considered in the distribution

Table: group_course_academic
Description: Indicates the historical distribution of grades for a given course
Fields:

Name	**Data type**	**Description**
distribution	text	Distribution values
distRange	text	Distribution ranges
StudentCount	bigint	Total number of students in the distribution
failRate	bigint	Failure rate of
droRate	bigint	Dropout rate within the distribution
CountByDelay	bigint	Number of students who took the course outside of the predicted semester according to the curriculum
code	text	Field used as a foreign key of the course table using the code field

2.2.2.2 Architecture

An architecture with three parts is defined as shown in figure 2.2.2.2 : (A) data services in a backend that generally deliver the data in JSON format; (B) the LA tools that run in the Web browser; and (C) a conversion service of real identifiers -> anonymized

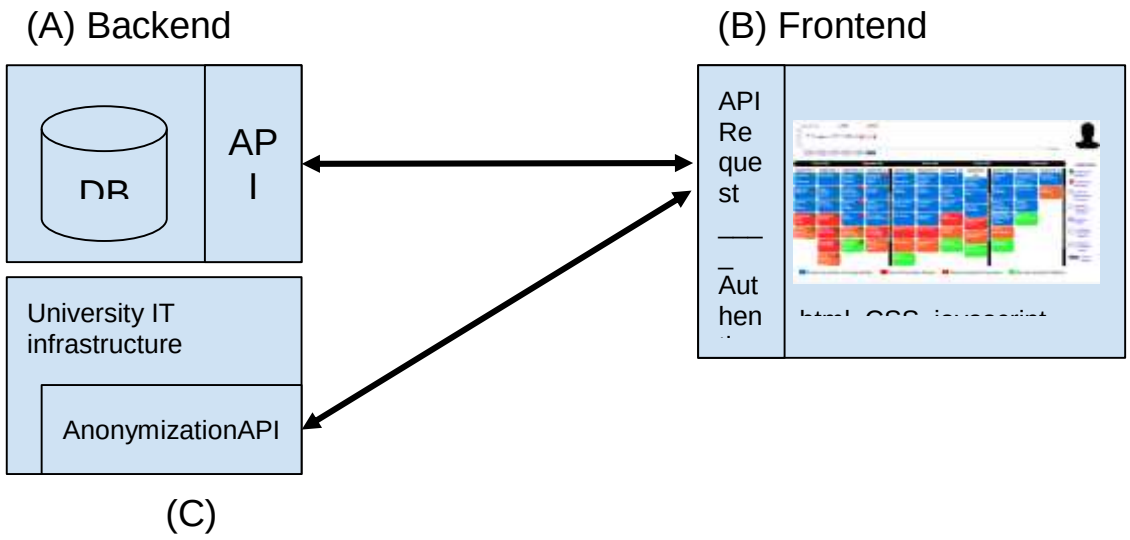


Figure 2.2.2.2 System architecture

UACh API - Counseling TrAC for program directors

For the messaging tool described in this document, 3 data services have been defined, plus the anonymization service that will be built by UACh . The services are detailed below:
GetProgramStructure (v1.0)

Name	GetProgramStructure
Version	1.1
Description "id ": string, "name": string,	This service provides the structure of a study program including the courses and in which semester they are located. Since a program can have several curricular meshes, the service requires that the year be specified and will deliver the current program structure in that year.
URL	server / GetProgramStructure
Parameters	program: string (example: 1708) year: integer (example: 2016)

Exit	{
	"program": PROGRAM_STRUCTURE,
	" lastDataUpdate ": string
	}
	
	PROGRAM_STRUCTURE: {
	"id": string,
	"name": string,
	" desc ": string,

	"plan": string,
	"year": integer,
	"tags": [string],
	"terms": [TERM]
	}
	
	TERM: {
	"position": int ,
	"name": string,
	"tags": [string],
	"courses": [COURSE]
	}
	
	COURSE: {
	"id": string,
	"name": string,
	" desc ": string,
	"credits": string,
	" historicGroup : GROUP_COURSE_ACADEMICS,
	"requisites": [string],
	"tags": [string]
	}

Example	Call : server / GetProgramStructure? program = 1708 & year = 2016

GetStudentAcademics (v1.0)

Name	GetStudentAcademics
Version	1.0
Description	
URL	server / GetStudentAcademics
Parameters	student: string (example: id0fa34) program: string (example: 1708) distributionBinType : string (example: "POINT")

Exit	{
	"student": STUDENT,
	" lastDataUpdate ": string
	}
	STUDENT: {
	"id": string,
	"program": string,
	"plan": string,
	" cohortYear ": integer,
	" previousPlans ": [string], // en If changes mesh
	" bachCompletedCourses ": number,
	" bachTotalCourses ": number,

	" estimatedTermsToCompleteBach ": number,
	" estimatedProbability ": number,
	" termAcademics ": [TERM_ACADEMICS]
	}
	
	TERM_ACADEMICS: {
	"year": integer,
	"semester": integer,
	" termAvg ": number, // PSP (appears as PGP)
	" accAvg ": number, // PGA
	" coursesTaken ": [COURSE_TAKEN]
	}
	COURSE_TAKEN: {
	" courseId ": string ,
	" section ": number , // parallel group or section when the course is given in several groups. If there are no parallel groups, comes 0.
	"grade": number ,
	" registration ": string , // completed, validated, validated, canceled
	"status": string, // Approved , Failed
	" cohortGroup ": GROUP_COURSE_ACADEMICS,
	" classGroup ": GROUP_COURSE_ACADEMICS, // statistics the course that semester and y that year and that parallel group (section)
	" historicGroup : GROUP_COURSE_ACADEMICS
	}
	
	GROUP_COURSE_ACADEMICS: {
	"distribution": [CHART_POINT],
	" distRange ": { number, number }, // min, max
	" studentCount ": number,
	" failRate ": number,
	" dropRate ": number,
	" countByDelay ": {"0": number, "1": number, "2": number, "3": number, ...}
	// count students taking the course with 1,2 ... n terms of delay
	}
	
	CHART_POINT: {
	"label": string,
	"value": number
	}

GetCourseAcademics (v1.0)

Name	GetCourseAcademics
Version	1.0
Description	
URL	server / GetCourseAcademics
Parameters	courseIds : string (code of courses separated by comma. example: INFO058, INFO090)

Exit	{
	"courses": [COURSE_ACADEMICS],
	" lastDataUpdate ": string
	}
	
	COURSE_ACADEMICS: {
	" courseId ": string,
	" courseName ": string,
	"equivalent": [string],
	"credits": number,
	"curriculums": [CURRICULUM],
	" avgFailRate ": number,
	" avgStudentCount : number,
	" avgIdealLoad ": number,
	" avgOnTime ": number,
	" avgFirstAttempt ": number,
	" academicsByTerm ": [ACADEMICS_BY_TERM],
	" academicsByLoad ": [ACADEMICS_BY_LOAD],
	" academicsByLate ": [ACADEMICS_BY_LATE],
	" academicsByTime ": [ACADEMICS_BY_TIME]
	
	}
	
	RESUME:
	{
	"program": string,
	" programName ": string,
	"plan": string
	}
	
	// distribution of academic performance on the course by term
	ACADEMICS_BY_TERM: {
	"year": integer,
	"term": integer,
	"academics": GROUP_COURSE_ACADEMICS
	}
	
	// distribution of academic performance on the course of all
	// students who took the class with certain number of parallel
	// load (number of courses taken in the same semester - 1)
	ACADEMICS_BY_LOAD: {
	" parallelLoad ": integer,
	"academics": GROUP_COURSE_ACADEMICS
	}
	
	// distribution of academic performance on the course of all
	// students who took the class termsDelayed terms late than
	// the ideal term in which the course is in the program
	// structure
	ACADEMICS_BY_LATE: {
	" termsDelayed ": integer,
	"academics": GROUP_COURSE_ACADEMICS
	}

	// distribution of academic performance on the course of all
	// students who have taken the course before "repetition" times
	ACADEMICS_BY_TIME: {
	"repetition": integer,
	"academics": GROUP_COURSE_ACADEMICS
	}
	
	GROUP_COURSE_ACADEMICS: defined in the previous service

Anonymization
The data is hosted on the backend in a database where the identifiers of the students have been anonymized. To allow the user of the tool to choose students by real identifiers (in our case the RUT or DNI), an anonymization service is defined that must be built by UACh . This service is not hosted in the LALA infrastructure, but not in the IT services infrastructure of the institution.

GetId (v1.0)

Name	GetId
Version	0.1
Description	This service returns the anonymized id from a "real" id delivered as a parameter
URL	to define with IT Unit xxx / GetId? student = 51242133
Parameters	student: string

Exit	{
	"id": string,
	" anonId ": string,
	"name": string,
	"email": string
	}

2.2.3 Adaptations in Universidad de Cuenca (UCuenca)

2.2.3.1 Design

Figure 2.2.3.1 shows the design diagram of the database that will record the data required to present the information of the different tools designed in U Cuenca. The tables that are displayed in blue have been inherited and extended (with the fields in red) of the generic database design; the rest of the tables are specific to the U Cuenca version.

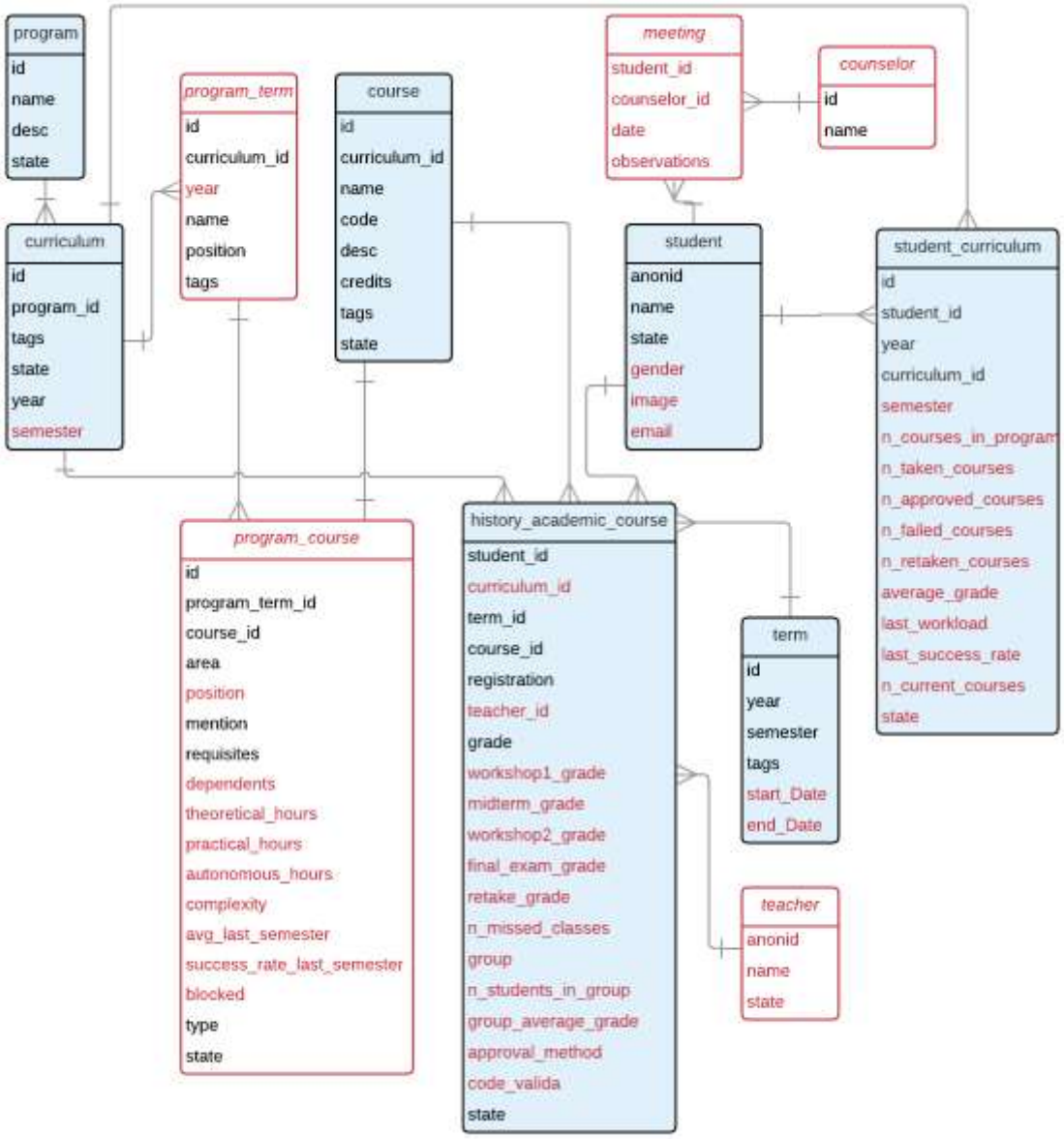


Figura 2.2.3.1 Database design

In this section, you will find *red* and *italic* tables or fields that differ from the generic model.

Table: program
Description: Indicates the careers or programs offered by the university. For example: Systems Engineering, Electrical Engineering, Medicine, etc.
FIelds:

Name	Datatype	Description
id	int4	auto incremental, field used as primary key.
name	text	Indicates the name.
desc	text	Indicates the description.

state	text	Indicates the status that can be: active or inactive (for a career that is no longer dictated at present).

Table: curriculum
Description: Indicates the curriculums (study plan) for the different careers or programs dictated by the university
Fields:

Name	Datatype	Description
Id	int4	Auto incremental, field used as primary key.
program_id	int4	Field used as foreign key of the program table.
year	int4	Indicates the year of creation or definition of the curriculum.
semester	text	Indicates the semester of the creation or definition of the curriculum (example: 1S, 2S).
tags	text[]	Indicates the labels associated with the curriculum.
state	text	Indicates the status that can be: active or inactive (for a career that is no longer dictated at present).

Table: program_term
Description: Indicates the terms on which the curriculum is governed. Description of the semesters that have been planned for the curriculum. Example: First Semester, Second Semester, etc.
Fields:

Name	Datatype	Description
id	int8	Auto incremental, field used as primary key.
curriculum_id	int4	Field used as foreign key of the curriculum table.
name	text	Name of the term. For example: First Semester, Second Semester, Fourth Semester.
year	int4	Indicates the year of the career to which the term belongs. It is used to group terms in planned years for a curriculum. For example, the term (1st semester) belongs to the First year, the term 2nd semester belongs to the First Year, the term 4 semester belongs to the second year.
position	int4	Used as an aid to visualization, it indicates the semester

		number or the position (column) in which that semester will be presented within the curriculum. When the curriculum is presented, it will be presented as a matrix; where the semesters are the columns and each course that is given in a semester will go in a different row of the corresponding column. Position refers to the column in which the semester will be displayed.
tags	text[]	Indicates the labels associated with the curriculum.

Table: course
Description: Indicates the courses associated with the different curriculums of careers or programs taught at the university. It will register all the courses that are dictated in the university, without concerning the career or program in which they are dictated. For example, if the Mathematics and Optional 1 courses are taught in more than one course or program, in this table there will be a single record for each course.
Concerning a course planned as optional (eg, Optative 1), the actual course (taken by a student in a specific academic period) will be another one, and will be determined for each academic period. Therefore, in this table, a record must be created for each optional course; as well as records for each course that was dictated as optative in the different academic periods. For example, assuming that in three different academic periods the optative course 1 was given as Advanced Programming in the first academic period and Programming for Cloud Environments in the second and third academic period; then, in the Course table there will be three registers: one for the optative 1 course, another for the Advanced Programming course taught in the first academic period, and a third for the Course for Cloud Environments taught in the second and third academic periods (no it is necessary to create two courses for Cloud Environments). During the execution of the academic period (in the history_academic_course table) the correspondence between these courses will be indicated.
Fields:

Name	Datatype	Description
Id	int8	Auto incremental, field used as primary key.
Name	text	Indicates the name.
code	text	Indicates the course code in the institution's system.
desc	text	Indicates a brief description.
credits	int4	Indicates the credits associated with the course. In case the number of credits varies depending on the career or program it will be indicated in the program_course table.
tags	text[]	Indicates the labels associated with the course.
state	text	Indicates the state that can be: ACTIVE or INACTIVE (for courses that are no longer dictated at present).

Table: program_course
Description:
Indicates the courses planned to be taught in each semester (program_term) of a curriculum. Describe the characteristics of a course for a given curriculum in a semester (program_term). The values in this table do not correspond to a specific student.
Concerning the courses planned as optative (value of the type = OPTATIVE field) (e.g., Optative 1), the actual course (taken by the student in a specific academic period) will be another one, and will be determined for each academic period. Like the course table, this table requires creating a record for each optative course; as well as records for each course that was dictated as optative in the different academic periods. For example, assuming that in three different academic periods the optative 1 course was given as Advanced Programming in the first academic period and Programming for Cloud Environments in the second and third academic period; then, in this table reference will be made to the three records that must exist in the Course table: one for the Optative 1 course, another for the Advanced Programming course given in the first academic period, and a third for the course Programming for Cloud Environments dictated in the second and third academic period (it is not necessary to create two courses Programming for Cloud Environments). During the execution of the academic period (in the history_academic_course table), the correspondence between these courses will be indicated.
Campos:

Nombre	**Tipo de dato**	**Descripción**
id	int8	Auto incremental, field used as primary key.
program_term_id	int8	Field used as a foreign key in the program_term table. It allows to indicate the semester for which a course for the curriculum has been planned (curriculum table) to which the semester belongs
course_id	int8	Field used as foreign key of the course table. It allows to indicate the course that has been planned to dictate.
area	text	Indicates the area to which the course belongs (Basic, Training, Qualification). A course can belong to different areas according to the grid (curriculum table) and the semester (field program_term_id).
position	int4	Used as an aid to visualization, it indicates the row number or the position in which the subject will be displayed within the column corresponding to the semester in which the subject will be taught. When the curriculum is presented, it will be presented as a matrix; where the semesters are the columns and each subject that is dictated

		in a semester will go in a different position (row) of the corresponding column.
mention	text	In the case of careers that have a major/minor, it indicates the mention to which the course corresponds.
requisites	text[]	A list with the codes of the courses that require to have been approved in order to take the current course. The list includes the codes of courses separated by comma.
dependents	text[]	A list with the codes of the courses that can be taken provided that the current course has been approved. The list includes the codes of courses separated by comma.
theoretical_hours	int4	Indicates the number of hours planned for the theoretical part of the course. These hours are for the curriculum (curriculum table) to which the semester belongs (field program_term_id).
practical_hours	int4	Indicates the number of hours planned for the practical part of the course. These hours are for the curriculum (curriculum table) to which the semester belongs (field program_term_id).
autonomous_hours	int4	Indicates the number of hours planned for the autonomous work that the students will dedicate to the course. These hours are for the curriculum (curriculum table) to which the semester belongs (field program_term_id).
complexity	text	I Indicates the complexity of the course. They will be percentages whose value could be: 25, 50, 75 or 100.
avg_last_semester	float8	Average obtained in the subject the semester (or previous year if the course is offered annually). Average between all the parallels offered.
success_rate_last_semester	float8	Approval rate in the subject the semester (or previous year if the course is offered annually). Calculated with the notes of all the students of all the parallels or groups offered.
blocked	text	If the subject is temporarily blocked to be offered. It will be blocked when any of the courses that are prerequisites of the current course (course_id)

		have not yet been approved.
type	text	Is the type of matter. OPTATIVE, ELECTIVE, COMPULSORY.
state	text	Indicates the state can be: ACTIVE or INACTIVE (for courses that are no longer dictated at present).

table: term
Description: Indicates the term or academic period in which the university dictated classes.
Fields:

Name	Datatype	Description
id	int8	Auto incremental, field used as primary key.
year	int4	Indicates the year of beginning of the academic period.
semester	text	Indicates the semester of the academic period. Example: 1S, 2S, 3S, 4S ...
tags	text[]	Indicates the labels associated with the term
start_date	timestamp	It is the start date of the semester in that subject. In month/day/year format.
end_date	timestamp	It is the end date of the semester. In month/day/year format.

Table: student
Description: Indicate the students registered by the University
Fields:

Name	Datatype	Description
anonid	int8	Anonymized student code.
name	bigint	Indicate the student's name. It will have value in case the academic data are not anonymized.
state	text	Indicates the status can be: ACTIVE, GRADUATE, GRADUATE, etc. Data that can be used for statistics but not to be presented in the system (a student that will not be analyzed with the visualization panel but its performance will be useful to obtain statistics, example the historical average in a subject).

gender	text	MALE or FEMALE
image	text	Location of the file with the student's photograph.
email	text	Student email.

Table: teacher
Description: Indica los profesores que dictaron cursos en la universidad
Fields:

Name	Datatype	Description
anonid	int8	Anonymized teacher code.
name	text	Indicate the name of the teacher. It will have value in case the academic data are not anonymized.
state	text	Status: ACTIVE or INACTIVE.

Table: history_academic_course
Description: Indicates the academic record of students at the University.
Courses that the student took or is taking in a curriculum. The courses that are taking in the current semester will have the value of the State = CURSANDO field.
There will be a record because every time the student took a course in a curriculum. That is, a record by: student, curriculum, term, course and registration (registration number or repetition).
In case of homologation or change of curriculum, a single record must be generated that corresponds to the course of the curriculum to which the student belongs (course after homologation or curriculum change).
Fields:

Name	Datatype	Description
Id	Int8	Auto incremental, field used as primary key.
student_id	int8	Field used as the foreign key of the *student* table.
curriculum_id	int4	Field used as a foreign key of the *curriculum* table.
term_id	int8	Field used as a foreign key of the *term* table that describes the academic period in which the student took or is taking a course
course_id	int8	Field used as foreign key of the *course* table.
registration	text	Indicate the registration number (intent) of the student (student_id) in the course (course_id). The possible values are: 1, if the registration corresponds to the first

		time the student takes the course; 2 if it is the second time; 3 if it is the third time.
teacher_id	int8	Identifier of the teacher who dictates the subject. Field used as a foreign key for the *teacher* table.
grade	float8	Indicates the final grade obtained by the student.
workshop1_grade	float8	Note obtained in the first Achievement.
midterm_grade	float8	Note obtained in the intercycle test.
workshop2_grade	float8	Note obtained in the second Achievement.
final_exam_grade	float8	Note obtained in the final exam.
retake_grade	float8	Note obtained in the substitution exam.
n_missed_classes	int4	Number of times the student did not attend classes (absences).
group	text	Group (classroom or parallel) in which the student studied the subject. In case a course, in a certain academic period (term_id), has been dictated in more than one parallel, this field allows knowing the parallel assigned to the student. If this course is validated, this field will have the CONVALIDATION value.
n_students_in_group	int4	Number of students who were classmates (classroom or parallel) in the course the student took.
group_average_grade	float8	Average grade of classmates for the course (course_id) taken by the student.
approval_method	text	Method of approval of the course: VALIDATED, APPROVED, CURSED, SCHOOL, REVALIDED, and SUFFICIENCY EXAM.
code_valida	text	Field used when the course taken was an optative course. In the case of courses planned as optative (e.g., Optative 1), the actual course (taken by the student in a specific academic period) will be determined for each academic period (term_id field). In these cases, the course_id field will have the code of the subject called Optative 1, and this field (code_valida) will have the code of the course that it effectively took as Optative 1.

		For example, in the scenario in which she is a student, she has repeated the optative 1 course where: i) The first time she took Optative 1, she took the Graphic Design course; ii) The second time he took the optative 1 course, he took the course Graphics in Three Dimensions. Therefore, there will be two records in this table with the value of the course_id field corresponding to the Optative 1 course. However; the first record will have the value of the code_valida field with the course_id corresponding to Graphic Design, and the second record will have the value of the code_valida field with the course_id corresponding to Three Dimensional Graphics
state	text	Indicates the state. Ex: APPROVED, REPROVED, VOIDED, REPROVED FOR FAULTS, CURSING

Table: student_curriculum
Description: This table records data (e.g., academic performance) about the different curriculums that a student has taken. There will be a record for each curriculum that the student took.
Fields:

Name	Datatype	Description
id	int8	Auto incremental, field used as primary key.
student_id	int8	Field used as the foreign key of the *student* table.
year	int4	Indicates the year of entry of the student to the curriculum.
curriculum_id	int4	Field used as a foreign key of the *curriculum* table.
semester	text	Indicates the semester of the student's entry into the curriculum.
n_courses_in_program	int4	Number of courses in the student's curriculum
n_taken_courses	int4	Number of courses of the curriculum that the student has completed, to date. Both approved and non-approved courses are considered. Courses taken more than once (which the student repeated) will be considered one time only.
n_approved_courses	int4	Number of courses of the curriculum that the student has approved, to date.

n_failed_courses	int4	Number of courses of the curriculum that the student has not approved, to date. If a course was not approved more than once, it will be considered only once to determine the amount.
n_retaken_courses	int4	Number of courses that the student took more than once, to date. If a course was taken more than two times it will be known only once to determine the amount.
average_grade	float8	Average grade of the courses approved by the student, to date.
last_workload	float8	Workload during the previous semester. It is calculated by adding the amount of theoretical, practical and autonomous work hours) of the courses the student took the previous semester.
last_success_rate	float8	Rate of approval of courses of the student in the previous semester. It is calculated by dividing the number of courses approved in the previous semester for the number of courses taken in the previous semester.
n_current_courses	int4	Number of courses the student is taking (only courses of the current semester).
state	text	Indicates if the student is ACTIVE or INACTIVE in the curriculum. Data that can be used for statistics, but not to present in the system. When the value is INACTIVE the curriculum will not be analyzed (e.g., on the display panel) but the student's performance will be used to obtain statistics (e.g., the historical average in a subject). There will only be one active curriculum per student and per career.

Table: counselor
Description: Indicates the counselors registered at the university.
Fields:

Name	Datatype	Description
id	int4	Counselor code.
name	text	Indicates the name of counselor.

Table: meeting
Description: Record observations about counseling sessions held with students.
Fields:

Name	Datatype	Description
id	Int8	Auto incremental, field used as primary key.
student_id	int8	Field used as the foreign key of the *student* table. Indicates the student that was attended in the counseling sesión.
counselor_id	int4	Field used as a foreign key of the *counselor* table. Indicates the counselor who provided the counseling sesión.
date	timestamp	Date and time of the counseling sesión.
observations	text	Observations of the counseling sesión.

2.2.3.2 Architecture

An architecture with two parts is defined as shown in the following figure: (A) data service in the backend that delivers the data in JSON format, and (B) the LA tools (dashboard) that run in the Web browser.

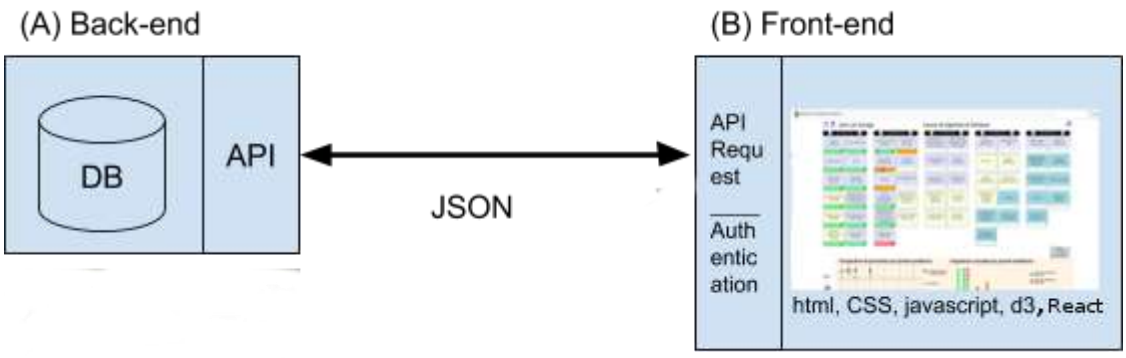


Figure 2.2.3.2 Architecture

UCuenca API - Counseling System

For the counseling tool described in this document, five data services have been defined for the time, there will be more services as the scale grows or the tool grows.
These services are detailed below:

GetStudentCurriculum (v1.0)

Nombre	GetStudentCurriculum
Versión	1.0
Descripción	This service delivers all the curriculum or program that the student is taking, as well as all the courses of the same.
URL	server/GetStudentAcademics
Parámetros	student : string (example: 1, 2, 3, ...) program : string (example: 1, 2, 3, ...)

Salida	{
	"student" : STUDENT,
	}
	
	STUDENT :{
	"id" : string,
	“name” : string,
	“state” : string,
	“email” : string,
	“student_curriculum”: STUDENT_CURRICULUM
	}
	
	STUDENT_CURRICULUM : {
	“id” : integer
	"year" : integer,
	“semester”: string,
	“n_courses_in_program”: integer,
	"n_taken_courses": integer,
	"n_approved_courses": integer,
	"n_failed_courses": integer,
	"n_retaken_courses": integer,
	"average_grade":double,
	"last_workload":integer,
	"last_success_rate": double,
	"n_current_courses": integer,
	"state": string,
	"curriculum": CURRICULUM
	}

	CURRICULUM:{
	"id": integer,
	"state": string,
	"year": integer,
	"semester": string,
	"program": PROGRAM
	“program_term”: PROGRAM_TERM
	}
	
	PROGRAM: {
	"id": integer,
	"name": string
	}
	
	PROGRAM_TERM: {
	"id": integer,
	"year": integer,
	"name": string,
	"position": integer,
	"program_courses": [PROGRAM_COURSE]
	}
	
	PROGRAM_COURSE: {
	"course_id": integer,
	"course_nombre": string,
	"course_codigo": string,
	"course_estado": string,
	"course_creditos": integer,
	"program_course_id": integer,
	"program_term_id": integer,
	"area": string,
	"position": integer,
	"mention": string,
	"requisites": string[],
	"dependents": string[],
	"theoretical_hours": integer,
	"practical_hours": integer,
	"autonomous_hours": integer,
	"complexity": string,
	"avg_last_semester": double,
	"success_rate_last_semester": integer,
	"blocked": string[],
	"program_course_state": string
	
	}

GetStudentAcademics (V1.0)

Nombre	GetStudentAcademics
Versión	1.0
Descripción	This service provides all the student's academic record, including subjects that have been and have not been approved.
URL	server/GetStudentAcademics
Parámetros	student_id: integer, curriculum_id: integer

Salida	{
	“historyAcademics”: [HISTORY_ACADEMICS_COURSE],
	}
	
	HISTORY_ACADEMICS_COURSE: {
	"student_id": integer,
	"curriculum_id": integer,
	"term_id": integer,
	"course_id": integer,
	"registration": string,
	"teacher_id": integer,
	"grade": double,
	"workshop1_grade": double,
	"midterm_grade": double,
	"workshop2_grade": double,
	"final_exam_grade": double,
	"retake_grade": double,
	"n_missed_classes": integer,
	"group": string,
	"n_students_in_group": integer,
	"group_average_grade": double, "state": string,
	"approval_method": string,
	"code_valida": string,
	"id": integer,
	"term": TERM
	}
	
	TERM: {
	"id": integer,
	"year": integer,
	"semester": string,
	"tags": string[],
	"start_Date": date,
	"end_Date": date
	}

GetAveragePartners (V1.0)

Nombre	GetAveragePartners
Versión	1.0
Descripción	This service delivers the averages of the classmates of the student in each of the subjects studied.
URL	server/GetAveragePartners
Parámetros	student_id: integer, curriculum_id: integer, course_id: integer, term_id: integer, group: string

Salida	{
	“data”: double[],
	“Position”: integer
	}

GetStudentHistorySessions (v1.0)

Nombre	GetStudentHistorySessions
Versión	1.0
Descripción	This service provides all the history of the sessions that the student already had with the counselors.
URL	server/GetStudentHistorySessions
Parámetros	student_id: integer

Salida	{
	“historyStudentSessions”: [MEETING],
	}
	
	MEETING: {
	"id": integer,
	“observations”: string,
	“date”: date,
	“counselor”: COUNSELOR
	}
	
	“historyAcademics”: [HISTORY_ACADEMICS_COURS
	COUNSELOR: {
	"id": integer,
	"name": string

	}

SaveNewMeetingObservations (v1.0)

Nombre	saveNewMeetingObservations
Versión	1.0
Descripción	This service sends data to be stored in the database, and delivery as a result of success or error.
URL	server/saveNewMeetingObservations
Parámetros	student_id: integer, counselor_id: integer, observations: string, date: date

Salida	{
	“result”: string
	}

Ejemplo	

3. Frontend: Adaptation of the counseling tool

This section describes the adaptation of the Counseling Tool, focusing on the frontend. Unlike the backend, the frontend does not have a unique generic model due to the different needs that each university has. However, concerning models of visualizations, we have taken as an example, the work done by the University of Leuven due to the systematized work they have done in the development of counseling systems (Charleer et al., 2018, Millecamp et al., 2018), adapting its tools and the previous experience of European projects like ABLE and STELA. The following describes how each university adapted the Counseling Tool, including the design process in connection with the "LALA Framework" as well as the visualization screens and functionalities in each case. In the case of ESPOL, UACH, and U. Cuenca, the Counseling Tool is oriented to the decision of the student curriculum advancement, having general information about the courses, while in the case of PUC the counseling tool is oriented to the decision in specific courses, having specific detailed information of courses.

3.1 ESPOL

The process to design the tool was based on an adaptation of processes that combined Design Thinking, agile methodologies, and human-computer interaction.

- **Step 1: Define the problem**

To understand the problems in our institution, we used the tools of the LALA framework to collect information on the institutional dimension: LALA canvas, eight interviews with institutional leaders, focus groups with seven teachers, four students, in addition to surveys that were completed by 204 students and 24 teachers. The results identified the need to improve the counseling system currently held by the university since 2013.

- **Step 2: Obtain requirements**

Because ESPOL already has a counseling system in place, our approach was to raise requirements regarding what additional visualizations could enrich the counseling session. That is why we decided to ask through a form all the teachers who give counseling. The results were classified and grouped as indicated in figure 3.1.1

Figure 3.1.1 Results of obtaining requirements step

- **Step 3: Build a prototype and evaluate**

    - 3.1 Low Fidelity prototype

We designed a paper prototype where it identified the new sections that the current counseling system would have and where they would be located.

    - 3.2. Mockup

Our mockup, as shown in Figure 3.1.2, was made in Paint. We interviewed nine teachers, asking them whether the visualizations were understood and if the new functionalities were useful or not. It is important to point out that in all the figures, the names used are fictitious.

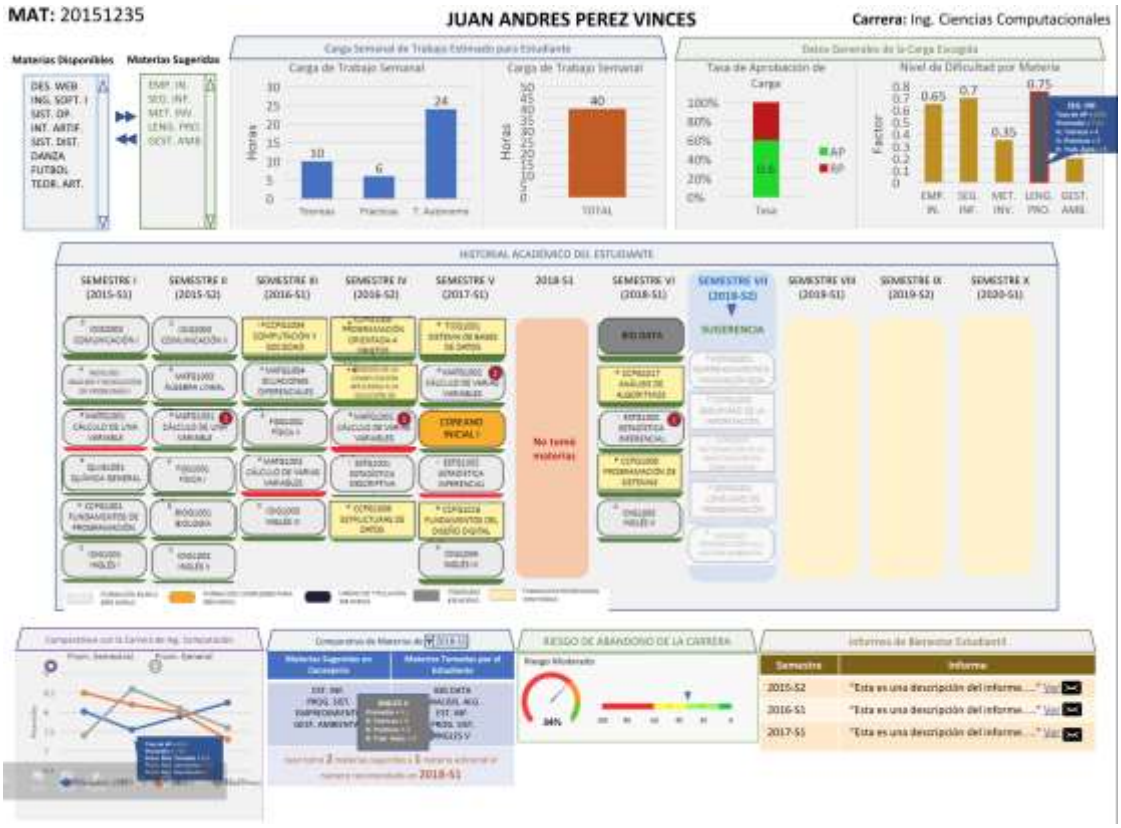


Figure 3.1.2 Mockup

One of the main observations was that the teachers saw much information in a single window. Thus, based on the feedback received, we created a new proposal divided into cards as shown in Figure 3.1.3.

Figure 3.1.3 cards version – high fidelity prototype

- 3.3. Beta Version

To develop the beta version, we interviewed 14 teachers. Each of them was asked to perform different tasks with the tools (e.g., indicate which subjects the student has failed). For this, we define the following usability goals

Homework:	Approved subjects in the academic record
Process:	The user will be asked to enter the academic record option. The user will be asked, how many subjects approved for the second time has the student? The user's response will be recorded by a member of the evaluation team in the "Approved Materials Evaluation Form."
Time:	1 min
Usability goal:	The user can identify the approved subjects for the second time without major effort: ● 90% of users can identify the subject.

Usability component:	Learning

Homework:	Subjects taken in academic record
Process:	The user will be asked to enter the academic record option. The user will be asked, in what semesters did the student take the subject Statistical Tools for Quality? The user's response will be recorded by a member of the evaluation team in the "Material evaluation form taken"
Time:	1 min
Usability goal:	The user can use the mouse-over to see the trajectory of the subject: • 90% of users can use mouser-over. The user can identify the subject taken by the student without major effort: • 90% of users can identify the subject.
Usability component:	Learning

Homework:	Average subject taken in academic record
Process:	The user will be asked to enter the academic record option. The user will be asked, how many students obtained the same average in the subject Industrial Processes? The user's response will be recorded by a member of the evaluation team in the "Subject Matter Evaluation Form."
Time:	1 min
Usability goal:	The user can identify the subject taken by the student without major effort: ● 90% of users can identify the subject

Usability component:	Learning, Memorability

Homework:	Rank qualifications matter taken in academic record
Process:	The user will be asked to enter the academic record option. The user will be asked, what percentage of all the students during the semester who took the subject Industrial Processes obtained the same rank of marks that the student? The user's response will be recorded by a member of the evaluation team in the "Matter range evaluation form taken."
Time:	1 min
Usability goal:	The user can identify the subject taken by the student without major effort: ● 90% of users can identify the subject
Usability component:	Learning, Memorability

Homework:	Suggested materials
Process:	The user will be asked to enter the option available subjects. The user is indicated that the student wants to register in the following subjects: **Production control systems, Methodology for continuous improvement, Management of the sustainable supply chain, Management of groups, Web applications, Pricing policies, and agricultural markets**. The student is doing pre-professional internships for a total of 10H per week. The user will be asked, what subjects would you suggest to the student? The user's response will be recorded by a member of the evaluation team in the "Suggested subject evaluation form."
Time:	3 min

Usability goal:	The user does not recommend the X subjects for complexity: • 90% of users do not recommend the subjects. The user performs the task in less than 2 minutes: • 90% of users do it in less than 2 minutes The user remembers the new features: • 90% of users use the new functionalities
Usability component:	Efficiency

ANSWERS FORM

TYPE	**ANSWER**
approved subjects	
subjects taken	
average subjects taken	
Subjects taken range	
suggested subjects	

Additionally, the teachers completed an adaptation of the system usability scale (SUS) by Brooke (1996). The results indicated how easy to understand the new visualizations was for them.

Survey

	Strongly disagree	disagree	Undecided	Agree	Totally agree
The Available Subjects module is easy to use					
The information shown in the Academic History module is clear					
The information shown in the Statistics module is clear					
I can navigate between the new windows without difficulty.					
I am able to recognize the actions that I can carry out through the new options presented in the counseling system. **Example: See statistics of the subjects, level of difficulty of the subject.**					
I recommend the use of the new options of the counseling system because I consider it useful					
I think I would need the support of a technical person to be able to use the new options of the counseling system					
I found that the new functions were well integrated into the counseling system					
I imagine that most people would learn to use the new options of the counseling system very quickly					

Suggestions:

Thus, the first beta version is shown (Figure 3.1.4 – Figure 3.1.5 – Figure 3.1.6):

Figure 3.1.4 Statistics window - First Beta Version

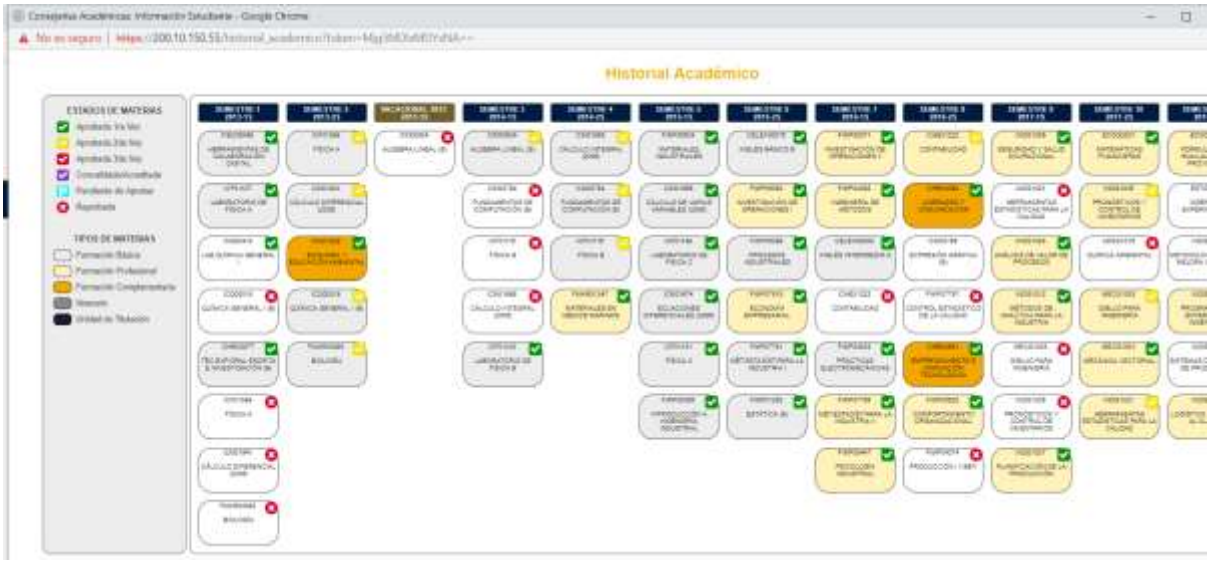


Figure 3.1.5 Academic History- First Beta Version

Figure 3.1.6 Available Subjects - First Beta Version

Finally, after meetings with personnel in charge of the Counseling System: developers and designers, the final beta version is presented below. It consists of 3 windows: Statistics, Academic History, and Available Matters.

Statistics

In this window, one can see a comparison of the student's average with his or her peers from the same cohort and all students who have studied the student's career (Figure 3.1.7).

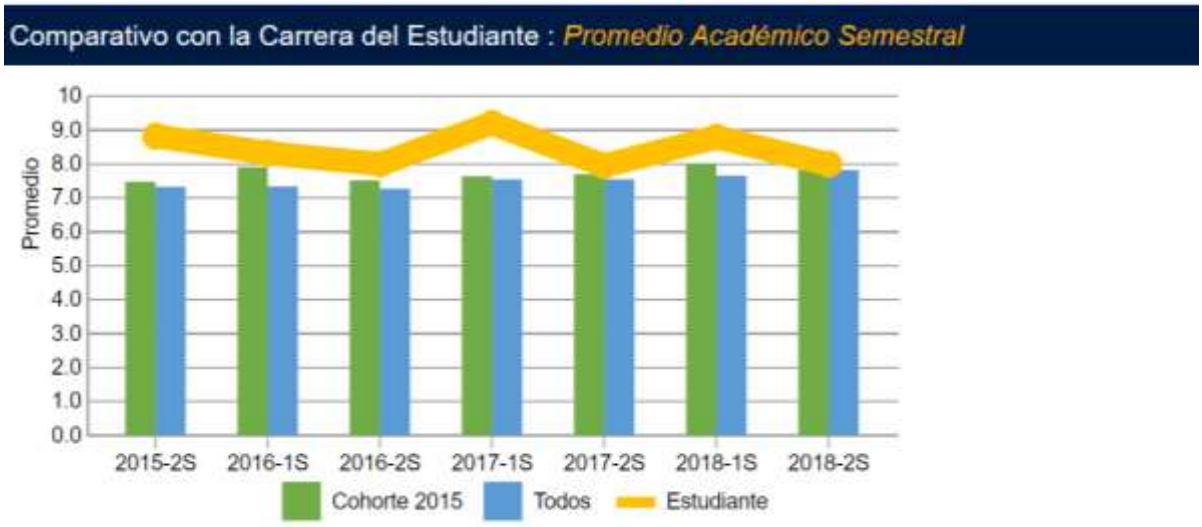


Figure 3.1.7 Comparison chart with the student's career

Additionally, a comparison of subjects can be visualized where one can contrast the subjects suggested by the counselor versus the subjects that the student decided to take and how their performance was. In addition, there is the option to select the comparative history of subjects in the different semesters (Figure 3.1.8).

Figure 3.1.8 Comparison of subjects suggested vs. subjects taken

The last display is a table where the counselor can review the follow-up of a case that has been reported to the Student Welfare Department (Figure 3.1.9).

Figure 3.1.9 Student's follow-up trajectory made by the to the Student Welfare Department

Academic History

In this window, one can see the whole academic history of the student. Additionally, one can click on each subject and review the student's average concerning their classmates and all the students that took the subject in that semester (Figure 3.1.10 y Figure 3.1.11).

Figure 3.1.10 Academic History

Figure 3.1.11 Student's grade comparison between students in the same and different classes during the semester

Available Subjects

Finally, simulations can be made in this window when choosing the subjects that would be taken during the new semester. Additionally, the weekly workload that it would represent (Figure 3.1.12) and the difficulty level (Figure 3.1.13) are displayed.

Figure 3.1.12 Weekly workload

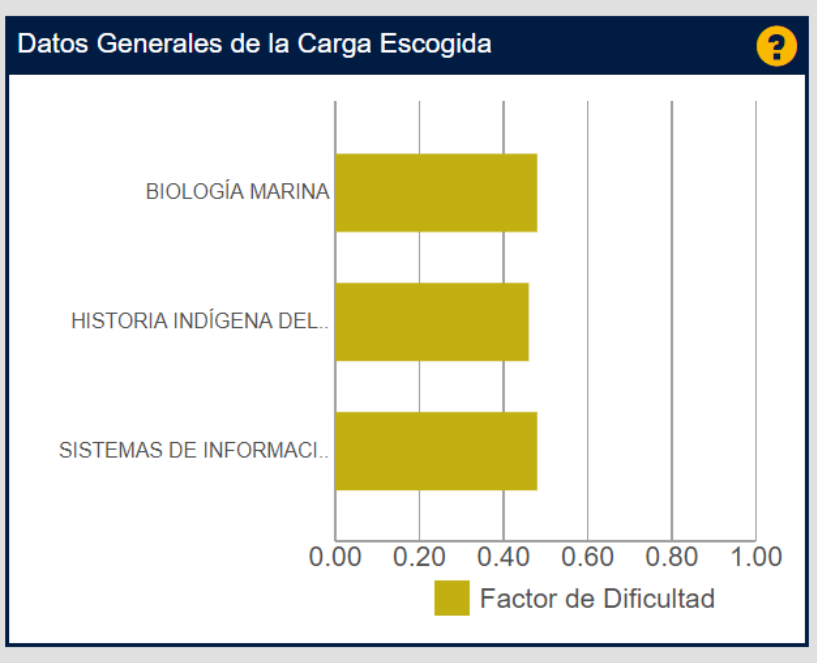


Figure 3.1.13 Difficulty level

3.2 UACh

For the phase of requirements and design, a guide protocol was defined on how meetings should be held with the different career directors of the faculty in order to obtain the requirements to build the proposed application.

This protocol defined meetings with individual users, with a duration of one hour in which questions were asked about the actions performed by them when facing an application for registration, registration, and cancellation of a subject. At the end of the meeting, they were presented with a preliminary design of the visualization panel that was sought to develop for the application process, finally making the capture of possible new requirements.

A total of five requirements gathering meetings were held with the directors of Civil Engineering courses in Computer Science, Civil Engineering in Civil Works, Mechanical Civil Engineering and finally Electronic Civil Engineering. In addition, a final validation meeting of the prototype was held that met all the requirements raised in the previous sessions.

Finally, the requirements and design decisions made with the end users in the meetings described below were evaluated by the LALA team of the UAChand modifications were made by incorporating better usability, look & fell and viability practices concerning the available data and the security policies of the information technology department of the Institution.

First meeting

The LISSA visualization panel was presented to the first user to participate in the definition of requirements, an idea that arises due to the needs that this visualization panel developed at the KU Leuven University meets. This, since in this context LISSA was created to support the tasks of academic counseling, tasks that in the context of the Faculty of Engineering Sciences

is carried out in a very similar way by a school director, specifically when answering before the applications that are sent by students at the moment of registering and / or registering their subjects .

Conclusions of the first meeting

Due to the eventual usefulness of developing a tool similar to LISSA, it is established in this meeting as the first and great advance in the construction of the visualization panel, the requirement to visualize the curriculum of a faculty career instead of qualifications. In exams presented by LISSA. It was also decided to maintain the deployment of the distribution of notes by period that presents in the upper part this visualization panel developed in KU Leuven.

Second meeting

From the conclusions of the first meeting, the following mockup was elaborated (Figure 3.2.1) to guide the second meeting.

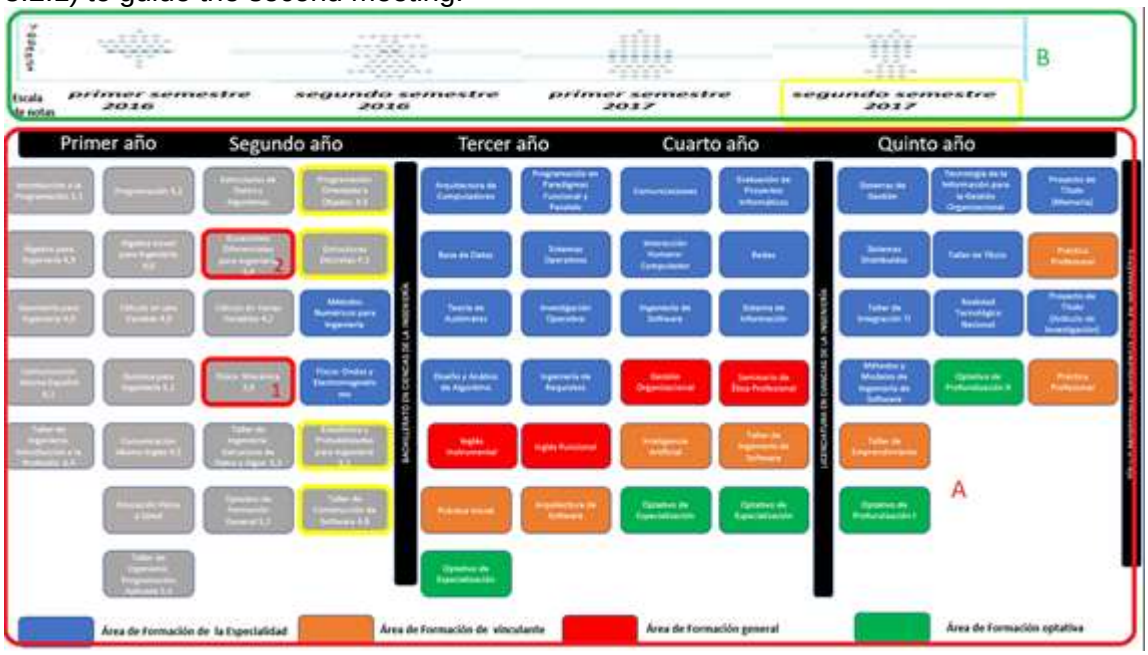


Figure 3.2.1. First elaborated mockup

The original ideas of LISSA for the presentation of course grades in exam periods were used, which in the local context of the faculty correspond to semesters, as well as a section to graph the performance in these periods. It is important to note that this design was chosen considering maintaining the design of the official catalogs that represent the curricula of the careers of the Faculty of Engineering Sciences (https://www.uach.cl/uach/_file/ing - 5b0c7bc00d9d7.pdf). This allows users to feel familiar with the application.

Description of the first prototype

In summary, the application will have two contexts designed to display information from two different contexts, which are: context of the curriculum (A) and context related to student performance (B), as shown in the previous image.

An important need for a school director is to be able to quickly know the progress made by a student, so to satisfy this requirement it was established that in this visualization the subjects already approved will present a gray background color with their corresponding qualification, while failed subjects have a red border, the same color as the number inside that represents the number of times the course has been failed. While, as shown in the same figure, the subjects with a yellow border represent exclusively the subjects studied in the highlighted

period with the same color in the upper part (B), which in the example corresponds to the second semester of 2017

When a user clicks on a course, a graphic will be displayed indicating the distribution of grades of the students who attended the course at the moment in which the applicant student obtained the indicated grade, in a similar way to what LISSA does. In this first meeting, the requirement to visualize historical data of approval of the subjects also arose. When you click on an unfilled course, it acquires a green border, and a graphic is displayed in the lower right corner based on historical data showing the difficulty of the course (Figure 3.2.2)

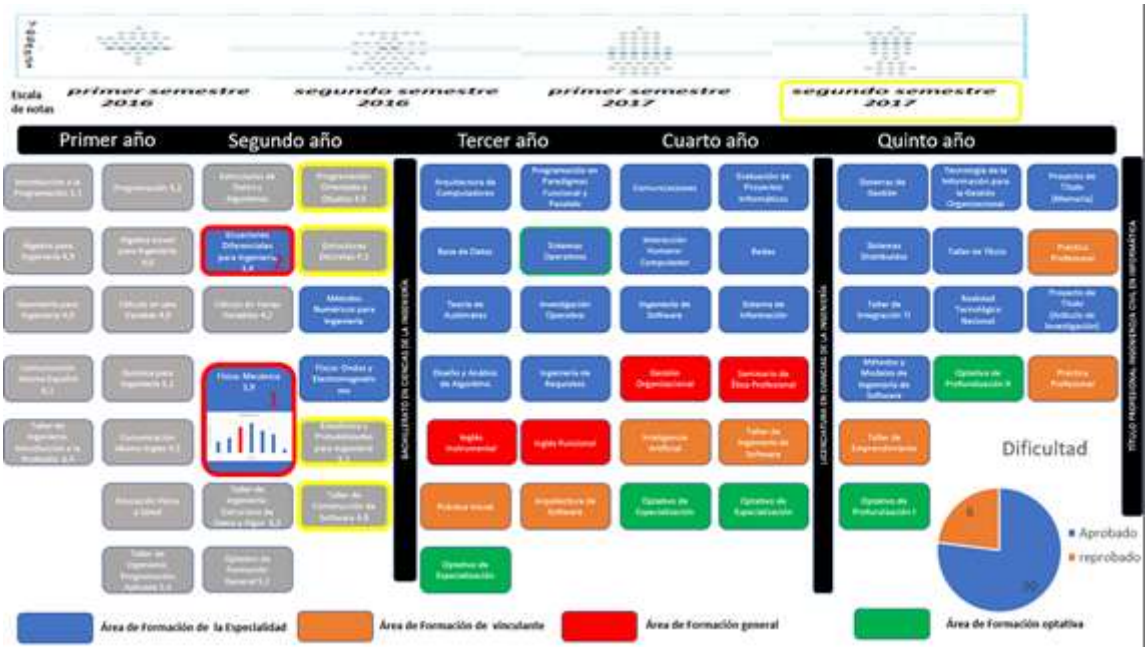


Figure 3.2.2. Visualization of the performance in the course "Physics: Mechanics" and a graph of the historical difficulty of the course "Operating System"

Conclusions of the second meeting:
In summary, in this session the following requirements were obtained: display curriculum, show curricular progress, identify subjects by semester, show evolution of performance, visualize performance in a course, display historical data, indicate number of repetitions of a course and show grades of subjects studied.

Third meeting

In the third meeting a new requirement arose which refers to the incorporation of a section associated with personal data of the student who sends the request, through an image, which allows by means of some action (mouseover or click) to display certain Student data, such as: name, year of admission, city of origin, email, curriculum, etc. It is important to point out that a useful aspect for users is to be able to communicate easily with the student if, before making a decision about the application, it is necessary to discuss certain topics that seem pertinent to the director in a personal way, this through a direct link associated with the student's email. In this way the display panel will have a new section that will include information about the student, so this application will have a total of three sections, which will be: context of the

curriculum (A), context related to student performance (B) and finally information context personnel referring to this one (C).
An aspect not considered in a first design lies in the need to identify annulled semesters, a requirement addressed through identification through a gray background within the periods of study. At this point, a second mockup was developed, which appears in Figure 3.2.3.

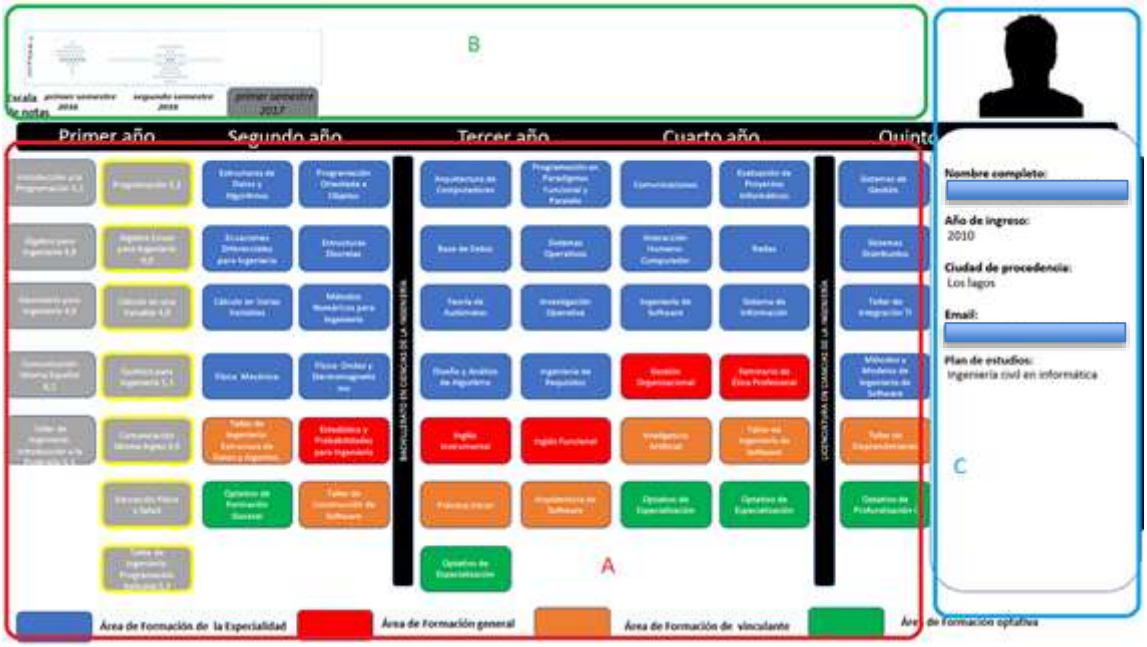


Figure 3.2.3 Second prototype

Fourth meeting

In this session it was finally decided to change the visualization of subjects already taken through the iconography assignment that indicates the final situation of the module, allowing to keep visible to which area of training the subjects belong. On the other hand, unlike the second design of the interface, in this meeting, we chose a scatter plot in section B, since this type of graph serves to highlight patterns or correlations (International Development Research Center, sf), for example, patterns that show good performance. Such points of dispersion will be held together by lines to better clarify the existing correlations.

Conclusion of the meeting

The only requirement that arose in this session corresponds to the identification of canceled subjects, with the consequent differentiation between cancellations made by the student and those made by the respective school management. This is possible to observe in Figure 3.2.4.

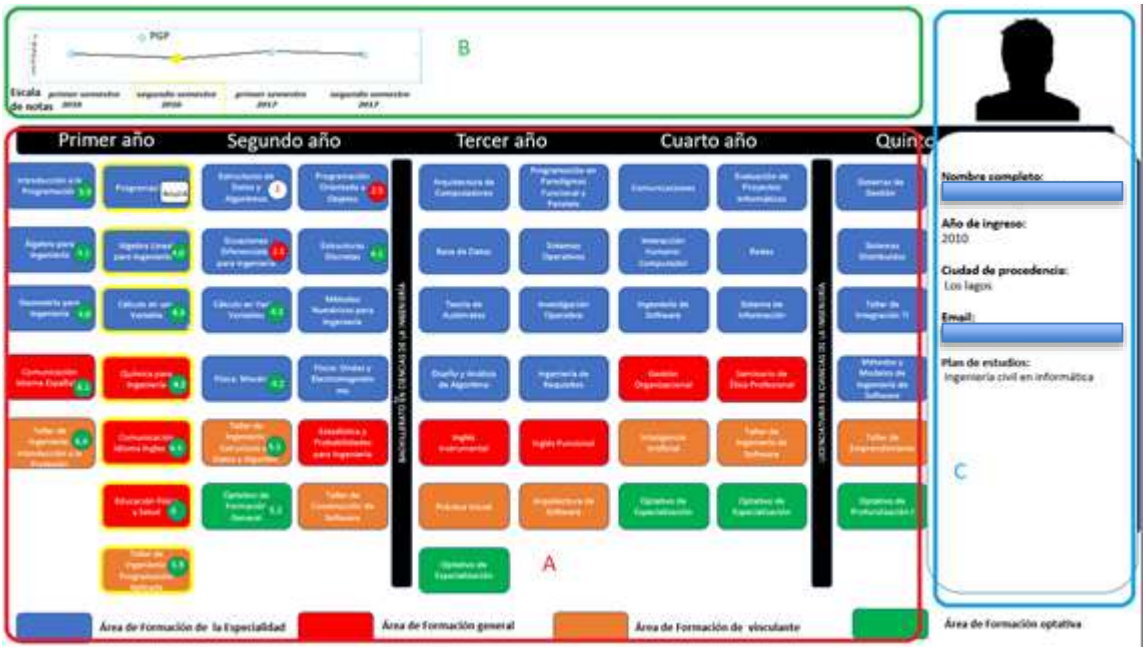


Figure 3.2.4 Third Protoype

Fifth meeting

In this meeting, the latest requirements were defined by the users. In the first place, the need to know the requirements of a subject was established. For this, the " req " icon is assigned to those subjects that are a requirement of a previously selected course. The latter, when clicked, will also show a white circular icon with a green number inside that will correspond to the number of associated credits. In addition, the graph of difficulty of a subject, shown in Figure 4, was changed to a location below the performance graph of a given course, this in order to facilitate users to compare the performance of a course in a period-specific with historical performance. The last elaborated mockup is presented in Figure 3.2.5 that follows

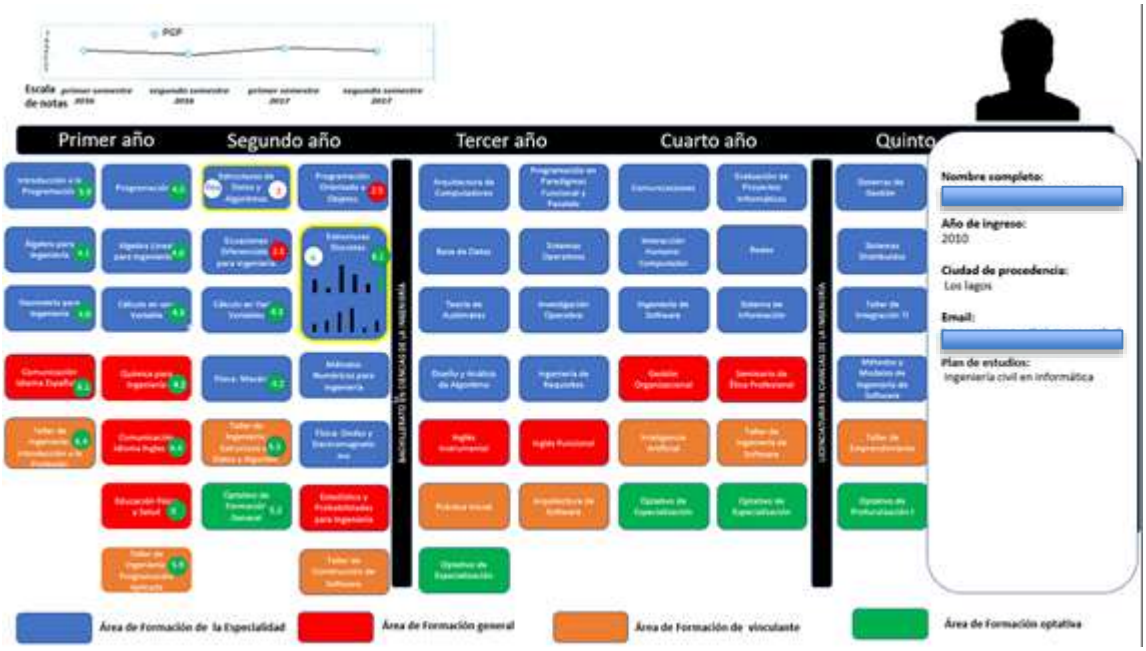


Figure 3.2.5 Fourth prototype

Evaluation stage

For this stage, three school directors were asked to use the application performing specific tasks (tasks adapted to the director who would use the application) to answer a series of questions (Questionnaire). Once the period of use of the application ended and the respective questionnaire was answered, they were asked to complete a survey in relation to the perception they had when using the application. Next, the artifacts used are detailed.

Questionnaire

Next, the questions elaborated in validation process are listed; all the participants could answer them correctly.

1. Can you observe at a glance what has been the student's progress in his / her stay at the university?
2. In what city does the student live?
3. How many semesters has the student canceled?
4. How many subjects does the subject [Name Subject 1] have as a requirement?
5. Can you identify the evolution (positive or negative) of the PGA throughout the student's career?
6. Can you easily determine the subjects taken in a given semester by who sends the application?
7. Is there a course that has been failed more than once?
8. Could you indicate how many subjects the student has canceled throughout his / her career?
9. When the student passed the subject [Name Subject 2], how was the performance of their classmates? The vast majority subject been agreed by? or did he fail her?
10. For the same course of the previous question: Historically: What has been the performance of students who have already completed this subject?

Note: [Name of Subject 1] and [Name of Subject 2] were varying for each of the participants, to contextualize the experience to the corresponding program for each user

Survey results

The survey consists of a series of affirmations, and each one of them has five options that express a greater or lesser degree of affinity with the affirmation. To carry out this survey, the Likert technique was used. The results are detailed in the following table.

Aspect evaluated	**Affirmation**	**Average**
User interface	The application is attractive	5
Usability	The application is intuitive	5
Usability	Accessing the desired information is easy	4.66
Utility	The application helps in decision making	5

After the analysis made by the LALA team of the requirements and the feasibility of available data and in accordance with the privacy and security policies of the University, the application is as follows:

Login page of the application, only for users previously registered by the administrator and complying with the security policies of the UACh (Figure 3.2.6)

Figure 3.2.6 Login page

Application header (logout , help): Displays the name of the program and allows searching by the student's RUT or accessing the help (Figure 3.2.7) :
Displays the student's ID (RUT) (the actual id that the user enters) in an editable text field

- The "Load" button reloads the display panel for the entered RUT
- The date of the last update of the data is shown on the right.
- The help button (?) On the right displays a "balloon" with the legend of the display panel.

Figure 3.2.7 Application header

Curricular Structure: main area of the application that shows in a matrix type layout the structure of a study plan (Figure 3.2.8)
Each column represents one semester of the study plan. In each semester, the courses are listed down.
The semesters are organized in "cycles" that are defined in the study plan. For example, the first "baccalaureate" cycle includes the first 4 semesters of study. The name of each cycle is shown in the black background separators.
The academic information of the student "in counseling" is superimposed on the visualization of the study plan. In this way, the user can visualize the state of academic progress that the student has. The prototype implements this with colored circles that show the student's final grade in each course.
When selecting a subject (click), its block is displayed showing academic information, and the rest of the subjects of the study plan are attenuated.

Figure 3.2.8 Curriculum

Course: The academic information is shown as a colored bar next to it (Figure 3.2.9). Next, it improves the "panning" of all the information, where the columns of subjects have meaning and not the rows. The grade must also be in number, for which it has been widened (with a circle of the same background color) in the position where the rating is located. The background color is one of 4 that represent different ranges of the range of ratings. For UACh
1-3.5 -> #d6604d
3.5-4 -->#a7dc78
4-4.5 ->#a7dc78
4.5-7->#66b43e

If the course has been previously taken, canceled or rejected a smaller diameter as appears aligned with the circle rating. The color of the new circle represents the grade obtained using the same scale. Circles corresponding to canceled subjects are gray.

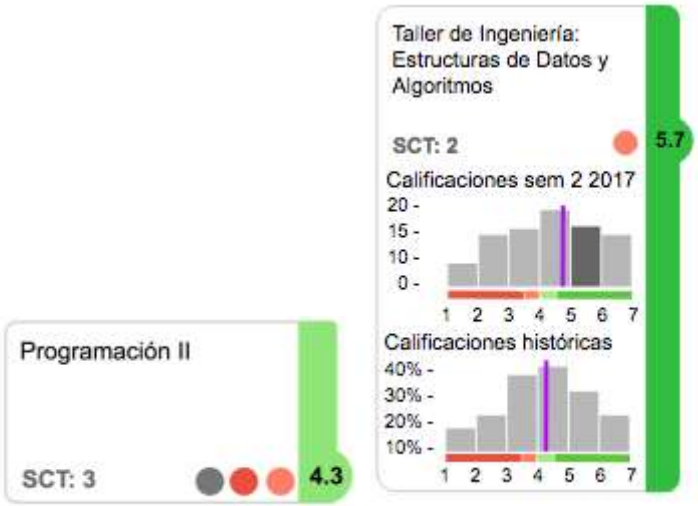


Figure 3.2.9 Course

Graph of average score by term: the graph shows the weighted and accumulated semester average of the student in the semesters he has completed (Figure 3.2.10).

Note that the x-axis does NOT coincide with the columns of the study plan. This has the purpose of distinguishing the semesters that the student has taken, of the "ideal" semesters of the study plan. We emphasize this distinction because it is very rare that the N- th semester of a student coincides with the N- th semester of the program. This is because students fall behind in failing subjects.
PSP and means weighted semi-annual average.
PGA means Accumulated General Average. These averages in time allow to see positive or negative tendencies and the academic history of the student.
The x-axis labels are also buttons that change the view of the bad one. By pressing a specific semester on the x-axis, the student's academic information of "that moment" is displayed, highlighting the subjects that the student was studying in that semester and attenuating the rest of the subjects of the study plan. The following figure shows this.

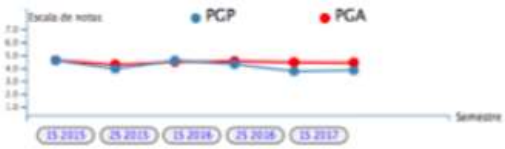


Figure 3.2.10

3.3 UCuenca

This section presents the process followed for designing the student academic performance dashboards that will support counseling activities.

Requirements elicitation

The need to have learning analytics tools that support students counseling activities by allowing to analyze the academic performance of students was identified as a result of executing the Institutional Dimension process suggested in the LALA Framework. Therefore, in this section, we describe the process followed to elicit the corresponding requirements.
In the context of U Cuenca where there is no a student counseling process, and there are no IT tools to support it, the strategy of obtaining requirements was based on taking as a reference the experience of other partners of the LALA Project. Specifically, to obtain requirements, a combination of interviews and focus groups was carried out; where, in both cases, the prototypes of learning analytics tools generated by partners of the LALA Project (ESPOL, UACH, KULEUVEN) that had either experience in student counseling processes or had prototypes of tools were used as support artifacts. The use of these artifacts and the experience of our partners allowed us to: i) Have a common understanding of what learning analytics tools are; helping to convey to stakeholders the concepts of dashboards in the academic context, preventing them from having their interpretation of what a dashboard is (e.g., a report that lists pre-filtered information) ii) Establish a common vocabulary. iii) Evidence that each institution of higher education has its approach and academic requirements, and iii) Obtain information and interaction requirements on learning analytics tools.

Obtaining requirements was divided into two main activities in which we had as main questions: What do you want from a counseling system for the student counseling processes? What information/data would you like it to contain?

Interviews with Faculty and IT Department Directors

The first activity consisted of interviews with directors from U Cuenca whose objective was to involve the directors in the process and obtain their support by showing the benefits of implementing learning analytics tools. Then these stakeholders would be the promoters that would help us to involve other stakeholders, both academics and technicians. The following interviews were carried out:

i) Two interviews, one with the Engineering Faculty dean (faculty previously selected to do the piloting), and another with the faculty sub-dean. Deans and sub-deans make high-level academic decisions in a faculty and delegate responsibilities to career heads and teachers.
ii) An interview with the Director of Information Technology of U Cuenca, who among other responsibilities manages aspects related to IT at the university level, including software and academic databases.

For the interviews, we used a power point presentation based on the one provided by ESPOL, which was a kind of guide to elicit requirements that included their mockups and some questions to guide the process of eliciting requirements. We adapted this presentation by including other questions and screenshots of the high fidelity dashboard prototypes designed by UACH and KU LEUVEN.

We also used the high-fidelity online prototypes (initial versions) implemented by UACH and KU LEUVEN. These prototypes allowed us to not only show the possible structure of academic dashboards, but also possible interactions with them (i.e., what additional information is shown when the user clicks on a visual element). Being able to interact with dashboards and observing how they could be used during a counseling session (a conversation between a counselor and a student) allowed stakeholders to understand better how an academic dashboard supports counseling sessions.

Focus Groups with Career Directors

The second activity consisted of focus groups with engineering faculty career heads and IT department technicians. The objective was to elicit detailed requirements about dashboards for analyzing the academic performance of students; not only information requirements but also interaction preferences.
In the focus groups participated engineering faculty staff (Dean, Sub-dean and 4 career heads), 2 technicians from the IT department and one member of the LALA Team. U Cuenca does not have a department or advisory unit to support students in their academic problems. However, career heads are the ones that students look for in order to obtain assistance regarding their professional career in the curricular structure and other academic issues. On the other hand, IT Department technicians not only know the U Cuenca IT infrastructure, but also its informatics systems and databases. They were included in the focus groups to anticipate the possible lack of data that avoid meeting other stakeholders needs. If data was

not available (or would not be possible to obtain them) to satisfy information requirements, IT staff communicated it to us.
We use the same power point presentation that was used in the interviews with directors, but we had more in-depth discussions about the pre-formulated questions included in the power point presentation, as well as about interaction alternatives (using as examples initial versions of high fidelity prototypes provided by other partners). When analyzing mockups and prototypes, we asked questions like: Is the information shown in this section useful for you? What other information do you need to support conversations with students? What interaction mechanism or visualization technique do you prefer?

As result of the requirements elicitation activity, we got a list of requirements like: The main section of the dashboard should show the structure of the career program (the subjects that a student must complete) instead of showing only the subjects a student has taken in each academic period. The counselor must be able to identify the current status of a student in the career program (approved subjects, failed subjects, subjects that should have been taken but have not been).

Information and interaction requirements

After the requirements elicitation activity, we got a list of information requirements and interaction preferences. This list was refined as a result some iterations with the design activity, obtaining the following requirements specification.

- Any subject that is not available for a student will be presented as blocked.
- For each subject, the name of the subject will be presented with a background color that will depend on the area (Basic, specialization). A subject status bar should show the grade obtained the last time the student took the subject, and the grades obtained in previous attempts. The status bar should allow the reader to differentiate the grade obtained by the student in his/her last attempt from the grades obtained in previous attempts.
- The subject status bar must allow reader clearly identify if in the last attempt (the last time the subject was completed) the student approved, failed, or if he/she is currently taking the subject. In addition, if the student has no completed all the subjects of a term (i.e., semester) but he/she is taking subjects planned for higher terms, the subject status bar must highlight the not completed subjects.
- If a student is currently taking a subject, the status bar must be shown using a different color. The status bar must show the grades corresponding to previous attempts.
- If a student is not allowed to take a subject because it does not meet the prerequisites, it must be represented graphically.
- The text format used to show subject names must vary depending on the subject type. For example, the subjects whose type is optional could be presented in cursive and the compulsory subjects in uppercase.

When clicking on a subject:

- All the other subjects must be disabled.
- The courses that are prerequisites of the selected course and those that depend on it must be highlighted.

- A pop-up window will be presented with the name of the subject, a histogram that locates the student's grade in comparison with the grades of their classmates, and the status bar of the subject.
    - In the case of an optional subject, the name of the subject taken must be shown.
    - If the subject is still not taken by the student, only the prerequisites and dependent subjects will be highlighted.
- If the subject is currently being taken, only the name and status bar with previous attempts grades will be presented in the pop up (the real name in case of the optional ones), the histogram will not be presented. If the user clicks on the popup, the detailed information will be presented, but not the histogram that places the student's grade compared to the rest of his/her classmates.
- If the user clicks on a subject that the student is not allowed to take because he/she did not fulfill the prerequisites, the prerequisites and dependent subjects must be highlighted, but the selected subject must be highlighted showing prohibition (for example, showing its contour in red).

Graph of averages per academic period

A graph should be presented showing the student's average grade and the average grade obtained by his/her classmates (other students in the same group or parallel) per each academic period.

- To obtain the average grade of the classmates, for each subject that the student has taken in an academic period, the average among the classmates' grades (who took the subject in the same academic period and the same classroom group or parallel) that coursed the student will be obtained. Then the average among the averages obtained previously must be calculated.

When you click (or move the cursor over the node) in a node (bar, dot, etc) that represents the average grade of the student):

- All the subjects that make that average will be highlighted in the curriculum, that is, all the subjects that the student studied in that academic period. The mechanism used to highlight the subjects in the curriculum should allow to identify those that the student approved or lost in the corresponding academic period.
- It will be highlighted, in the status bar of each subject, the grade obtained in that academic period.
- Performance information will be presented in the academic period (number of subjects studied and approved, averages of subjects passed and failed). The average obtained by the classmates will also be presented. This information could be presented in a pop-up window.

When the mouse cursor is hovering over a subject that a student has already taken, the academic period in which the student took the subject must be highlighted. If the mouse cursor is hovering over a grade located in the status bar, the corresponding academic period must be highlighted. In the subject status bar, there may be the grades of the last, in case the student has previously lost the subject.

- To highlight the academic period could be highlighted either the descriptive text of the academic period or the corresponding node (point, bar, etc.), this in the "Graph of averages by the academic average."

- When the mouse cursor is hovering over a subject that is being taken, only the current academic period should be highlighted. Not the node that presents information on the average since that information will not exist in the graph.
- When clicking on a node that represents the student's average grade, only the average grade of the student's classmates will be presented.
- In the averages the subjects that are being studied in the current academic period should not be considered.

Graph of subjects taken by academic period

A graphic should be created in which each subject that the student took in each academic period is represented. A subject could have been taken in more than one academic period because the student fails the subject (he has several enrollments in that subject in different periods).

The representation of each subject must show:

- If the student approved or lost the subject in that academic period.
- If the student is taking the course in the current academic period.
- The attempt number (enrollment) of that subject in that academic period.

When you click (or move the cursor) on the representation of the subject

- The corresponding subject will be highlighted in the curriculum. The mechanism used to highlight the subjects in the curriculum should allow to identify if the student approved or lost it in the corresponding academic period.
- It will be highlighted, in the status bar of the subject, the grade obtained in that academic period.

When the mouse cursor is hovering over a subject that has been taken previously, it should be highlighted:

- The representation of the subject in the "Graph subjects studied by academic period" corresponding to the academic period of the last time he attended the subject.
- The academic period in which the student the course.

If you move the cursor over a note located in the status bar of a subject in the curriculum:

- The representation of the subject of the "Graph subjects studied by academic period" corresponding to the academic period in which it obtained that note.
- The academic period to which this grade corresponds should be highlighted. Remember that in the status bar of the subject there may be both the note of the last attempt (last registration) or the note of previous enrollments, in case the student has previously lost the subject. To highlight the academic period, the descriptive text of the academic period in the graph could be highlighted.

If the cursor is moved on a subject of the curriculum that is being studied in the current academic period, it should be highlighted the representation of the subject and the current academic period.

Course planning view for the next semester

The student's curriculum will be presented, but only the subjects that the student is allowed to take the following semester will be enabled. The dependencies (prerequisites) between subjects must be taken into account.

In case the semester is running, only the subjects that the student could have taken in the current semester (in execution) will be shown. In this case, they will be the subjects that they are taking in the current semester plus the subjects that they could have taken in the current semester.

- The information presented in the subjects will be the same as that presented in the student's curriculum view (name of the subject and bar of states).
- In the enabled subjects that the student is allowed to take in the following semester, the subject complexity level must be shown. The levels can be 25, 50 and 75 (which can be represented in text or graphically).
- In the habilitated subjects, the name of the subject will be presented, and in the state bar of the subject, the level of complexity of the subject will be presented.

When the mouse cursor is hovering over a disabled subject, no action will be taken.

When the mouse cursor is hovering over an enabled subject, the following information should be displayed in a pop-up window:

- Complexity.
- The approval rate for the last semester in which the subject was taught,
- Average grade of the last semester in which the subject was given and,
- A number of theoretical, practical and autonomous work hours planned for the subject.

When clicking on an authorized subject, a subject will be increased in the subject graphics, and the repetition number of that subject will be indicated (if it will be taken in 1st, 2nd or 3rd enrollment).

When you click on a previously selected subject, the selection and representation in the graph are eliminated.

When the cursor is hovering over the selected subject in the curriculum, the corresponding representation in the graph is highlighted.

When the cursor is hovering over a representation of the subject in the graph, the corresponding subject in the program structure view is highlighted.

Design

The design process was carried out in several iterations with the career directors, technical staff and members of the LALA Project (both U Cuenca and KU LEUVEN). In the different iterations, the design and the requirements were refined. The participation of members of the KU LEUVEN team was very important in the design activity since their knowledge and experience in both student counseling activities and visualization techniques allowed to produce significant changes in terms of the use of visualization elements. As a result of this phase, two high fidelity visualization panel prototypes were generated: i) Student academic performance dashboard for counseling; designed to support counseling activities and be used in conversations between the counselor and the student. ii) Student academic performance dashboard for teachers; designed to be used by teachers and allow them, based on the analysis of the academic performance of their students, to identify either student that require support or opportunities to improve the teaching process. This last dashboard has not been raised by other universities that make up the LALA Project and its design arose from the deep analysis of the needs of U Cuenca (unlike the first visualization panel that was inspired by the work done in other universities that are part of the LALA Project).

The first iteration of the design

Student academic performance display panel for counseling

Based on the results obtained in the requirements survey, the initial version of the prototype of the dashboard was produced to monitor the academic performance of the students. This dashboard will support the counseling activities and will be used in conversations between the counselor and the student. The overview of the display panel is shown in Figure 3.3.1. It is important to point out that the name used in the figures is fictitious.

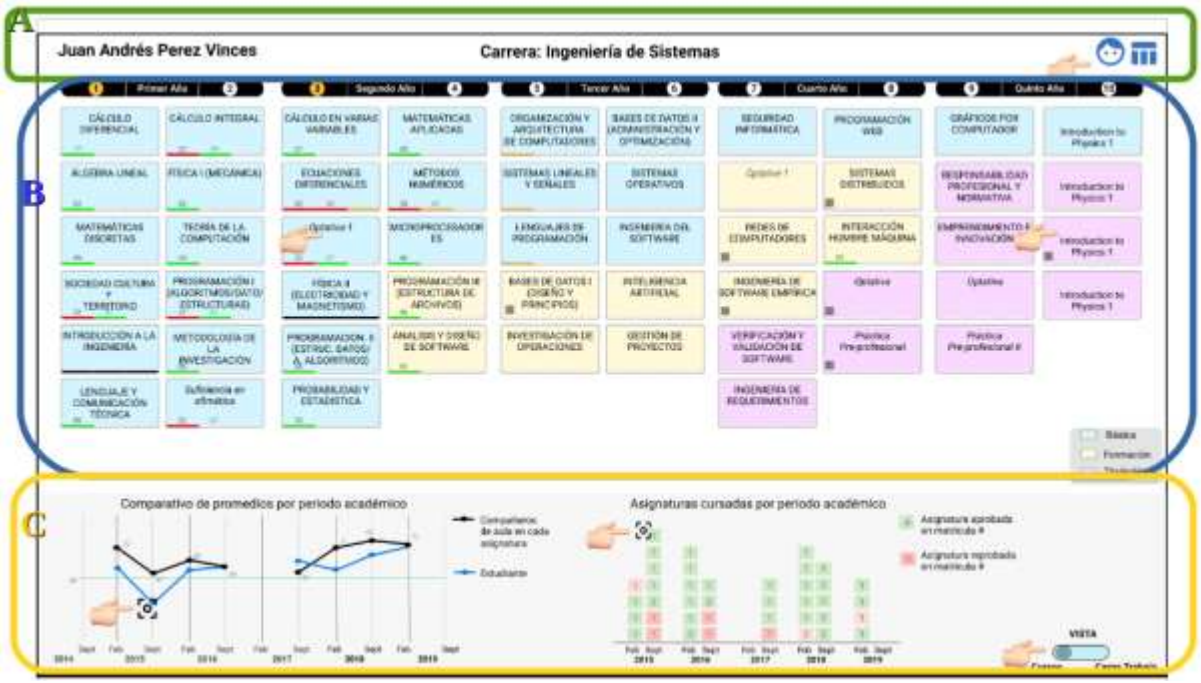


Figure 3.3.1 First Prototype

As can be seen in Figure 3.3.1, the curriculum or curricular program of a student is shown (course planning for each semester and year). Each course is represented by a rectangle

being able to differentiate the courses that the student has taken, has approved, has failed, and pending to take. In addition, for each course taken, the grades obtained in each of the attempts are displayed.

Description of the first prototype version

The application will have three sections or contexts in which it will be possible to appreciate: (A) the student's information context, (B) the context of the study plan, and (C) the context that refers to the student's performance.

For a counseling system, it is very important, above all, to know what is the performance or performance of the student and the progress that has been made since the beginning of his career or from a certain period, in addition to this, it must be easy and quick to identify the current situation of the student in the curriculum, for which, and to satisfy this, it was determined that in this visualization (B) the subjects that the student has already approved will be presented with a green bar with their respective grade, while Failed subjects will be presented with a red bar with their respective qualification. Also, in case of the student have repeated the subject, a bar will be presented for each time he/she repeated the subject with its respective color and grade.

Student Information Context (Section A)

As shown in Figure 3.3.1, this section is practically a menu bar consisting of four sections:

1. The student's full name.
2. The name of the career that student is pursuing.
3. Detailed information icon of the student's grades.
4. Student's Session History icon.

When clicking on this icon (Figure 3.3.2), a popup will appear in which you can see the student's information, as well as the averages of the student during the career and, additionally, a graph indicating the probability of risk of abandoning it (Figure 3.3.3).

Figure 3.3.2 Student Information Icon

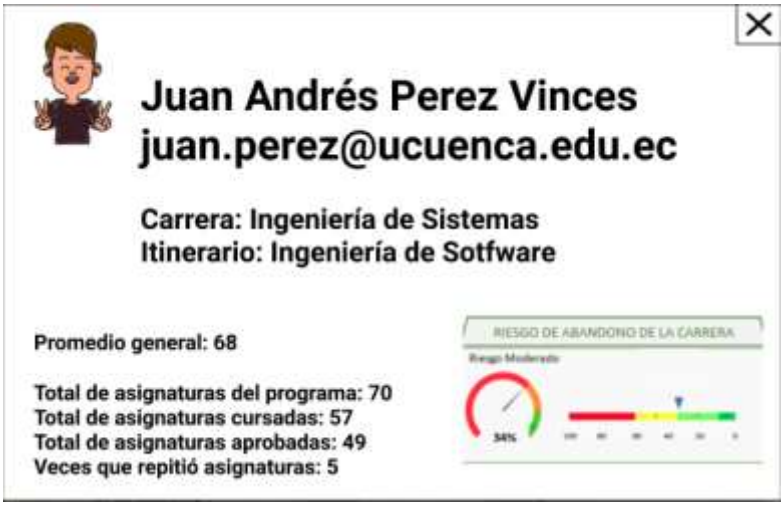


Figure 3.3.3 PopUp Student statistical information

On the other hand, by clicking on the icon of the History of the Counseling Sessions (Figure 3.3.4), it was defined that a history will be shown with all the counseling sessions that the student had previously.

Figure 3.3.4 Counseling Session History icon

Context of study plan (Section B)

This context (Figure 3.3.1), shows or breaks down all the subjects of the curriculum that the student is studying, as well as each of the qualifications obtained in the subjects that have already taken place.

By clicking on any subject in the study plan, the edges of the subjects that are required and the subjects that depend on it will be highlighted immediately (Figure 3.3.5).

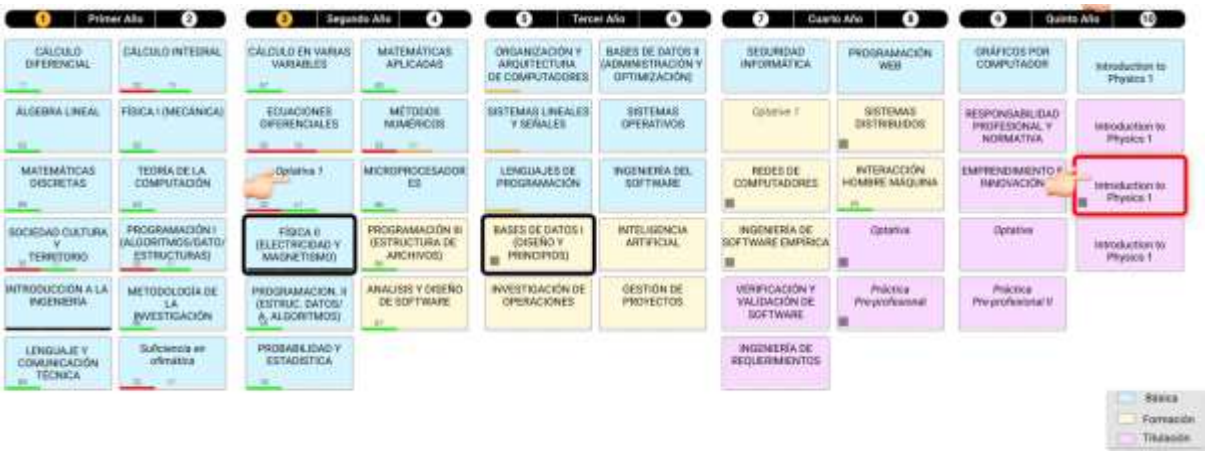


Figure 3.3.5 Study plan - Edge enhancement

Study Plan

Additionally, when clicking on a subject that has already been completed and approved, a popup with a histogram of the comparison of the student's grade with the grades of his / her classmates, as well as the student's grade and a bar at the bottom of said popup with the color of approved or failed according to the last time attended the subject, as shown in Figure 3.3.6.

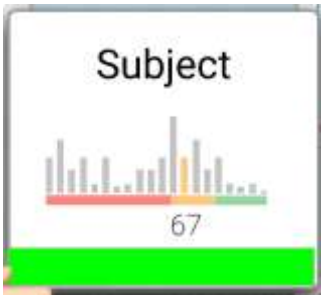


Figure 3.3.6 Comparative histogram student with classmates

The graphic representation of the student's grade is green, while those of their classmates are gray.
If the user clicks on this popup, a much larger popup will appear with the same histogram and with more detailed information about the comparison (Figure 3.3.7).

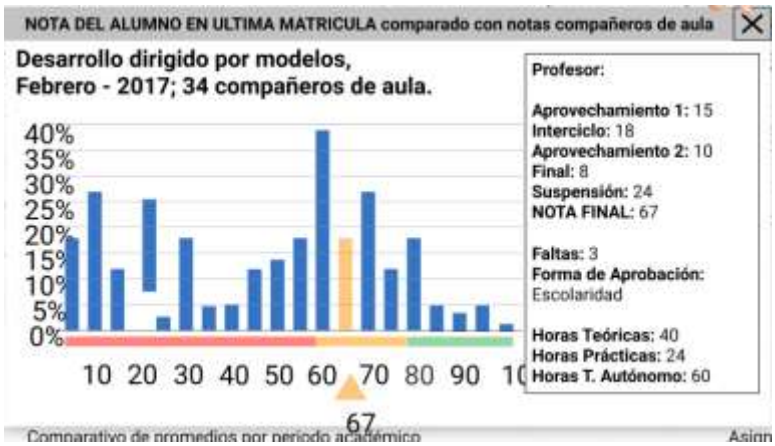


Figure 3.3.7 Detail Student Comparative Histogram with classmates

In this case, the graphic representation of the student is green, while that of his classmates is blue, and the grades and percentages of how many students have the respective qualification are best described. Similarly, on the right side of Figure 3.3.7, you can see more detailed information about the student's grade and the subject.

Student performance context (C)
This context practically shows the graph of student performance during his career. We can differentiate two types of graphs and separate them by sections for a better and faster understanding of the user:

- Comparison of averages by academic period

As shown in Figure 3.3.8, we have a type of line graph in which each point represents a semester taken. The lines and dots of blue color, show the averages of qualifications of all the subjects for each period or cycle that the student attended, while the lines and dots of black color, show the averages of qualifications of all the subjects for each period or cycle that the classmates of the student attended, at the top or bottom of each point of black these averages are shown. With this, we can make a comparison between the averages of the student in each semester with the averages of their peers in the same semesters and classrooms.

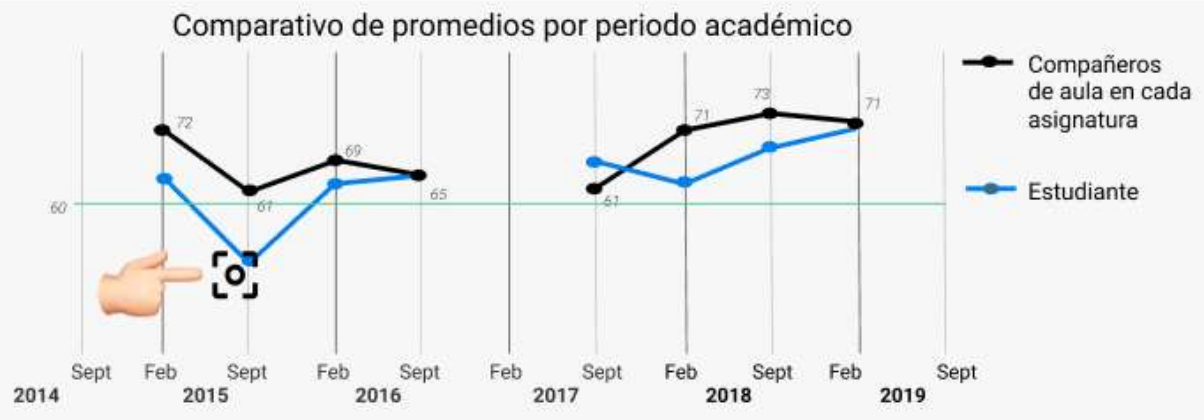


Figure 3.3.8 Comparison of averages per academic period

Also, when placing the cursor over a blue dot, we will see information about the averages of the student, of his classmates, among others; likewise, the edges of the subjects he studied in that semester are highlighted with the red or green color of failure or approval respectively.

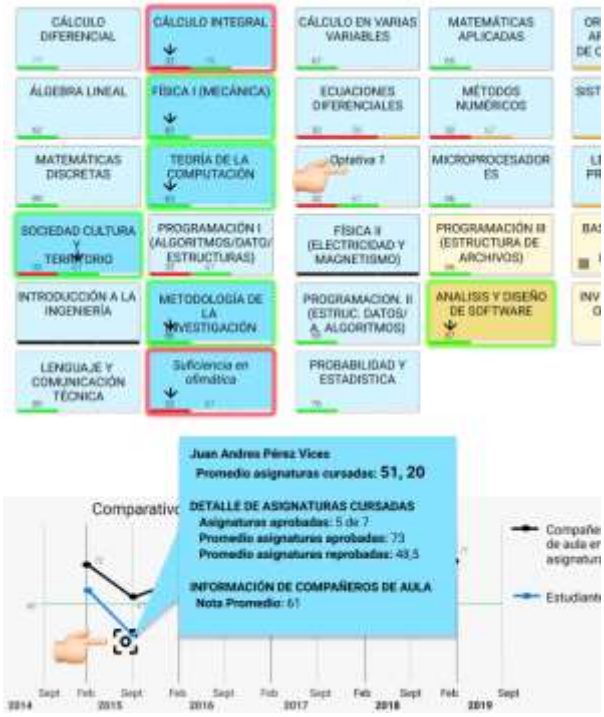


Figure 3.3.9 Student averages information

- Courses Taken by academic period

In this section, we have a graph (Figure 3.3.10) composed of squares that represent the courses a student has taken in each academic period in which he or she has enrolled.
Each of the squares is a course taken, and the position of the same indicates the academic period or semester in which the student studied the subject. Also, we can differentiate whether or not approved this subject according to the background color of the rectangle (green for approved, red for failed) and, also, within it we have a number, which indicates the number of times the subject was taken.

When placing the cursor over any of the rectangles, the corresponding subject in the study plan will be highlighted.

Figure 3.3.10 Subjects taken by academic period

- Weekly Work Load Hours

Finally, in the lower right part of the display screen, we have a switch or switch that allows us to change views from courses to workload and vice versa (Figure 3.3.11):

Figure 3.3.11 Change view to workload

By clicking on this switch, immediately disable all subjects that the student has already approved or cannot take due to lack of approval of subjects requirements, in addition, disappears the graph of **Comparison of averages per academic period** described above and replaces a new one named **Hours of workload Weekly**.

The operation of this new view is simple, since it is a simulator of what courses the student can take the following semester and how many hours of load will have in all the courses you select; then, by clicking on a subject that, if he/she can the next semester, the graph of hours of weekly workload will be updated, adding the hours of the selected subjects. In addition to this, the weekly load hours are categorized by practical, theoretical and autonomous hours, this, in order to provide better understanding to both the user and the student in the counseling session (Figure 3.3.12).

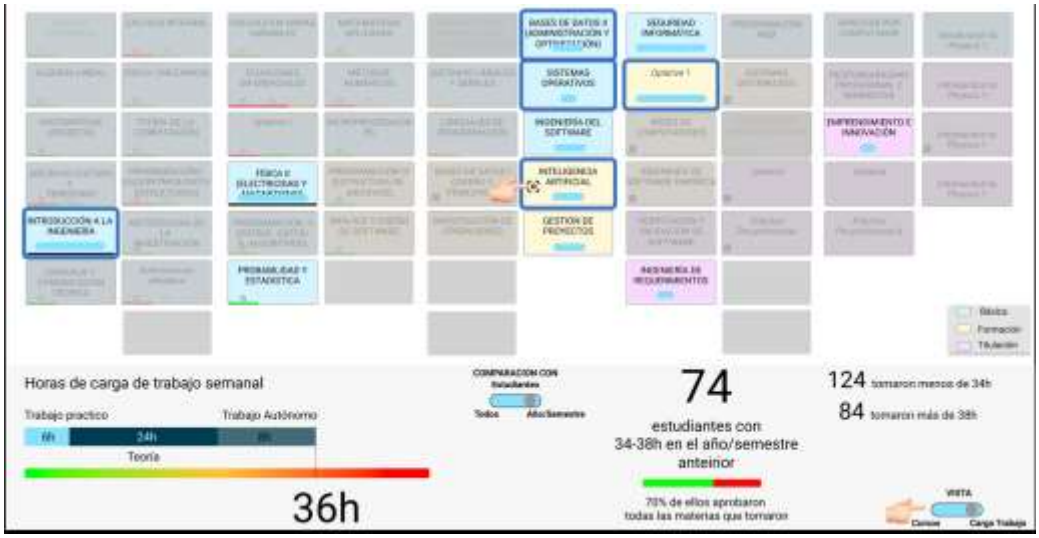


Figure 3.3.12 Workload View

It also shows the number of students with the same workload selected and how many passed and failed subjects for this weekly workload.
Finally, by placing the cursor on a selected subject, a small box will be displayed indicating, among other things, the complexity of the subject.

Student academic performance display panel for teachers

Additionally, and as a very important point, it has been established in these iterations that there is a need to design and implement a tool that is unique for teachers. This visualization panel would allow them to review (through histograms) information about the courses they are dictating, as well as the grades and averages of the students visualizing graphically. This, in order to provide the teacher with a tool in which they can know the performance of either students or the course in general, and keep track of the performance of the courses. The overview of the display panel is shown in Figure 3.3.13. It is important to point out that the name used in the figures is fictitious.

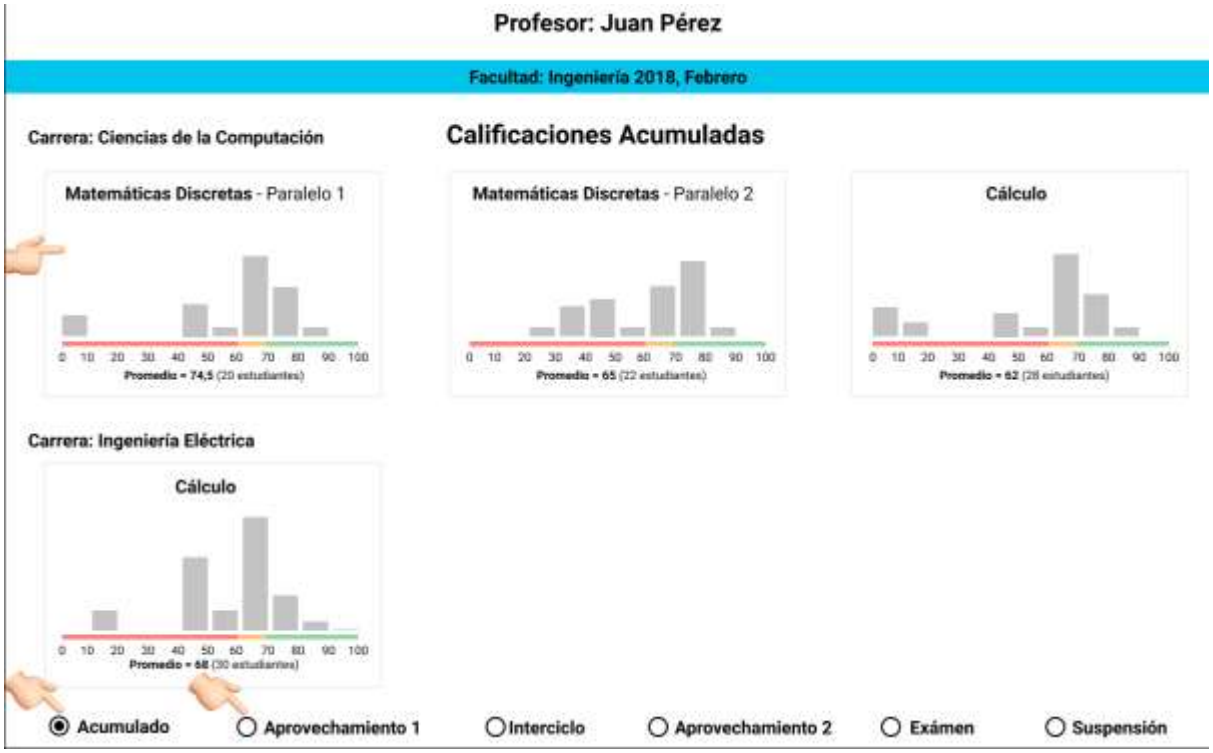


Figure 3.3.13 First prototype display panel for teachers

The tool is made up of two different views for the teacher: main view, in which the teacher can visualize all the courses he is teaching with a histogram for each course in which the averages of each student are indicated and the general average of all the classroom, as presented in Figure 3.3.14.

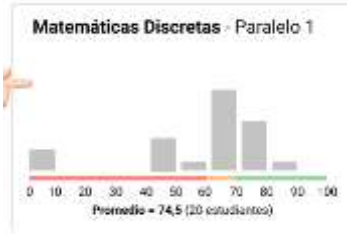


Figure 3.3.14 Histogram Cumulative grades of the students

Also, a selection bar is presented in the lower part of the main view, in which you can choose a variation of the histograms for the grades, which may be: Accumulated, Exploitation 1, Intercycle, Exploitation, Examination and Suspension (Figure 3.3.15).

Figure 3.3.15 Histogram selection bar

In the main view, if the teacher clicks on a histogram, a new view will be presented in which the evolution of the students in the subject is presented through a graph of parallel coordinates, as shown in Figure 3.3.16.

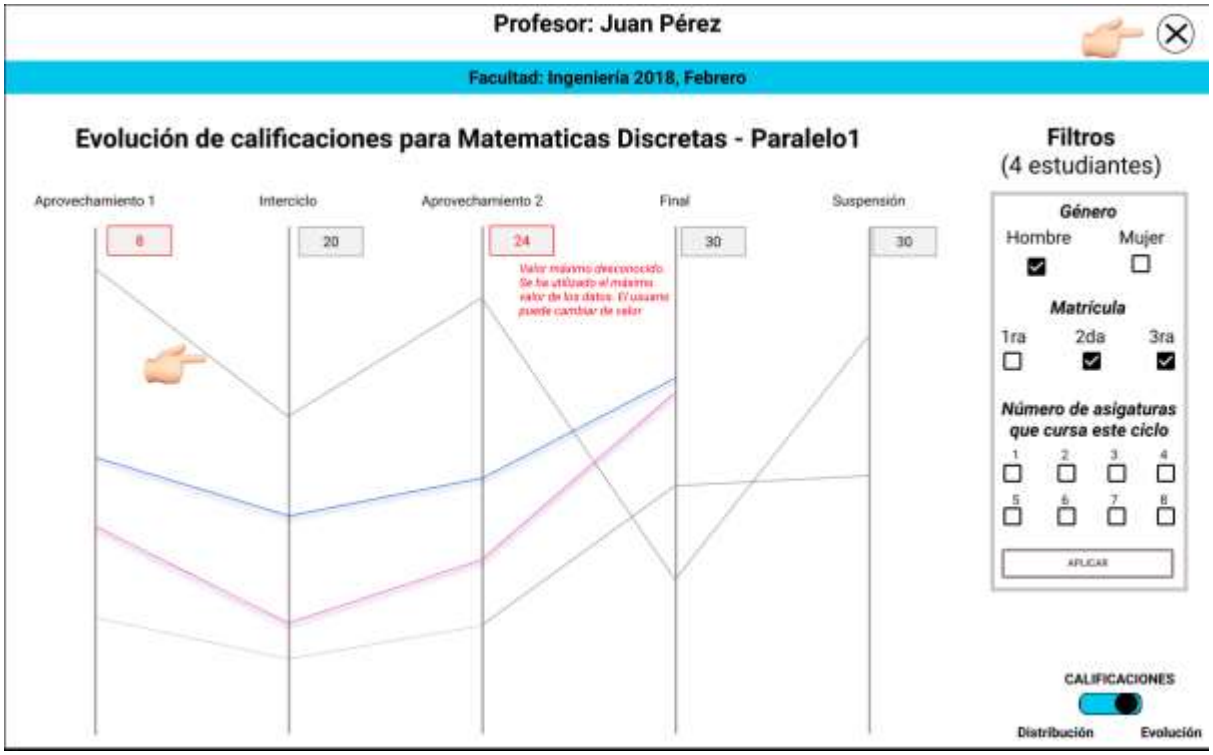


Figure 3.3.16 Evolution of qualifications

In this view, you can also choose or filter the graph according to the student's gender, the enrollment number, and the number of subjects that students are studying in that semester (Figure 3.3.17).

Figure 3.3.17 Rating evolution filter

Conclusions of the first iteration

In this session the following requirements were obtained: Show student information, display curriculum, show curricular progress, identify subjects by semester, show performance evolution, visualize student performance with the whole course, display historical data, indicate number of repetitions of a course, show grades of courses taken, show comparison between the grades of the student and those of their classmates, show a comparison between the student's semester average with that of their classmates, and simulate the load of work in the next semester according to the subjects that the student decides to take. Additionally, an

exclusive prototype for the decent ones was implemented, to visualize the performance of the course through histograms in each subject that it teaches.

Iterations of refinement of design

Once the first prototype of the visualization panels was generated, several sessions were held with the career directors, who at the same time are teachers, other teachers (4 additional teachers) and students (15) to validate and refine both the requirements of information as the interaction characteristics.

Student academic performance display panel for counseling (Figure 3.3.18)

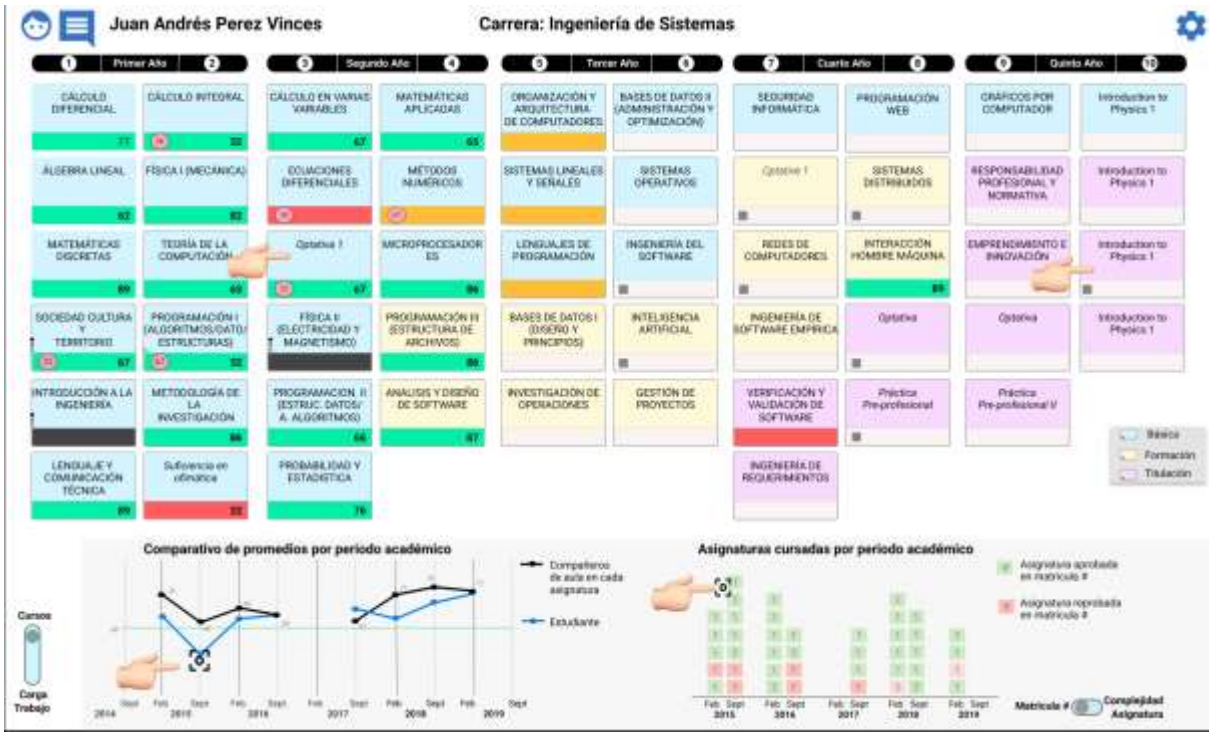


Figure 3.3.18 Second Prototype

In this meeting, first, it was decided to modify the Student Menu Bar somewhat, implementing a new icon called configuration (Figure 3.3.19). The functionality of this icon, among other things, is to show and hide the complexity of each of the subjects.

Figure 3.3.19 Student Information Bar

Regarding the Study Plan Context, Figure 3.3.20 shows how the visualization of the grades and the bars was changed for the times that the subject was repeated. Now, the subjects will have in the lower part a single status bar with the red color approved and the green color

corresponding to the last time the subject was studied. This change was made to facilitate the reading of the student's current status in the curriculum; allowing to identify more easily the current situation of the student in each course. The last grade obtained in the subject can be seen in the lower right part of each subject has passed or not, while the previous grades are in the lower left of each subject in a red circle indicating that failed. Additionally, two more colors were added to the status bar of each subject, the first one is black, which indicates that the student is studying a higher cycle but the subject is from a lower cycle and has not yet completed, and the second color added is the color orange, which indicates that the student is currently studying the subject.

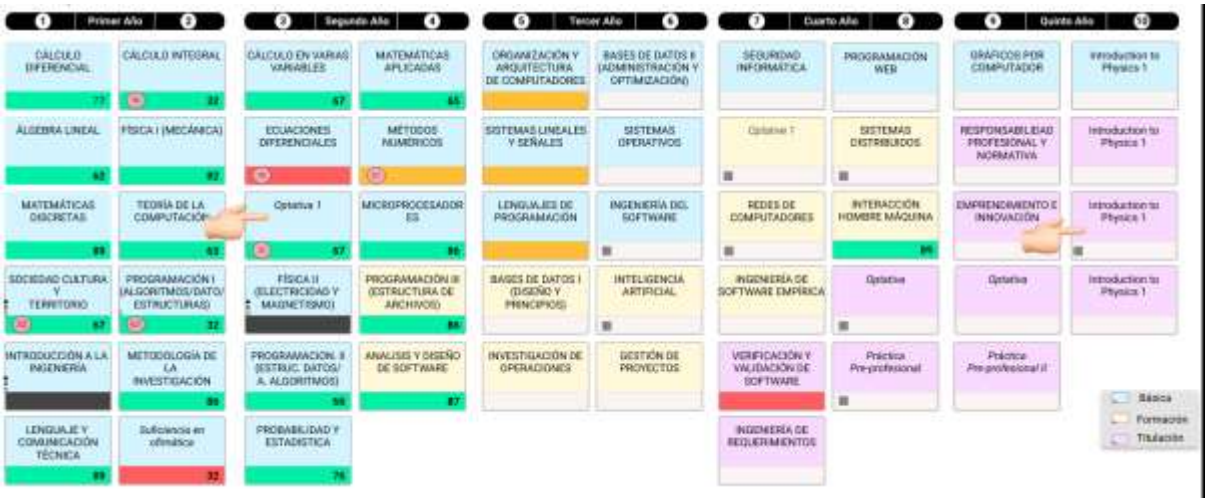


Figure 3.3.20 Study Plan

In the context of student performance, as shown in Figure 3.3.21, a small change was made in the visualization, and a new requirement was added. The change made is to move the switch, or workload view switches from the lower right to the lower left but no longer horizontal, but vertical, this because of the new requirement.

Figure 3.3.21 Switch change to workload view

The new requirement was added in the position in which the change switch was in workload view:

Figure 3.3.22 Switch change to complexity

As shown in Figure 3.3.22, this switch will serve to change between complexity and number of enrollment for the graph Subjects taken per period, as can be seen in Figure 3.3.23.

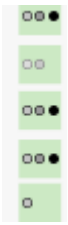


Figure 3.3.23 Complexity view

The complexity is represented as follows:

- A white circle with a black border: 25%.
- A gray circle with a black border plus a white circle with a black border: 50%.
- A black circle plus a gray circle with a black border plus a white circle with a black border: 75%.

Student academic performance display panel for teachers

Finally, regarding the tool for the teacher, a validation has been carried out with the experts and the visualization has been modified minimally, so that it is a little more user-friendly. The changes made are how the information on the number of students and the general average of the course are presented in the histograms (Figure 3.3.24).

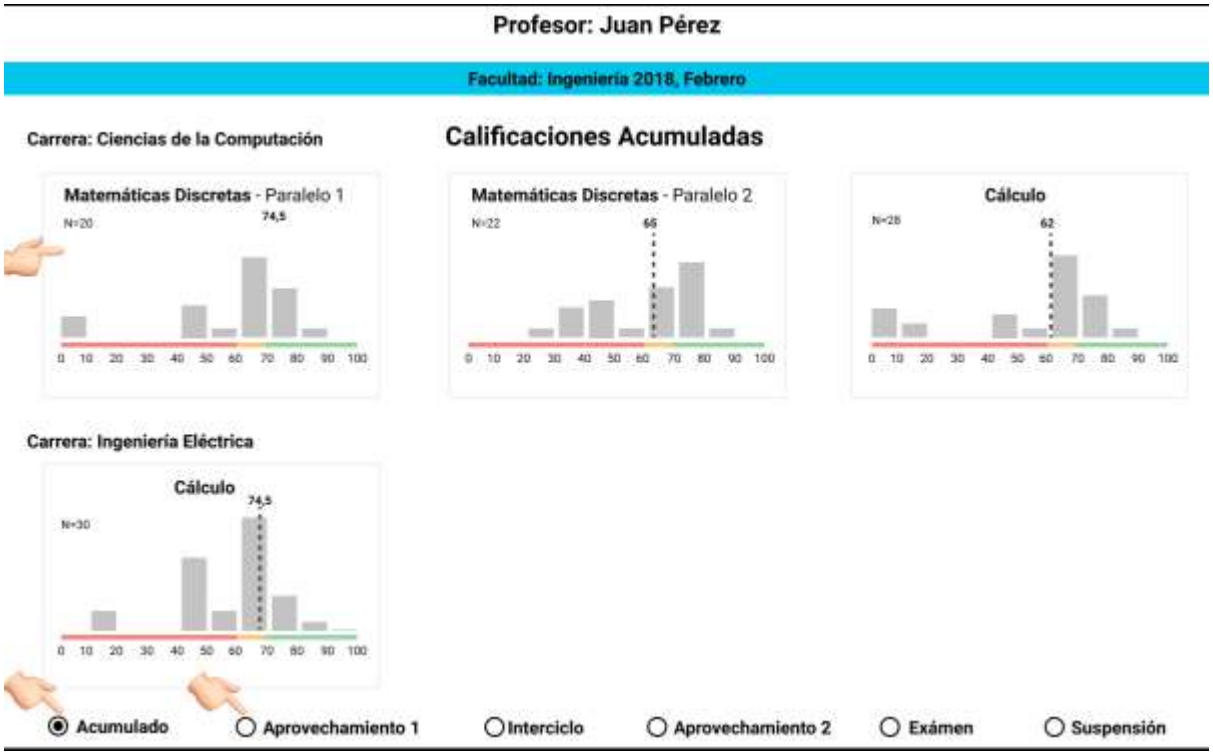


Figure 3.3.24 Second prototype tool for the teacher

Conclusions of Design Refinement Iterations

In summary, in this meeting a configuration icon was added, the status bars of each subject was changed by a single one and the visualization of the qualifications, the change switch was changed from view to workload and a new switch to change the complexity of the subjects describing each complexity. Finally, the main visualization of the visualization panel for teachers was modified.

3.4. Pontificia Universidad Católica (PUC): NoteMyProgress

This section provides a detailed description of the NoteMyProgress (NMP) tool. NoteMyProgress is a mentoring tool designed to support students' self-regulation strategies in online courses in an automatic and personalized way (http://tech4dlearn.com/index.php/notemyprogress/). Through interactive visualizations, it offers actionable aggregated information about students' activities in an online course and its interaction with its contents. The objective is to promote students' reflection on their learning strategies in order to motivate informed decision making to improve their performance. Specifically, the tool has functionalities to support the following learning self-regulation strategies: (1) goal setting and strategic planning, (2) time management, (3) self-monitoring, (4) self-assessment, and organization (note-taking). All of these strategies are related to the self-learning capacity that students are expected to develop throughout their careers to become lifelong learners.

The NMP tool was initially designed for the Coursera platform. However, its architecture is easily adaptable to any other Learning Management System (LMS), such as Moodle, in order to support traditional online courses or blended learning practices. Specifically, the tool consists of a web platform and a plugin for Google Chrome. The plugin collects the student's activity on the LMS and gives the student the option to take notes while studying the course. The web platform offers the visualization of the student's activity in a graphical and interactive way to facilitate the monitoring of their activities.

This section describes NMP and its design process in detail. The first sub-section presents the architecture of the NMP tool; the second sub-section describes the backend of NMP; the third section describes the functionalities and interface of NMP (Frontend). Finally, the last subsection details the design process followed to obtain the final version of the tool.

Architecture: Overview

NoteMyProgress is a tool designed to complement any Learning Management System (LMS) and takes advantage of the learning and administration functionalities they offer. NMP has two main components: **(1)** a plugin developed in Javascript (Google Chrome), which collects information about the learning activities of students in the LMS platform; and **(2)** a dashboard developed in Ruby (2.3.1) on Rails (5.1.3), which analyzes the collected data and creates interactive visualizations (d3.js version 3) that help the user to follow his learning process. Figure 3.4.1 presents the NoteMyProgress architecture.

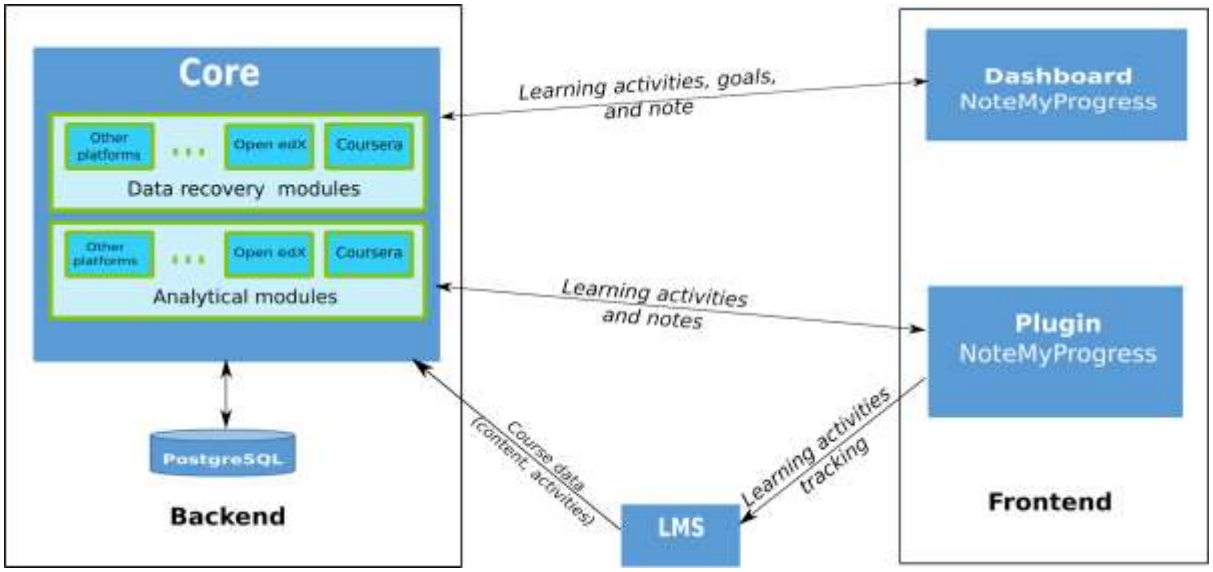


Figure 3.4.1. NoteMyProgress Architecture.

Once a student starts his working session in the LMS, the NoteMyProgress plugin tracks the URLs visited and stores them to identify the activities the student performs on the LMS. The URLs are sent to the NoteMyProgress dashboard, where they are analyzed and decoded in the modules corresponding to the source LMS. In other words, the URL is treated in order to identify which course activity it corresponds to and thus, be able to track the student's activities in the LMS without necessarily embedding NMP into the LMS platform. The student data model is stored in a PostgreSQL 9.5 database. The NoteMyProgress dashboard uses the information from the student model and displays the information to the students. In addition,

the dashboard offers a form functionality that allows students to define and monitor their own learning goals in the course. Finally, the NoteMyProgress plugin also has a functionality that allows students to take notes during their study session in the LMS. These notes can then be managed and printed from the NoteMyProgress dashboard.

NoteMyProgress uses two types of data sources: **(1)** NMP log files and **(2)** external APIs (collect external LMS data). Logfile data is stored through the NoteMyProgress Plugin when the student initiates a study session in the LMS. This information is supplemented by information on the progress of the student's activities in the course offered by the LMS.

Data retrieved from these data sources are classified and stored in the database. Indicators are then generated and used to visualize data related to students' actions in their study sessions in the LMS. For example, for an LMS such as Coursera, the indicators are: number of videos initiated and completed, evaluations initiated or completed, sessions completed, time invested in study sessions, among others. These indicators are organized by objectives to offer students a visualization of their commitment to the course, their performance, and their efficiency during their work sessions.

The first version of NMP was designed for Coursera's MOOC platform, but its architecture could be adapted to any other LMS. The following are the necessary considerations to install this first version of NMP to institutions that use Coursera as LMS. A more complete description of the features of the NMP databases is provided below to be able to adapt it to any other LMS.

Hardware required:

- 1 server (32 GB memory, 256 disk space) to host the web application and database. Depending on the concurrence of users, a dedicated server for the database should be considered.

Software required:

- Ubuntu server 16.04.
- Ruby 2.3.1
- Rails 5.1.3
- PostgreSQL 9.5
- Puma web server (latest version)
- Nodejs (Javascript Runtime)
- App courserareachexports (https://github.com/coursera/courseraresearchexports). It is used to request and download the reports of each course to update the activities completed by the students.
- NoteMyProgress tool (Plugin, Dashboard)

Other requirements:

- The user account in the LMS (depending on the LMS)

Personnel required:

- Specifically, the implementation and maintenance of NMP require a technician of the institution in charge of managing servers and web platforms. The person in charge of the tool has the following tasks: downloading LMS reports, updating data from reports, initial loading of course information, among others.

Test Plugin.
The NoteMyProgress plugin can be downloaded and installed from the Google web store. However, the plugin is not publicly available and can only be accessed via a direct link, since it should work only for courses for which it has been adapted to. That is, each institution must adapt the source codes available in the NMP GitLab repository (see next Sources Repository section) and vary the content referring to the courses. This change allows that, when students download the plugin, they can visualize the courses offered by the institution. In the following link, one will find a sample plugin adapted to some of the MOOCs courses offered by the Pontificia Universidad Católica de Chile in the Coursera platform: https://chrome.google.com/webstore/detail/notemyprogress/aghbcfhpjnmgkgafdbcaljgegcimcmng?authuser=2.

Access to source repositories.
The source codes of the NMP Plugin and the display panel are available from two separated repositories at https://git.cti.espol.edu.ec/LALA-Project/PUC

NMP Backend

The NMP backend is formed by the database and a web processing engine that allows storing and classifying the URLs captured by the NMP plugin during the student's study session in the LMS. This section introduces the NMP database model. Some of these tables should be modified if you want to adapt the tool to another LMS different from Coursera. The tables to be modified are indicated in the description of the database model presented below.

Once the tool has been adapted and deployed in its context, the loading and extraction of data (courses, previous editions, materials) for later analysis is done through scripts or instructions executed manually. To extract the information from the courses and activities included in the course design, we have developed a script that can be downloaded from the following link:
https://drive.google.com/open?id=1YY_tVCxRi-z8GBkqW90P8fqnIGgQGDph.

This script is responsible for creating three files with .csv extension: a file with information about the learning activities, a file with information about the different editions of the course and a file with information about the students registered in the course (it takes the students from the beginning of the course). In addition, we have developed two scripts that are responsible for creating .csv files with information about the sessions and activities carried out by all the students who took the course in previous editions, which can be downloaded in the following link:
https://git.cti.espol.edu.ec/LALA-Project/PUC/tree/master/scripts%20NoteMyProgress

NoteMyProgres tool database
The student data model is stored in a PostgreSQL 9.5 database. The current data model collects real-time information on the activities performed by the student during a study session (sequential activities with a time difference of less than 30 minutes) and information provided by the LMS (courses, student events in the current and previous editions). Figure 3.4.2 shows the database model defined for NMP. Below, we give the details of the most relevant tables of the data model. Some of these tables should be modified to adapt to an LMS other than Coursera.

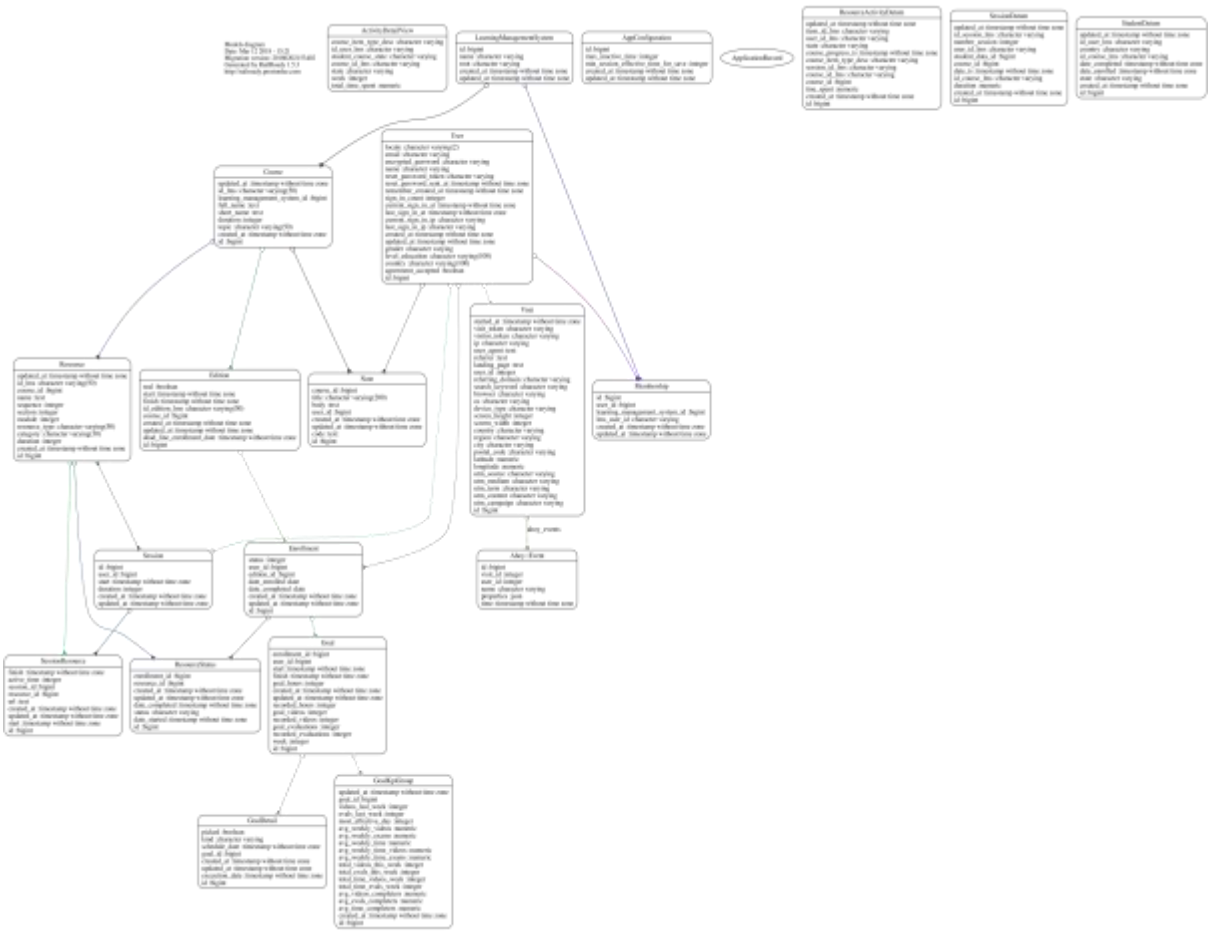


Figure 3.4.2. NMP Database Data Model

Table: LearningManagementSystem
Description: This table stores the LMS data where NMP will be displayed. Currently, this table is configured for the LMS of Coursera, but the fields were defined generically so that, just by changing its content, the tool could be deployed in any other web-based LMS, such as edX, Open edX, MiriadaX, Moodle, Blackboard or Canvas.
Fields:

Name	Data type	Description
Id	bigint	The field used as a foreign key for the *LearningManagementSystem* table.
name	character varying	Indicates the name of the LMS supported by NMP.
root	character varying	Indicates the base URL of the LMS on your web page.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: Membership
Description: This table stores the data of the LMS in which a student is enrolled, as well as the unique identifier in that LMS.
Fields:

Name	Data type	Description
Id	bigint	The field used as a foreign key for the *Membership* table.
user_id	bigint	The field used as a foreign key for the User table.
learning_management_system_id	bigint	The field used as a foreign key for the LearningManagementSystem table.
lms_user_id	character varying	Indicates the unique id given to the student on a learning platform.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: Course
Description: This table stores the information about the LMS courses in which you want to use the NMP tool for students' support. This table must be modified each time you want to use NMP in a new course. Specifically, the characteristics of the course such as its name, the number of weeks of duration or its identifier in the LMS where NMP is displayed must be indicated.
Fields:

Name	Data type	Description
Id	bigint	The field used as a foreign key for the *Course* table.
Id_lms	character varying	Stores the unique course identifier given by the learning LMS where the course is offered.
learning_management_system_id	bigint	The field used asa foreign key for the *LearningManagementSystem* table.
full_name	text	Stores the full name of the course.
short_name	text	Stores an abbreviation of the course name.

duration	integer	Stores the number of weeks the course lasts.
topic	character varying	Stores the domain or learning area addressed in the course.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: Edition
Description: This table stores the data about the course editions in which NMP is displayed. That is, a course can have several editions, with different students in each edition. This table stores this information in order to be able to use NMP in different editions of the same course.
Fields:

Name	**Data type**	**Description**
Id	bigint	The field used as a foreign key for the *Edition* table.
Id_editions_lms	bigint	Stores the unique identifier for each edition by the learning LMS.
course_id	bigint	The field used as a foreign key for the *Course* table.
start	timestamp	Stores the date you start editing the course.
finish	timestamp	Stores the date on which the course edition ends.
dead_line_enrollment_date	timestamp	Stores the deadline for users to enroll in the course edition.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: user
Description: This table stores data about users using the NMP tool. Specifically, this table contains the information related to the user in order to access the demographic information tool for statistical analysis.
Fields:

Name	**Data type**	**Description**

Id	bigint	The field used as a foreign key for the *User* table.
email	character varying	Stores the e-mail with which a student registers in NMP.
encrypted_password	character varying	Stores the user's encrypted key.
name	character varying	Stores the user's full name.
gender	character varying	Stores the user's gender.
level_education	character varying	Stores the user's level of education.
country	character varying	Stores the user's country of origin.
agreement_accepted	boolean	Indicates whether the student accepted the informed consent to use the data.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: Resource
Description: This table stores data about the learning resources or activities to the course. This table changes for each course and must be adapted to its content. The contents of this table must be adapted manually for each new course in which NMP is going to be used.
Fields:

Name	**Data type**	**Description**
Id	bigint	The field used as a foreign key for the *Resource* table.
Id_lms	character varying	Stores the unique identifier assigned to the resource by the learning LMS.
course_id	bigint	The field used as a foreign key for the *Course* table.
name	character varying	Stores the name of the resource.
sequence	integer	Stores the sequence number of the resource in a learning session.
section	integer	Stores the section number to which the resource belongs.

module	integer	Stores the module or week number to which the resource belongs.
resource_type	character varying	Stores the resource type (Lecture, exam, peer review, programming, quiz, supplements)
category	character varying	In this development, it fulfills the same function as resource_type.
duration	integer	Stores the time to resolve the resource assigned by the teacher to the resource.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: Note
Description: This table stores data about the notes taken by students. The notes will always be generic, regardless of the course in which NMP is displayed.
Fields:

Name	**Data type**	**Description**
Id	bigint	The field used as a foreign key for the *Note* table.
course_id	bigint	The field used as a foreign key for the *Course* table.
user_id	bigint	The field used as a foreign key for the *User* table.
title	character varying	Stores the title assigned by the student to the note.
body	text	Stores the text of the note taken by the student.
code	text	Stores the complete HTML code with the text of the note.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: Enrollment
Description: This table stores data on students enrolled in a course edition.
Fields:

Name	**Data type**	**Description**
Id	bigint	The field used as a foreign key for the *Enrollment* table.

user_id	bigint	The field used as a foreign key for the *User* table.
edition_id	bigint	The field used as a foreign key for the *Edition* table.
status	integer	Indicates whether the student passed (1) or failed (2).
date_enrolled	date	Stores the date the student enrolled in the course edition.
date_completed	date	Stores the date the student completed the course edition.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: Session
Description: This table stores the data about the study sessions of students in a course. This table allows retrieving information related to the sessions, such as the time spent in the working sessions and the day of the week in which students' spent more time working on the course.
Fields:

Name	**Data type**	**Description**
Id	bigint	The field used as a foreign key for the *Session* table.
user_id	bigint	The field used as a foreign key for the *User* table.
start	timestamp	Stores the date and time the study session started.
duration	integer	Stores the time in minutes that the session lasted, which is calculated according to the time in each activity.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: SessionResource
Description: This table stores data about the resources a student used in a study session.
Fields:

Name	**Data type**	**Description**
Id	bigint	The field used as a foreign key for the *SessionResource* table.
session_id	bigint	The field used as a foreign key for the *Session* table.

resource_id	bigint	The field used as a foreign key for the *Resource* table.
active_time	integer	Stores the time in minutes that the student invested in the resource.
URL	text	Stores the time the URL visited by the student to access the resource.
start	timestamp	Stores the date and time you logged on to the resource.
finish	timestamp	Stores the date and time when the resource was no longer used.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: ResourceStatus
Description: This table stores the data on the goals set by the students for each week.
Fields:

Name	**Data type**	**Description**
Id	bigint	The field used as a foreign key for the *ResourceStatus* table.
enrollment_id	bigint	The field used as a foreign key for the *Enrollment* table.
resource_id	bigint	The field used as a foreign key for the *Resource* table.
date_completed	timestamp	Stores the date and time the student completed a resource.
status	character varying	Stores the value that indicates whether the student completed or started a resource.
date_start	timestamp	Stores the date and time the student initiated a resource.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: Goal
Description: This table stores data about the resources that are initiated and completed by students during their study sessions. This information should typically be completed from the data provided by the LMS. If the LMS does not provide information about the results of the exercises through the URLs, a solution should be implemented that allows communication with the LMS database to retrieve that information.
Fields:

Name	**Data type**	**Description**
Id	bigint	The field used as a foreign key to the Goal table.
enrollment_id	bigint	The field used as a foreign key for the *Enrollment* table.
user_id	bigint	The field used as a foreign key for the *User* table.
start	timestamp	Stores the date and time when the set goal begins.
finish	timestamp	Stores the date and time the goal ends.
goal_hours	integer	Stores the number of hours planned to be invested in the week.
recorded_hours	integer	Stores the number of hours spent in the planned week.
goal_videos	integer	Stores the number of videos planned to watch during the week.
recorded_videos	integer	Stores the number of videos watched by in the planned week.
goal_evaluations	integer	Stores the number of evaluations planned for the week.
recorded_evalautions	integer	Stores the number of evaluations performed in the week.
week	Integer	Stores the number of the week to which the planned goal corresponds.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: GoalDetail
Description: This table stores data about the detail of the days students planned to accomplish his goals.
Fields:

Name	Data type	Description
Id	bigint	The field used as a foreign key for the *GoalDetail* table.
goal_id	bigint	Field used as a foreign key for *Goal* table
picked	boolean	Stores a value that indicates if the date was selected.
kind	character varying	Stores the type of goal defined by the user.
schedule_date	timestamp	Stores the date and time when planning starts.
execution_date	timestamp	Stores the date and time planned to execute the goals.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: GoalKpiGroup
Description: This table stores the statistical data presented to students during planning to support the strategic planning of new goals.
Fields:

Name	Data type	Description
Id	bigint	The field used as a foreign key for the *GoalKpiGroup* table.
goal_id	bigint	The field used as a foreign key to the *Goal* table.
videos_last_week	integer	Stores the number of videos watched by the student the week before planning.
evals_last_week	integer	Stores the number of assessments conducted by the student the week before planning.
most_effective_day	integer	Stores the most effective day presented to the student during planning.
avg_weekly_videos	numeric	Stores the average video the student has watched per week.

avg_weekly_exams	numeric	Stores the average number of assessments the student has taken each week.
avg_weekly_time	numeric	Stores the average time the student has invested per week.
avg_weekly_time_video	numeric	Stores the average time the student has spent per week watching videos.
avg_weekly_time_exams	numeric	Stores the average time the student has spent each week conducting assessments.
total_videos_this_week	integer	Stores the total number of videos viewed during the week.
total_evals_this_week	Integer	Stores the total number of evaluations carried out during the week.
total_time_videos_week	integer	Stores the total time spent by the student watching videos during the week.
total_tiem_evals_week	integer	Stores the total amount of time students have spent doing assessments during the week.
avg_videos_completers	numeric	Stores the average video observed by students who completed the course in previous editions in one week.
avg_evals_completers	numeric	Stores the average of evaluations made by students who completed the course in previous editions in one week.
avg_tiem_completers	numeric	Stores the average time invested by students who completed the course in previous editions in one week.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date on which the data was updated.

Table: ResourceActivityDatum
Description: This table stores data about the resources used by students in previous editions of a course.
Fields:

Name	**Data type**	**Description**
Id	bigint	The field used as a foreign key for the *ResourceActivityDatum* table.

item_id_lms	character varying	Stores the unique identifier given in the learning LMS for the resource used.
user_id_lms	character varying	Stores unique identifier given to the user in the learning LMS.
state	character varying	Stores the value that indicates whether the resource was completed or started.
course_progress_ts	timestamp	Stores the date and time the resource was accessed.
course_item_type_desc	character varying	Stores the type or category of resource accessed.
session_id_lms	character varying	Stores the unique identifier for the session in which the student used the resource.
course_id_lms	character varying	Stores the unique identifier assigned by the learning LMS to the course.
course_id	bigint	The field used as a foreign key for the *Course* table.
time_spent	timestamp	Stores the time in minutes that the student has spent on the resource.
createa_at	timestamp	Indicates the date the record was added
update_at	timestamp	Indicates the date the data was updated

Table: SessionDatum
Description: This table stores data about the resources used by students in different sessions of previous editions of a course.
Fields:

Name	**Data type**	**Description**
Id	bigint	Field used as a foreign key for the *SessionDatum* table
user_id_lms	character varying	Stores unique identifier given to the user in the learning LMS.
Id_session_lms	character varying	Stores the unique identifier for the session in which the student used the resource.
number_session	integer	Stores the session number for a student. All students start with session 1.
student_data_id	bigint	The field used as a foreign key for the *StudentDatum* table.

id_course_lms	character varying	Stores the unique identifier assigned by the learning LMS to the course.
course_id	bigint	Field used as a foreign key for the *Course* table
date_ts	timestamp	Stores the date and time of the session.
duration	numeric	Stores the time in minutes spent by the student in the session.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date the data was updated.

Table: StudentDatum
Description: This table stores data about the resources used by students in previous editions of a course.
Fields:

Name	**Data type**	**Description**
Id	bigint	The field used as a foreign key for the *StudentDatum* table.
Id_user_lms	character varying	Stores unique identifier given to the user in the learning LMS.
id_course_lms	character varying	Stores the unique identifier assigned by the learning LMS to the course.
country	character varying	Stores the student's country of origin
state	character varying	Stores the value indicating whether the student completed or did not complete the course.
date_completed	timestamp	Stores the date and time the student completed the course.
Date_enrolled	timestamp	Stores the date and time the student enrolled in the course.
createa_at	timestamp	Indicates the date the record was added.
update_at	timestamp	Indicates the date the data was updated.

NMP Frontend

In this section, we present the details of the NMP functionalities offered through the plugin for the Google Chrome browser and the web dashboard. For the dashboard, we give the details of each of the visualizations designed to support the students' self-regulation strategies.

NoteMyProgress Plugin

The plugin supports students' self-regulatory strategies as they perform their learning activities in the LMS. Specifically, the plugin allows students to monitor (Self-monitoring) and become aware of the time invested during their study sessions (Time management). Figure 3.4.3a shows the visualizations of the session time presented to the student. Also, the plugin allows students to take notes on the course material (Organization). Figure 3.4.3b shows the main interface of the notebook.

Figure 3.4.3: Plugins visualizations and notebook interface

NoteMyProgress Dashboard

The NoteMyProgress dashboard provides an interface to support students' self-regulatory strategies outside the LMS. The dashboard offers three main functionalities (see Figure 3.4.4.): **(1)** *Monitor your learning process*, students can self-monitor and become aware of their learning process in the course through interactive visualizations; **(2)** *Manage and take notes*, students can manage notes taken from the NoteMyProgress plugin and create new notes; **(3)** *Goal setting*, students can define their own learning goals for each week of the course.

Figure 3.4.4.: Options menu in the NoteMyProgress dashboard.

Functionality Goal Setting

The goal definition module allows students to define their personal goals for each week of the course. This functionality aims to make students reflect on their degree of commitment to the course and the time available to achieve their individual goals. Figure 3.4.5. Shows the interface for defining students' goals. First, students select the course and the week for which they want to define his goal. Then students select the day or days they plan to conduct their study sessions. Finally, students enter the number of hours they plan to spend, the number of videos to watch, and the number of evaluations to perform.

In order to support the strategic planning of students at the time of defining their goals. NoteMyProgress presents a set of indicators of their performance in the previous week. Specifically, it provides information on their overall performance in the course so far, on the activities to be performed in the week they are planning and on the performance of other students who completed the course in previous editions (see Figure 3.4.6.). The objective of these indicators is to make students aware of their past performance and that of other students who passed the course in previous editions, in order to plan their goals in a more strategic and informed manner.

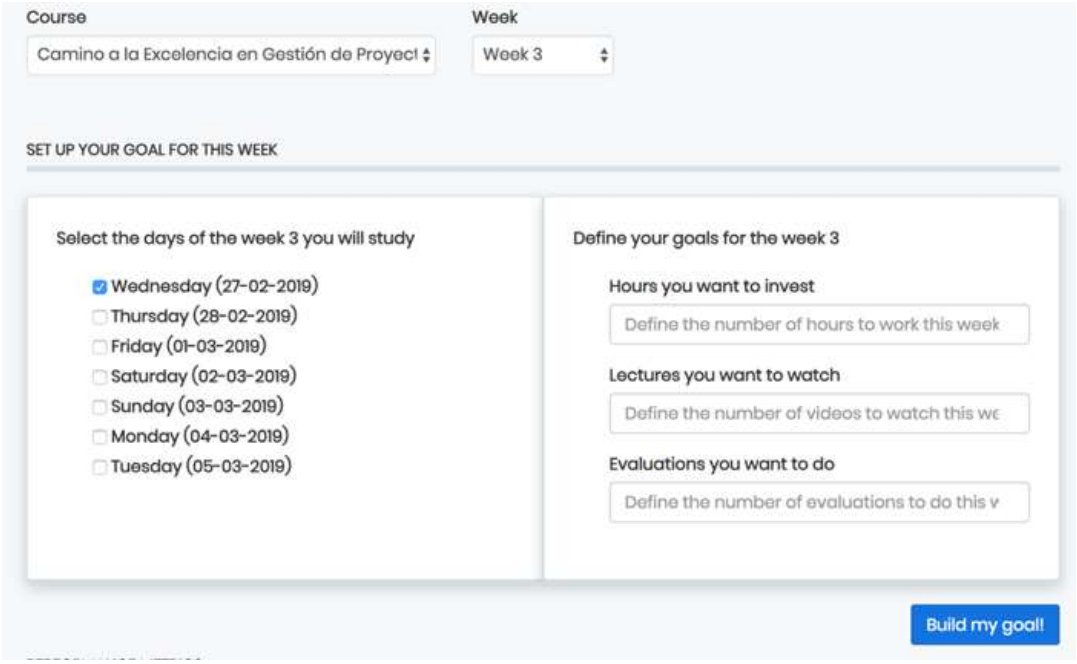


Figure 3.4.5.: Interface for Goal Setting

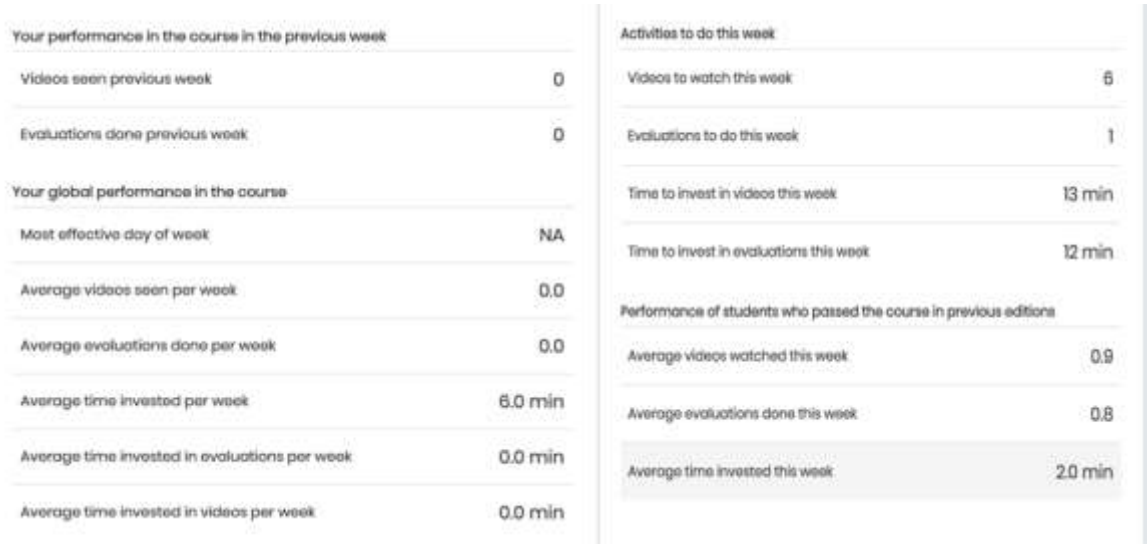


Figure 3.4.6.: Performance metrics to strategic planning

In addition, the Goals Planning module allows students to monitor the level of achievement of the defined goals as shown in Figure 3.4.7. This figure shows the fulfillment of their goals and allows students to compare themselves with other classmates.

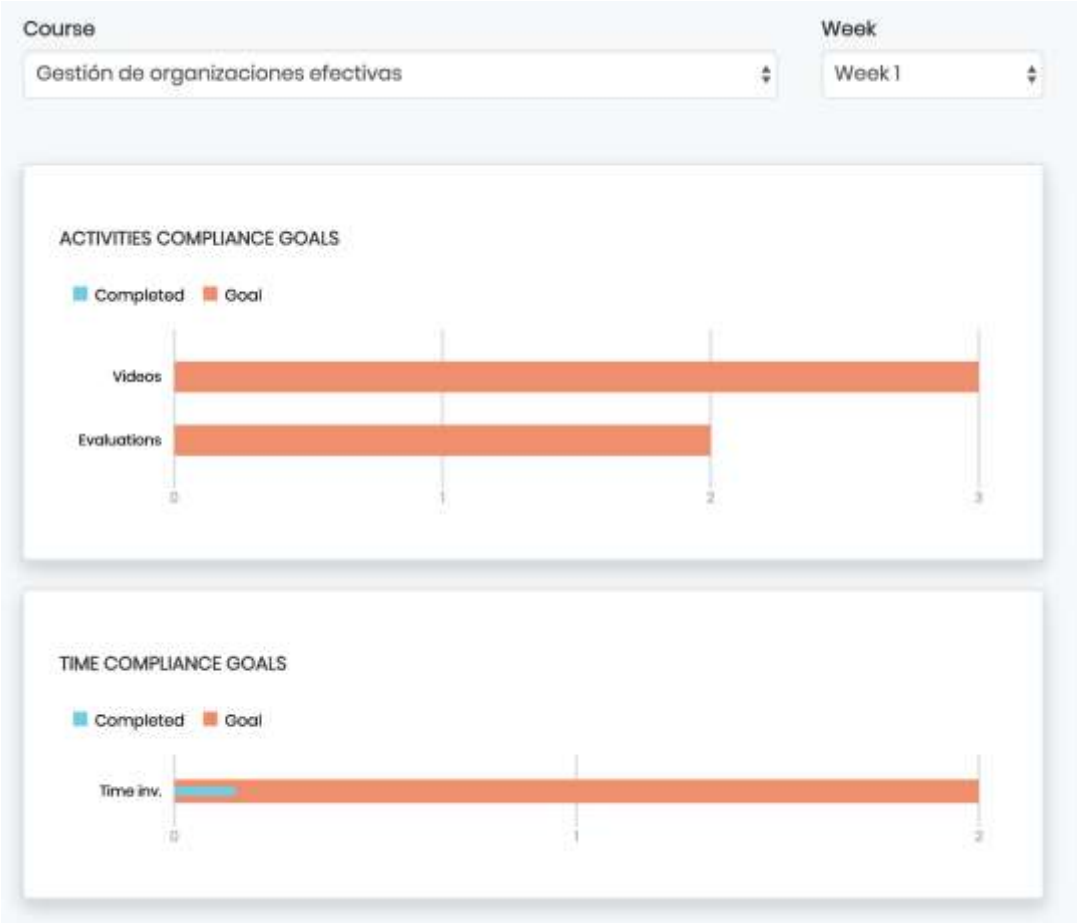


Figure 3.4.7.: Visualization to goal setting monitoring

Functionality Notes Management

The note management module allows students to manage notes taken from the NoteMyProgress plugin notebook (Figure 3.4.8.). In this module, students can create new notes, as well as search, delete and modify notes taken. In addition, students have the option of downloading a note or a set of notes in .pdf format.

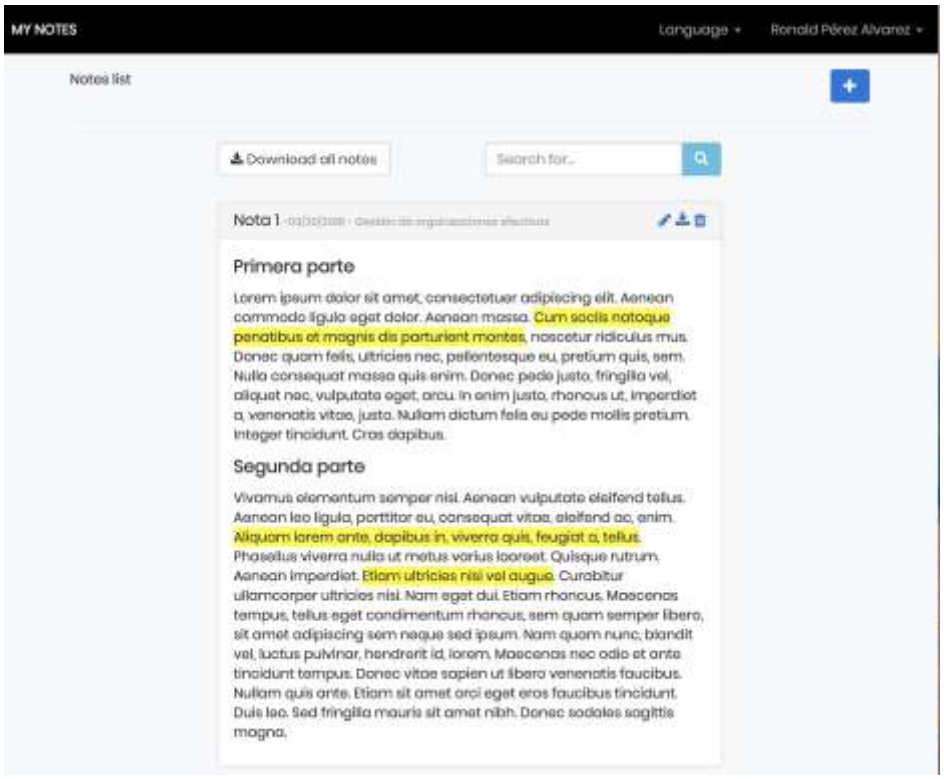


Figure 3.4.8.: Note Interface

Functionality Self-monitoring

The self-monitoring module allows students to monitor and reflect on their learning process in the course. This module is composed of two dashboards that present visualizations of students' performance and effectiveness in performing the learning activities defined in the course (see Figures 3.4.9 and 3.4.10). Visualizations related to students' effectiveness allow them to analyze how effective they were in the course activities. Students can also compare their time spent on activities with the time required for each activity, the number of activities initiated with the number of activities completed, and they can know which day and time of the week was the most effective.

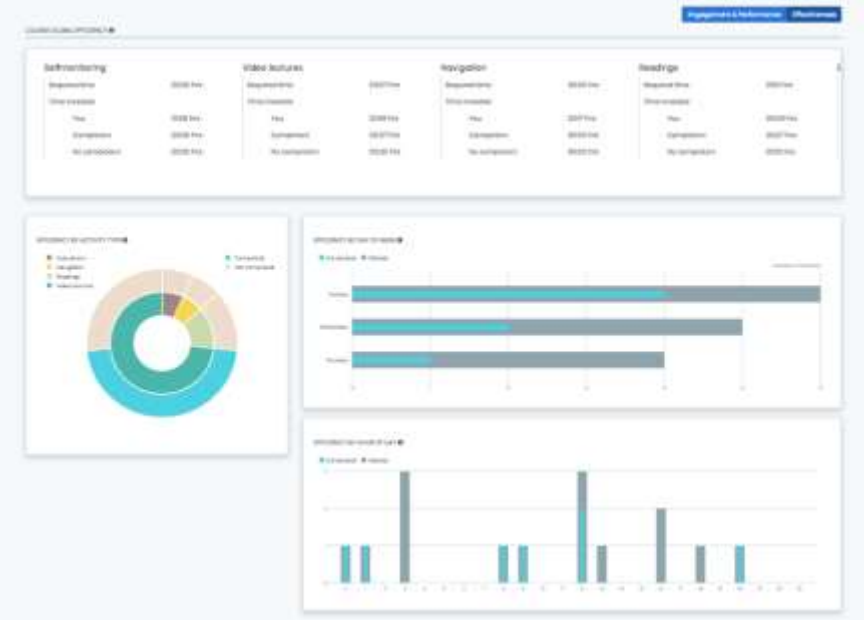


Figure 3.4.9.: Effectiveness dashboard displays

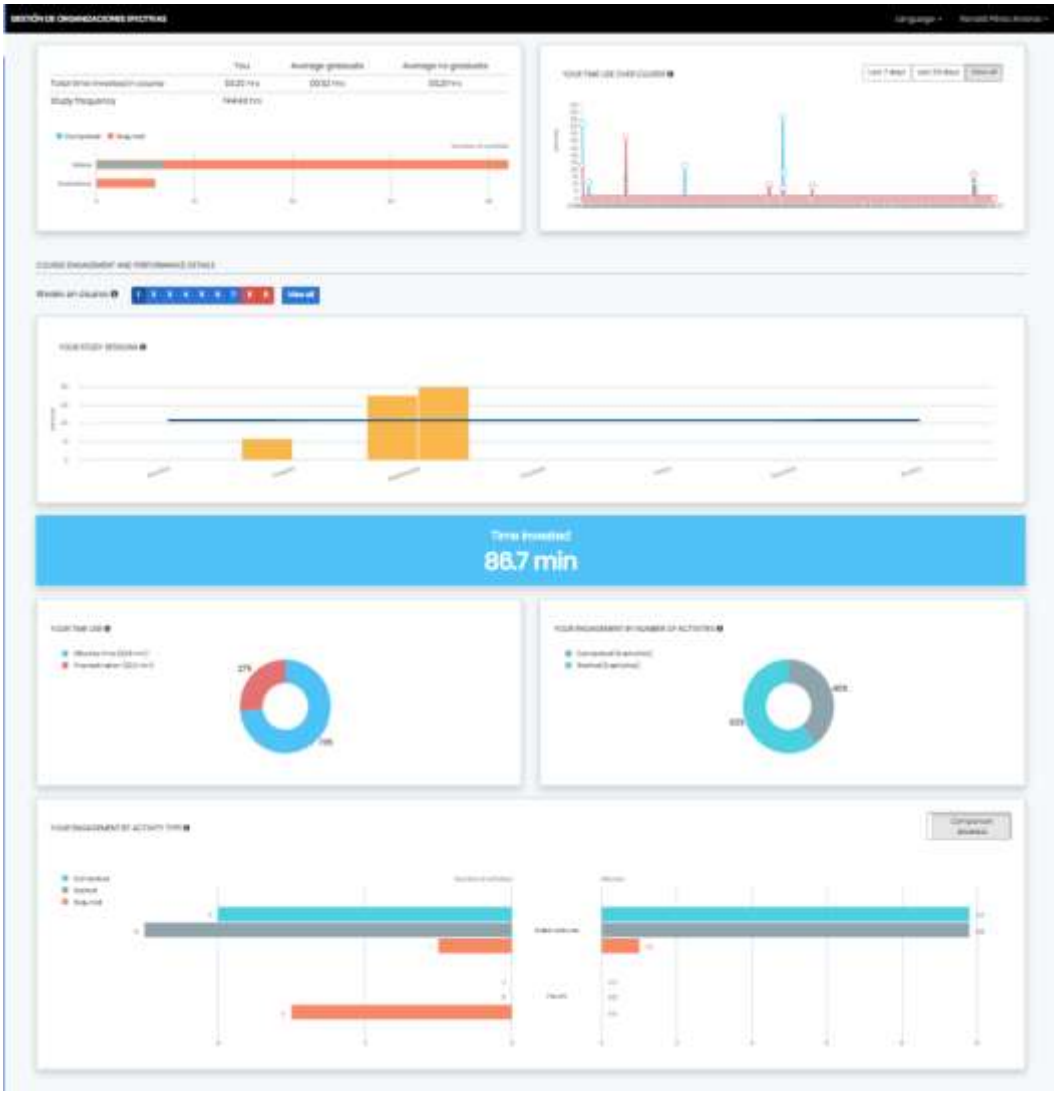
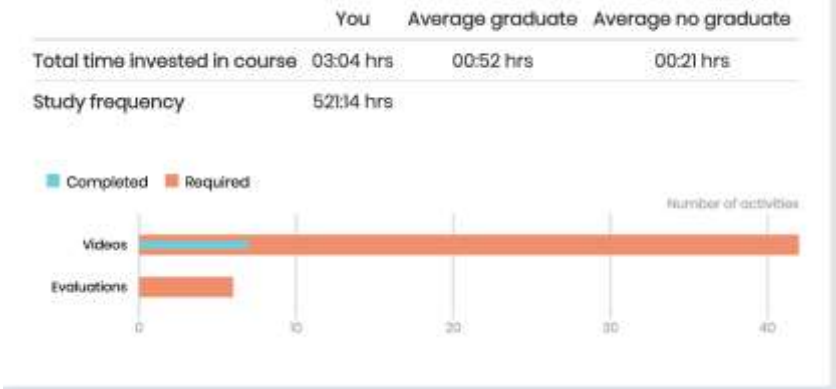
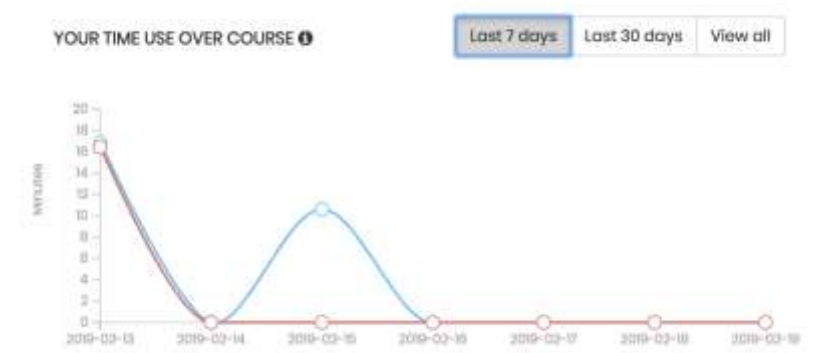
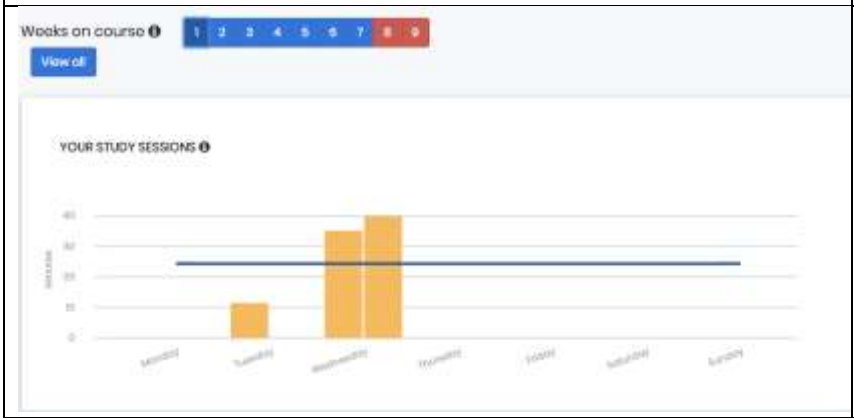
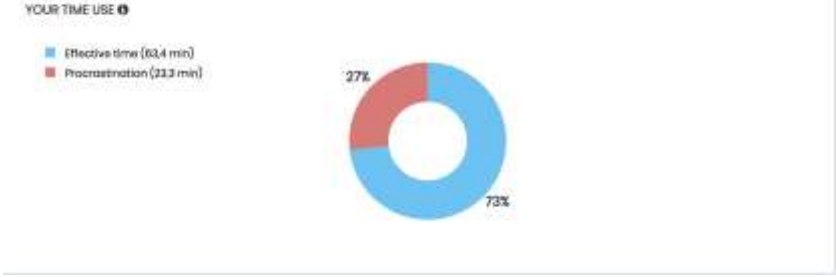
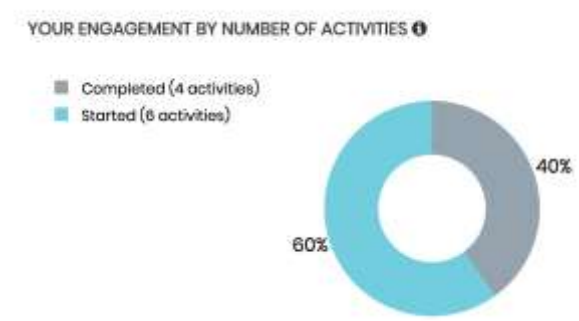
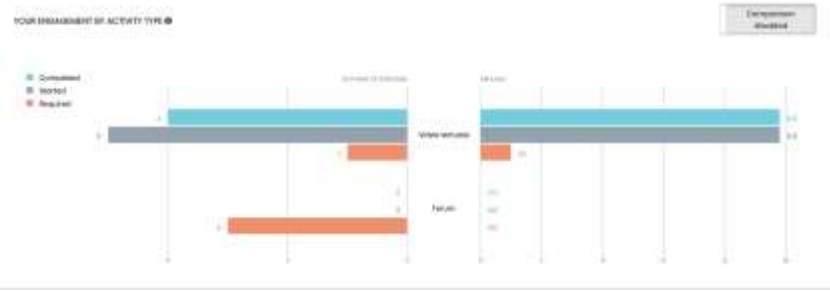
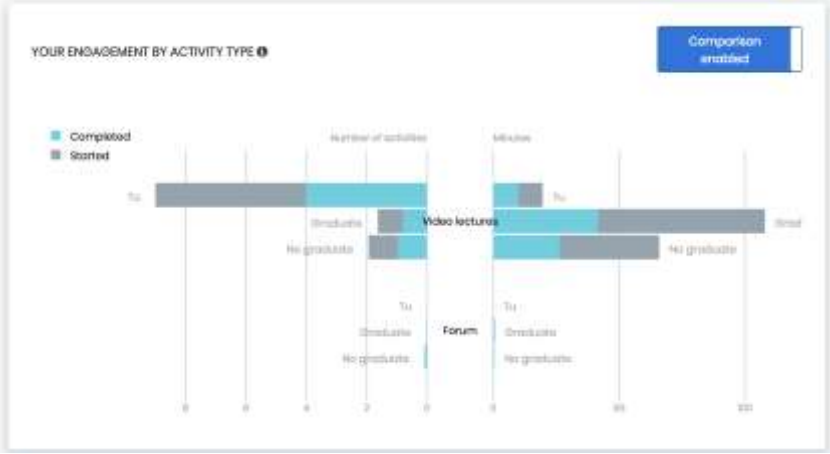


Figure 3.4.10.: Dashboard for performance monitoring

Table 3.4.1. Presents the set of visualizations related to the student's performance. The majority of visualizations are interactive so that students can analyze in detail their information by periods or categories of activities.

Visualization	Description
	This visualization shows the time invested in the students' activities versus time required defined by the teacher, the videos completed versus videos required for that week, the evaluations made versus evaluations required for that week, the frequency of study and the time invested by other students in the course.
	This visualization allows students' to analyze their performance concerning the time invested in activities of the course and activities outside of the course (Procrastination), during a study session. Students can see their performance of the last seven days, 30 days or the whole course.
	This visualization allows students' to analyze the time invested in the study sessions of a specific week of the course.
	This visualization allows students to analyze how they invested their time during the week. It compares the time invested in the course activities versus the time invested in other activities outside of

	the course (Procrastination).
	This visualization allows students to analyze their commitment to the course activities. It compares the number of activities initiated versus the number of activities completed.
	This visualization allows students to analyze their commitment to the course activities (time invested and activities carried out). The visualization shows the time required to invest and the number of activities to be carried out in each category, the time invested on completed activities, and the time invested and number of activities initiated.
	This visualization allows students to compare their performance with the performance of other students who enrolled in the course in previous editions.

Design process and integration with the LALA frame

The design process followed for the development of the NoteMyProgress (NMP) tool (Pérez-Álvarez et al., 2018) is detailed below. To guide the design of NMP, we followed the Design-Based Research methodology (DBR) (Reimann, 2011). This methodology combines empirical research on education with theories oriented to the design of learning environments. This methodological approach was chosen for three main reasons: **(1)** to propose a technological solution driven by educational considerations; **(2)** to understand the impact of these analytical frameworks and solutions on real environments; and **(3)** by its interactive nature, which allows adapting the design to the changing field of research. The phases of this methodology are

analysis, design and implementation and evaluation. In addition, this methodology is easily aligned with the dimensions (Institutional, Technological and Ethical), defined in the LALA Framework.

For the application of the DBR methodology, the Interactive Learning Design (ILD) framework was used (Bannan-Ritland, 2003). This framework organizes the research process in 4 phases: **(1)** ***Informed exploration***, which studies needs, available theories, and target audience; ***(2) Design and implementation***, which consists of the design and implementation of the technology; **(3)** ***Local evaluation***, which aims to analyze the impact of the technology intervention at the local level; and **(4)** ***Wide evaluation***, which considers the adoption of the technology intervention to a wider audience. Below we present the details of the activities carried out in each of the phases and how they align with the LALA framework.

Phase 1: Informed Exploration

Taking into consideration the institutional dimension defined in the LALA framework, the institutional context of the Engineering School of the Pontifical Catholic University of Chile was taken into consideration. In 2015, the School of Engineering started the UC Online project, which is proposed as an alternative to adapt the curricula and current methodologies of the School to the needs of the current market (Pérez-Sanagustín et al., 2018). As part of this project, 13 MOOC courses available on the Coursera platform (https://www.coursera.org/ucchile), and 9 MOOC courses available on the Open edX platform (http://online.ing.uc.cl/) were created. The introduction of the school of engineering to online learning opened up new challenges that were not considered in traditional learning practices. One of the main challenges was to provide a monitoring tool for supporting students' in their self-regulatory process in the online courses offered by the university in an automatic way, without the need of the teacher intervention. That is, the objective of this tool was to offer an automatic monitoring system for supporting students' performance in online settings.

To determine the technical requirements of the tool (Technological Dimension of the LALA framework), we conducted a literature review on the available tools for SRL support in online settings reported between 2008 until 2016 (Pérez-Álvarez et al., 2016).

As a result of phase 1, we identified 5 key requirements for the design of NMP: **(R1)** the design of the tool should be considered as a complement that supports students in different LMS, in order to take advantage of the characteristics currently offered by LMS; **(R2)** to support SRL strategies that in the literature have proven to be effective for students in the context of online learning such as: goal setting, strategic planning, time management, self-evaluation, and note-taking (Milligan and Littlejohn, 2016; Veletsianos et al., 2016; Kizilcec et al., 2017); **(R3)** to provide comprehensive support to learners (within and outside the LMS); **(R4)** to provide different perspectives for information analysis to foster student's active decision making (Pintrich and Boekaerts, 2000). That is, students should have the opportunity to analyze their learning process. Visualizations that provide feedback to learners about their learning process should also be interactive, so that learners, according to their objectives and needs, can monitor aspects that are relevant for them; **(R5)** to visualize objectives or goals to be achieved in the course, as well comparison criteria to evaluate and monitor their performance or progress compared with other learners.

Phase 2: Design and implementation
This phase collects the preliminary requirements obtained from the literature review to generate a first design of a Beta version of the tool, in order to contrast the requirements obtained in phase 1 with the requirements of real users. The design of the first tool is the same in its architecture as the tool presented in the previous sections. In the design part, some details vary, which were modified for the final tool presented here, depending on the results of the local and wide evaluation (Phases 3 and four presented below).

Phase 3: Local evaluation
Two cycles were carried out for the evaluation of the design of the Beta version of NMP. Cycle 1 evaluated the usability and usefulness of NMP, while cycle two evaluated the adoption of this tool in a real controlled context.

Cycle 1: Evaluation of usability and usefulness. Beyond the results of usability and usefulness, this evaluation was conducted to validate the requirements obtained in phase 1 and support the design of a subsequent version of NMP. This study was carried out in the course "Management of Effective Organizations" offered by the Pontificia Universidad Católica de Chile on the Coursera platform. This 7-week course was first launched in October 2015. The case study was conducted only in the first two weeks of the course (week 3-4, March 2017). 4 experts (Women = 1, Men = 3) from 3 countries and 7 students (Women = 3, Men = 4) from 4 countries (Mexico, Ecuador, Costa Rica, Colombia) participated to evaluate the usability of the tool. Based on demographic data, the age ranges of the learners were: 1 under 25 years, three between 25 and 35 years and three between 36 and 45 years. Seven students, 6 of whom have a bachelor's degree or higher and one a secondary education. The experts were selected for their experience in system development, usability, and interface and MOOC design. Students participated voluntarily in the evaluation. The same seven students participated in the utility assessment.

The experts were invited to participate by e-mail and to register in the course as students in the. The experts received a guide with the activities to be carried out, both in the Coursera platform and in the NMP. The students, who enrolled during the evaluation period, were informed of the tool via an e-mail during the first week of the course, explaining the case study and inviting them to participate in the evaluation. The plugin was shared with MOOC participants via a Google drive folder. Participants had to voluntarily download and install the plugin manually, following an installation guide.

The usability of the tool was evaluated using a questionnaire designed by the evaluation heuristics proposed by (Nielsen, 1995). We selected the heuristic evaluation approach because it is an appropriate, efficient and highly effective usability evaluation method in the context of e-learning (Ssemugabi and Villiers, 2007). All questions followed a Likert 5-point scale, where one represents "Totally disagree" and five represents "Totally agree." The average evaluation given by students and experts was calculated for each of the Nielsen principles. To measure the students' perception of the tool's usefulness, we designed a different questionnaire for each of the functions included in the tool. This instrument was specifically designed to obtain qualitative information about the main functionalities of the tool. The questionnaire consists of 15 questions. 8 questions related to the different functionalities of the tool. For example, "The information shown in the visualizations is relevant to me." These questions follow a Likert scale of 5 points, where one is "Totally disagree," and 5 is "Totally

agree." In addition, the questionnaire contained 2 open-ended questions on the suggestion of new functionalities and general comments, 4 demographic questions and 1 question aimed at knowing your consent for the use of a future version of the tool. Table 3.4.1 shows the main results collected from this first evaluation (Pérez-Álvarez et al., 2017).

Table 3.4.1. Result of usability and usefulness evaluation of NMP

Cycle	Results
Usability and usefulness evaluation	- It is a usable tool - It is a useful tool for students - The installation process should be simple - The need to improve the tool interface - The need to improve visualizations

Cycle 2: Adoption evaluation. In this cycle, the adoption of NMP was evaluated in order to understand the interaction with different functionalities. The study was carried out in three MOOCs: (1) Gestión de organizaciones efectivas, which has a duration of 7 weeks; (2) Hacia una práctica constructivista en el aula, which has a duration of 10 weeks; and (3) Electrones en Acción, which has a duration of 4 weeks. All courses are offered by the Pontificia Universidad Catholica of Chile on the Coursera platform. This case study lasted 2.5 months (April, May and June 2017). A total of 126 students (Men = 70%, Women = 30%) from 10 countries participated in the case study conducted in this cycle. Demographic information was obtained from the data report downloaded from the Coursera platform, which provides little demographic data on students.

All students who enrolled during this time were sent an email in the first week of the course, explaining the case study and inviting them to participate in the evaluation. The plugin was uploaded to the Google web shop, and the students received the link to the tool plugin, which is installed directly from the shop by pressing the install button. Table 4.3.2 shows the main results obtained in this cycle.

Table 3.4.2 Result of evaluation of PWN adoption

Cycle	Results
Adoption evaluation	● Many students interacted only once, due to the lack of clarity and relevance of the information displayed. ● There was an increase in the adoption concerning the number of students using the tool in Cycle 1. ● The average frequency of student entry was 4.5 days. ● The greatest interaction occurred with the visualizations that show information about the time spent. ● Students engaged in activities related to Pintrich strategies such as self-control, time management, and organization.

Phase 2: Broad evaluation

As a result of the first two phases, a summary of the lessons learned was proposed to create the design of the first version of NMP. This version takes into account the requirements obtained from the literature and the requirements obtained through the evaluations with the students (end users). The final design of the tool incorporates a simple and agile installation

process to consider the diversity of students enrolled in the courses. In addition, a set of indicators were defined (see table 4.3.3) to organize the information presented to students and to create visualizations to support specific SRL strategies.

Tabla 3.4.3. *Indicators and strategies supported by each indicator. TM = Time Management, O = Organization, SP = Strategic Planning, GS = Goal Setting, SE = Self-evaluation.*

Type	Indicator	Description	Strategy supported
Engagement	time spent (course, session, category, procrastination)	time spent summarize	TM
	number of sessions	number of sessions achieved.	TM
	time required	total time estimated by the teacher to each category.	TM
	weeks on the course	number of weeks spent.	TM
	study frequency	average time among sessions.	
	notes taken	number of notes taken in the course.	O
Performance	activities completed	number of different activities completed.	TM, SE
	activities started	number of different activities started.	TM, SE
	activities attempted	number of different activities attempted.	TM, SE
	activities required	number of activities proposed by the teacher.	TM, SE
	videos planned to watch	number of videos planned by the learner to watch during the week.	SP, GS
	time planned to spend	time planned by the learner to spend during the week.	SP, GS
	evaluations to be taken	number of evaluations planned by the learner to do during the week.	SP, GS
Effectiveness	most effective day	day of the week and time of the week in which most activities are completed.	TM, SP

The final design incorporates robust and interactive visualizations. The interactive component in the visualizations allows students to perform a deeper analysis of their behavior and focus on the most relevant points according to their personal goals and needs. We propose a new set of interactive visualizations (see section 4.3.3 Frontend) organized according to the classification defined for the different indicators. In addition to interactivity, the design of the visualizations allows students to analyze information from different perspectives and periods: session, activity category, day, month or see a general scheme of the learning process until the current week. Different goals and standards for comparison were also added, which support the monitoring process of the students. First, we incorporated the objectives defined in the course (number of activities required, and time required to perform them). Secondly, we added a functionality for students to set their own objectives (number of videos to watch, number of evaluations to perform, time to spend, the day scheduled to study) and we designed a visualization to compare the planned objectives with the objectives achieved. Finally, a functionality was integrated to provide students with a comparison of their performance with the rest of the students in the course with data from previous courses. Social comparison has been shown to have a positive effect on student engagement and efficiency (Brusilovsky et al., 2015).

4. Adaptation of Early Warning dropout System

This section describes the process by which algorithms and tool visualizations were developed to predict early dropout in students. Early abandonment prediction tools were made adapting early prediction tools of universidad Carlos III de Madrid Madrid (Moreno-Marcos, Muñoz-Merino, Alario-Hoyos, Delgado Kloos, 2018)

4.1 Prediction for MOOCs in PUC

One of the contexts where predictions were carried out was in specific courses. In this case, the objective was predicting whether students were going to drop out of the course or not. This case has many differences with respect to academic prediction because of the context and the variables that can be gathered for the predictive models. On the one hand, it is not the same completing an individual course (with limited duration) than completing a whole degree, whose duration is longer. On the other hand, academic prediction frequently uses academic data as indicators (e.g., students' grades), while prediction in courses usually takes students' interactions with course contents and resources.

Particularly, the data that were used for prediction was taken from MOOCs (Massive Open Online Courses) hosted in Coursera. Coursera is an online platform, and Pontificia Universidad Católica de Chile hosts courses there. Nevertheless, the technical design of the predictive models is completely transferrable to other digital platforms regardless of courses are face-to-face or online. For this analysis, three courses have been considered: Electrons in Actions, Constructivist Classroom and Management of Effective Organizations. All these courses were taught in Spanish in Coursera, and they had an asynchronous delivery mode. This means that all materials were released at the beginning and learners could enroll and engage with the contents at any time. Therefore, there were no deadlines for completing any part of the course.

From course data, the steps that were considered for the development of the predictive models (which can be transferrable to other scenarios) are as follows.

1. Data cleaning (filtering of students)
2. Obtaining higher-level variables from raw data (indicators)
3. Obtaining the variable to predict (dropout in this case)
4. Development and assessment of the predictive models

What follows is a description of the steps that were taken for each of the abovementioned steps. First, it is essential to clean data to remove useless data for the analysis. For the case of MOOCs, it is very frequent that the number of registered learners is very high (there can be thousands of learners). However, many of them do not watch a single video, and they do not solve a single exercise. Because of that, it is crucial to filter those learners to reduce bias in the predictive models.

In this case, self-reported data were available related to learners' intentions and self-regulated learning (SRL) strategies. These data were obtained through an initial questionnaire. As not all the learners completed the questionnaire and variables obtained from it were going to be

part of the predictive models, only learners who filled the questionnaire were included in the predictive models.

Once the filter of students was carried out, it is necessary to define which variables are going to be used in the models. Raw data from Coursera include the learners' events in the platform. That includes the events of beginning a session, play a video, complete a video, attempt an assessment, complete an assessment, etc. However, these events are useless unless they are processed.

Because of that, a processing phase is needed to obtain higher-level variables from Coursera events and the initial questionnaire. In this context, seven categories of variables have been defined: (1) self-reported SRL strategies, (2) event-based SRL sequence patterns, (3) demographic variables, (4) variables related to learners' intentions, (5) variables related to activity, (6) variables related to videos, and (7) variables related to exercises. What follows is a description of each category:

- Self-reported SRL variables: They provide information about aspects such as whether students can seek help when they face difficulties, establish objective, plan their learning, self-evaluate, etc. These data are obtained from the initial questionnaire.
- Event-based SRL sequence patterns: They indicate patterns that are found in learners' interactions in the platform. These patterns include accessing the platform only to watch videos or solve exercises, accessing the platform to watch the videos first and then completing the exercises, accessing the platform to read the assessment questions first and then look for the answers in the videos, etc. These data are obtained from Coursera events.
- Demographic variables: They include data such as the learner background, age, gender, and binary variables about whether the learner is a student (in formal education) or not and whether the learner is working or not. These data are collected from the initial questionnaire.
- Variables related to learners' intentions: They include information about the number of hours the learner plans to engage with the MOOC, and binary variables about whether the learner is interested or not in the MOOC topic, whether the learner has previous experience with MOOCs, and whether the learner plans to take the summative assignments or not. These data are collected from the initial questionnaire.
- Variables related to activity: They include indicators such as the number of active days in the platform, the total time the student has interacted with the platform and the number of different sessions in which the learner has worked in the MOOC. These data are collected from Coursera events.
- Variables related to videos: They collect information about how learners interact with videos. They include the number of times a learner starts watching a video, completes a video, and reviews a video. Moreover, they include the percentage of opened, completed, and reviewed videos. These data are collected from Coursera events.
- Variables related to exercises: They collect information about how learners interact with exercise. They include the number of times a learner starts an assessment, completes an assessment, and reviews an assessment. Moreover, they include the percentage of attempted, completed and reviewed assessments. These data are collected from Coursera events.

Once the indicators are calculated for each learner, it is necessary to retrieve information about the dependent variable, dropout, in this case, to train the model. The definition of dropout can be context-dependent because unlike the final grade which is a fixed value; there can be several interpretations of dropout. For example, it may be considered that a learner has dropped out when he/she does not show interactions for a week. However, in another context, that inactivity period could be one month instead of a week, etc. Furthermore, a learner could be considered as a dropout in another context if he/she only watches the video lectures but does not complete the assessments. In this case, a learner is considered as a dropout if the learner does not have any interaction for four consecutive weeks. Nevertheless, learners who have completed at least 80% of the assessments have been removed to avoid false positives. The reason is that it can be assumed that if these learners do not access the course is because they have finished the course. Moreover, learners whose interactions only comprises the last days of the data collection period have been removed as there is less information about them. As the MOOCs considered were asynchronous, and learners could access to the contents at any time, data collection was restricted until December 2015. Therefore, learners who were doing the course by that period were removed as it is not possible to classify them as dropout or not.

The last step in the design process is the development of predictive models. In order to do that, R open-source software was used as a tool. Particularly, library caret of R was used. This open-source library implements several machine learning algorithms that can be used to develop predictive models. With this library, four algorithms were used to train the models: (1) Random Forest (RF), (2) Generalized Linear Model (GLM), (3) Support Vector Machines (SVM), and (4) Decision Trees (DT).

These algorithms were used to train the predictive models from labeled data, i.e., data which included both the indicators and whether the learners have dropped out or not. These models have the indicators of the seven abovementioned categories as an input, and their output is the probability a learner has of dropping out the course (between 0 and 1). This way, high values of that probability means the learner is at risk of dropout and corrective measures should be taken to address this issue.

4.2. Academic prediction in undergraduate programs

The second part of the dropout prediction was made in academic courses. In this prediction, we will try to calculate the probability of abandonment in the complete career and in the first years of the degree (bachelor), in different degrees or meshes of 3 universities such as: University of Cuenca (Cuenca, Ecuador), Escuela Politécnica del Litoral (Guayaquil, Ecuador) and the Universidad Austral de Chile (Valdivia, Chile). Unlike the prediction in the MOOCs, to make this prediction more global data of the degree and above all, academic data will be used.

The data taken for the prediction of the dropout in the degrees will be taken from the students' academic results, socio-economic data of the students and the data obtained from the different degrees or meshes. The predictions are for each university, but the variables used are valid for the calculation of the probability of abandonment in any other university.

As explained in the part of the prediction for MOOCs, the steps to follow for the development of predictive models are the following:

1. Data cleaning (student filtering)
2. Obtaining higher level variables (learning indicators)
3. Obtaining the variable to predict (in this case, dropout)
4. Development and evaluation of predictive models

These steps will be detailed later, since, for each university, different data and criteria were used to carry them out.

4.2.1 ESPOL

In the case of ESPOL, the data collected belong to 3 different categories: demographic data of the student (marital status, employment status, place of birth, socio-economic factor ...), data on the different careers (number of total credits, semesters of each career, identifier ...) and data of the student's performance (state of the subject, number of times the subject has been taken, subject grades, credits of the subject ...).

As we mentioned above, the data of those students who are not valid should be deleted. That is student data that is not correct or incomplete data. In the case of ESPOL, and although data was collected many years ago, these data were complete and were valid to continue with the second step.

For the academic prediction, the most representative data of the evolution of the students in the degree was chosen. These data were processed to achieve the highest level indicators. These indicators could be classified in the same categories mentioned above:

- Demographic indicators: These indicators show essential data of the personal life of each student, such as working life, marital status or sex. In this case, it was not possible to make use of the socio-economic status of each student, since these data could not be collected from a large number of students.

- Indicators of the student: These indicators show data of the student's status in the career as the approved percentage of credits, a weighting of the current average of the student in the career, the percentage of approved concerning the total of present.

- Indicators of each degree: With the data of the students and the data of the different degrees, a variable has been obtained that indicates the percentage of dropouts of each degree.

When the indicators were obtained, the next step was to determine the dropout. To define the output variable (dropout) and establish if a student had left the degree, a maximum of years was used without the student enrolling. In the case of ESPOL, and after observing that a large

number of students returned to their higher education even after abandoning them after several years, it was decided to dropout the case in which a student has not enrolled in any subject for more than five years.

Finally, the last step was the development of predictive models. In the case of ESPOL, the Python programming language and the free software of the Scikit-learn library was used. This open source library implements many machine learning algorithms with which the different predictive models were made. Different tests were made with the following algorithms: Random Forest, Decision Tree, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) and Gradient Boosting.

Through the use of these algorithms, different tests were carried out for the training of the models from the data in which previously it had been defined whether they had done dropout or finished the degree. This way, the predictive models were obtained, which have as input the aforementioned indicators and as output, the probability of finishing the career of each student. High probability values mean that the student has a high probability of graduating, while low values indicate a high probability of dropping out.

4.2.2 UACh

The next context for developing the predictive models is in Universidad Austral de Chile (UACH). In this case, only academic data are available. This means that there is only information about the students' results in the subjects he/she enrolled, despite he/she may have some subjects validated, annulated, etc. In this case, data was collected from 2011 onwards. This collection period supposes one of the first limitations for the data filtering. As there are students who started their degrees before 2011, it is necessary to discard them as their grades are not available from the beginning of their degrees. Moreover, students who have started their degree recently are also discarded as there is not enough information to determine whether they have finished their degree or not.

From the abovementioned academic data, five indicators have been obtained for the predictive models:

- Average grade: Indicates the average grade of the enrolled courses

- Ratio of coursed courses: Indicates the relationship between the courses taken (coursed) and the enrolled courses. A course may not be coursed if it is validated, recognized, annulated or transferred from a previous study plan.

- Ratio of annulated courses: Indicates the relationship between annulated courses and enrolled courses.

- Ratio of repetitions: Indicates the relationship between the number of courses a student is taking and the total number of calls of his/her courses. For example, if a student is taking five courses, but he/she is taking one of them for the second time (he/she is in the second call of the course), the ratio will be 5/6.

- Ratio of passed courses: Indicates the relationship between the passed courses and enrolled courses.

Previous variables can be obtained globally or by semester so that if a learner has taken three semesters, there would be 15 variables (5 variables for each semester). This can be useful to collect information about the evolution of the learner through the degree.

Once variables are obtained, the next step is determining the variable to predict. In this case, unlike the case of ESPOL and U. Cuenca, two variables are considered. The first one is dropout, and a student is considered a dropout when he/she does not enroll in any course for two consecutive semesters, which is the equivalent of an academic year. In Chile, unlike Ecuador, it is not frequent that students resume the degree after stopping it for a while. Therefore, the period for considering drop out is smaller than in Ecuador contexts.

Another common issue in UACH is that students take too much time to finish the first two years of the degrees (what is usually known as "Bachillerato"). It is possible to find students in the last stages of the degree with some Bachillerato courses pending to pass. Because of that, the second variable to predict is the number of years the students will take to finish the Bachillerato.

Finally, once the predictor variables and variables to predict had been defined, predictive models were developed. In this case, the R open source library called caret was used. With this library, the following algorithms were used: (1) Random Forest (RF), (2) Generalized Linear Model (GLM), (3) Support Vector Machines (SVM), (4) Decision Trees (DT), and (5) neural networks (NN). The output of these algorithms was the probability of dropout (between 0 and 1) for the case of dropout, and a continuous value representing the number of semesters a student was going to take to finish the Bachillerato.

4.2.3. UCuenca

In the case of UCuenca, the data collected belongs to 3 different categories: demographic data of the student (type of school attended by the student, socio-economic factor ...), data on the different degrees or meshes (number of total credits, semesters of each degree, identifier ...) and data on the student's performance (student's status, state of the subject, enrolment number, qualifications of the subject, credits of the subject ...).
The data provided by UCuenca had to be cleaned since the data of some students were incomplete. At the time of collecting the data, some students had already started the degree, and since the data from the previous years were not collected, many of the data were not complete.
Once the data provided was cleaned, the highest level indicators were calculated with the most representative data of the students and the meshes. These indicators could be classified into the following categories:

- Demographic indicators: In the case of UCuenca, only the socio-economic factor was used since no personal data of the students was obtained.

- Indicators of the student: These indicators show data of the student's status in the career as the approved percentage of credits, a weighting of the current average of the student in the career, the percentage of passing students concerning the total of students registered.

- Indicators of each degree: With the data of the students and the data of the different degrees, a variable has been obtained that indicates the percentage of dropouts of each degree in addition to the ratio between the number of semesters enrolled and the semesters of each mesh.

Once the indicators were obtained, the next step was to determine the output variable (dropout). To define the dropout and establish if a student had abandoned the career, a maximum of years was used without the student enrolling. In the case of UCuenca, it was decided to dropout the case in which a student has been enrolled for more than three years in any subject.
Finally, the last step was the development of predictive models. In the case of UCuenca, the Python programming language and the free software of the Scikit-learn library was used. This open source library implements many machine learning algorithms with which we proceeded to perform the different predictive models. Different tests were made with the following algorithms: Random Forest, Decision Tree, Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) and Gradient Boosting.

Through the use of these algorithms, different tests were carried out for the training of the models from the data in which they had previously defined whether they had done dropout or had graduated. In this way, we obtained the predictive models that have as input the indicators above and as output, the probability of finishing the career of each student. High probability values mean that the student has a high probability of graduating, while low values indicate a high probability of dropping out.

4.3. Dashboards

In the case of ESPOL, as in UACH and U. Cuenca, the early warning dropout system was considered as part of a visualization of the counseling system and not as a separate system. As a consequence, both tools were worked together, having the same methodology explained in section 3 Frontend: Adaptation of the Counseling Tool.

4.3.1 ESPOL

The visualization is part of the Statistics window and shows a thermometer indicating how near or far the student is approaching a possible dropout as well as the percentage. The indicators that are taken into consideration were explained in section 4.2.1. In addition, for the intervention and actions related to early prediction not only this graphic is used, but it is combined with those already seen in the counseling tool (Figure 4.3.1).

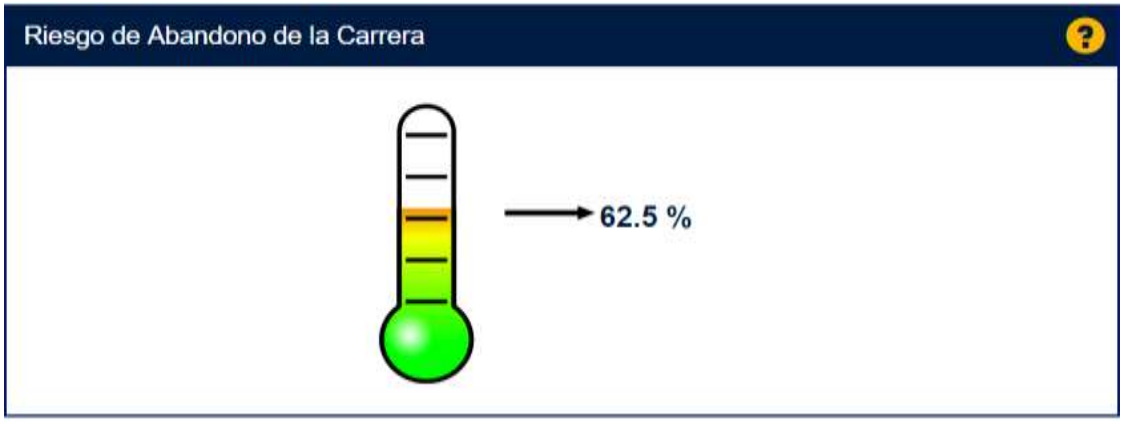


Figure 4.3.1 Ventana de riesgo de abandono de carrera

4.3.2 UACH

Módulo de avance y predicción de semestres para terminar

Para diseñar la herramienta de predicción, se analizaron las entrevistas realizadas al comienzo del proyecto LALA con los líderes institucionales y se elaboró una propuesta simple pero que ayuda a la problemática más frecuentemente mencionada en la Facultad de Ingeniería, la alta deserción de los estudiantes durante el bachillerato de ingeniería y el largo tiempo que toma finalizar los cursos de bachillerato.

En la parte superior derecha de la Figura 4.3.2 se observa las predicciones realizadas.

Figura 4.3.2 Módulo de avance y predicción de semestres para terminar

- Este módulo sólo será visible para los directores de programa que dirijan carreras que incluyan plan de bachillerato (se debe definir si está activo o no por cada uno, en la configuración)
- Muestra el porcentaje de avance con asignaturas aprobadas / asignaturas totales de los dos primero años que corresponde con el ciclo de bachillerato, para los programa de ingeniería.
- Toda la información vendrá en el servicio GetStudentAcademics (descrito en el Backend de la herramienta de Counselling).. Los nuevos atributos son:
    - "bachCompletedCourses" : number,
    - "bachTotalCourses" : number,
    - "estimatedTermsToCompleteBach" : number,
    - "estimatedProbability" : number

4.3.3 UCuenca

The system of early prediction of desertion was considered as part of a visualization of the counseling system and not as a separate system. As a result, both tools were worked together, having the same methodology explained in the counseling section. The visualization forms part of the student's statistical information window, and displays a visualization ("Risk of Leaving the Career") indicating how near or far the student is approaching a possible desertion. The indicators that are taken into consideration, were explained in the previous section. In addition, for the intervention and actions related to early prediction, not only this graphic is used (Figure 4.3.3) but it is combined with those already seen in the counseling tool.

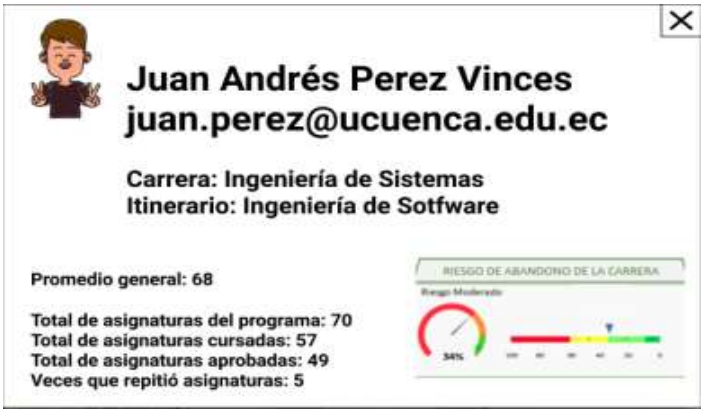


Figure 4.3.3 PopUp of Statistical informaton of the student

5. Adaptation and adoption of other tools

In addition to the counseling tools and "dropouts" or calculation of student desertion, LALA has adapted learning analytical tools that were already part of the ecosystem of some of the higher education institutions involved in the project. This is the case of OnTask, and VERA These are learning analytical tools that do not require many adaptations and that, thanks to LALA, were improved and adapted to increase their use and adoption by the different target groups. These tools, like the aforementioned tools, will be adapted to facilitate their use to institutions external to the project. This section describes the technical characteristics of these tools, emphasizing the aspects that were improved and adapted for the project.

5.1. VERA: Visualizador de Encuestas para la Reflexión Académica (Survey Viewer for Academic Reflection) (UACh)

Additionally, the implementation of LALA tools in UACh also contemplated a visualization and explanation tool for survey results, aimed at students and in order to encourage reflection on the results of these surveys. This tool, baptized as VERA (Visualizador de Encuestas para la Reflexión Académica) part of the experience in KULeuven where they developed the LASSI application (Figure 5.1.1) that shows the results of a survey made up of 60 statements (Learning and Study Strategies Inventory) and applied to first-year students. The similarities lie in the context of UACh, where surveys are also applied to the first-year students self - concept survey and learning strategies) but whose results are not known by the students surveyed

Figure 5.1.1 LASSI

VERA Database

To develop VERA, we used the following tables:

Table: survey

Description: survey entity

Fields:

Name	Data type	Description
id	text	Survey ID
label	text	label or name
text	text	description of the survey

Table: factor

Description: It is represents a factor of the survey (for example, factor ' emotional self-concept ' of the self - concept survey

Fields:

Name	Data type	Description
id	text	factor identifier
label	text	label or name
survey_id	text	reference to the id of the survey
text	text	description of the factor

Table: level

Description: entity level represents a factor level (e.g. low level)

Fields:

Name	Data type	Description
id	text	level identifier
label	text	label or name
factor_id	text	reference to the id of the factor
advice	text	description of the factor

Table: student_response

Description: represents the result of a student's answers, that is, what level each student has in each factor of each survey

Fields:

Name	Data type	Description
student_id	text	Student ID

survey_id	text	Survey ID
factor_id	text	factor identifier
level_id	text	level identifier
score	bigint	numerical value corresponding to the level (level) that the student has in the factor

Table: student
Description: student entity
Fields:

Name	Data type	Description
id	text	level identifier
yam	text	label or name
program_id	text	id of the program in which the student is

Table: cohort_count_by_level
Description: stores the number of students in each cohort of each program (career), for each level of each factor and each survey
Fields:

Name	Data type	Description
cohort_year	integer	year of the cohort
program_id	text	identifier of the program or career of the cohort
factor_id	text	factor identifier
level_id	text	level identifier
count	bigint	number of students

Table: program_academics_by_levels
Description: stores summary and aggregated information on academic performance of all previous cohorts, for each level of each survey factor
Fields:

Name	Data type	Description
factor_id	text	factor identifier
program_id	text	identifier of the program or career of the cohort
level_id	text	level identifier
percentageLow	number	percentage of students with low academic

		performance (2 or more subjects failed in the first year)
percentageMid	number	percentage of students with average academic performance (1 subject failed in the first year)
percentageHigh	number	percentage of students with high academic performance (no failing subjects in the first year)

Design

To design the application they opted for a series of iterations that began with the construction and design of non-functional prototypes taking as a starting point the LASSI application, employees to satisfy two major objectives. The first one consisted of obtaining the requirements to develop the tool. The second, on the other hand, was related to the definition of the design of the tool.

Although the baseline application was built, it was decided to start with the construction of prototypes to concentrate on the design, rather than on details of the implementation, considering also that the interaction is simple and the surveys are different.

To check progress, discuss design decisions and plan the following tasks, weekly meetings were held with the sponsoring professor and in some cases with entities of the Undergraduate Student Learning Support Unit (UAAEP), with the idea of taking advantage of and benefiting from their experience in working with students.

Finally, studies were conducted to assess the perception of end users

First Iteration

Initially, LASSI was used as a model since this tool fits the purposes of this work, which correspond to transmit a clear, brief and accurate message, as well as show the information in steps following a pattern and offering minimalist visualizations. In this context, when dealing with an adaptation, the greatest challenge lies within the framework of visual design and not in the development of software.

Given that the source code of the application was not available and the fact that the same surveys will not be used, work was done on the development of Mockups to represent and simulate interactions, to validate and evaluate the perceptibility and usability of the tool. First, a scheme consisting of two tabs was designed, one for the presentation and the other associated with the Learning Strategies survey. The first tab is shown in Figure 5.1.2 F and the second in Figure 5.1.3.

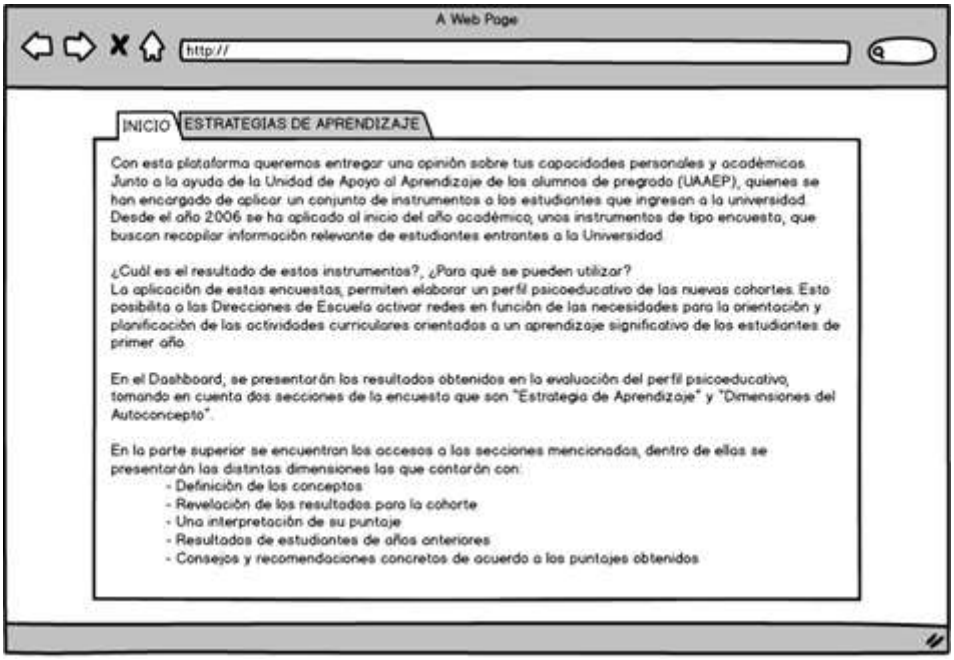


Figure 5.1.2 First Mockup: "Home tab "

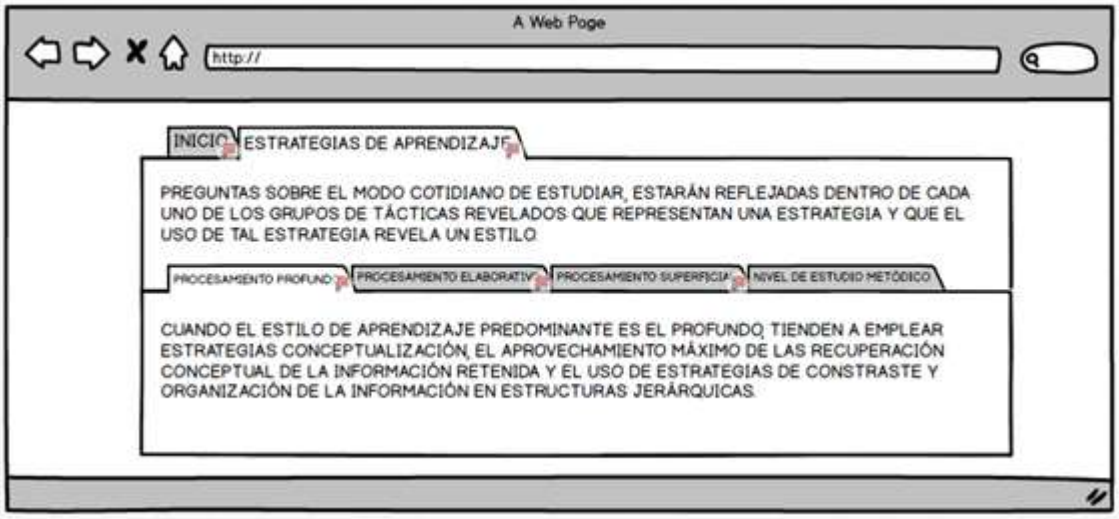


Figure 5.1.3 First Mockup: "Learning Strategy Tab"

In Figure 5.1.4, a brief description of what learning strategies are is shown. Below is the option to access each of its dimensions, revealing a brief preview associated with the first alternative.

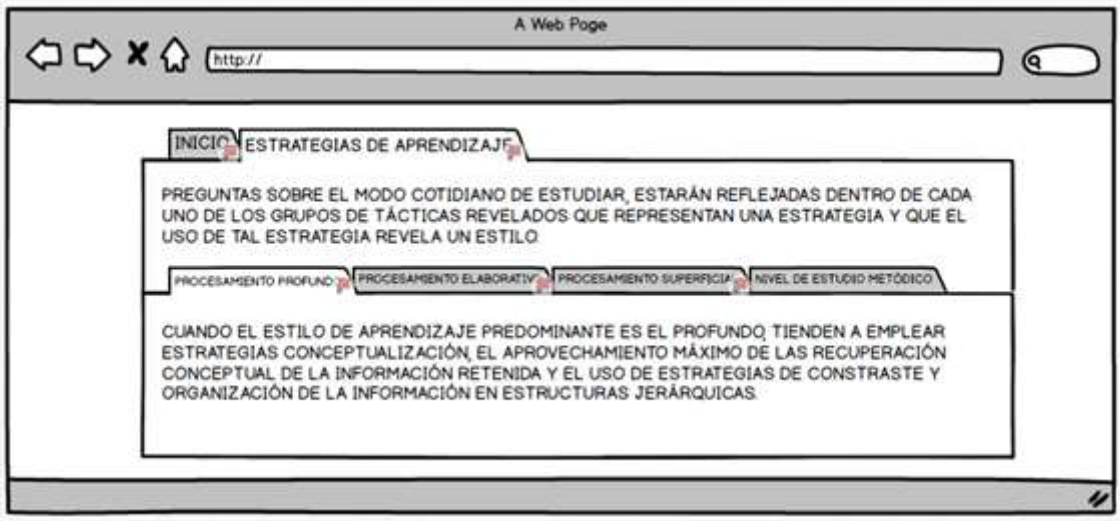


Figure 5.1.4 First Mockup: "Deep processing"

Once you enter one of the dimensions, first the general definition referring to Learning Strategies is hidden. Then, a series of elements that model a scheme to show the information to the user are shown: First, a definition associated to the dimension and a visualization in which the distribution of the results of the promotion to which the user belongs is revealed. . Then, a link appears that fulfills the task of showing and hiding information, which will be shown later in Figure 5.1.5. Finally, at the bottom you can see a button that allows access to a new tab with recommendations to improve this ability. In the following figure you can see the tab with recommendations that emerges after pressing the button.

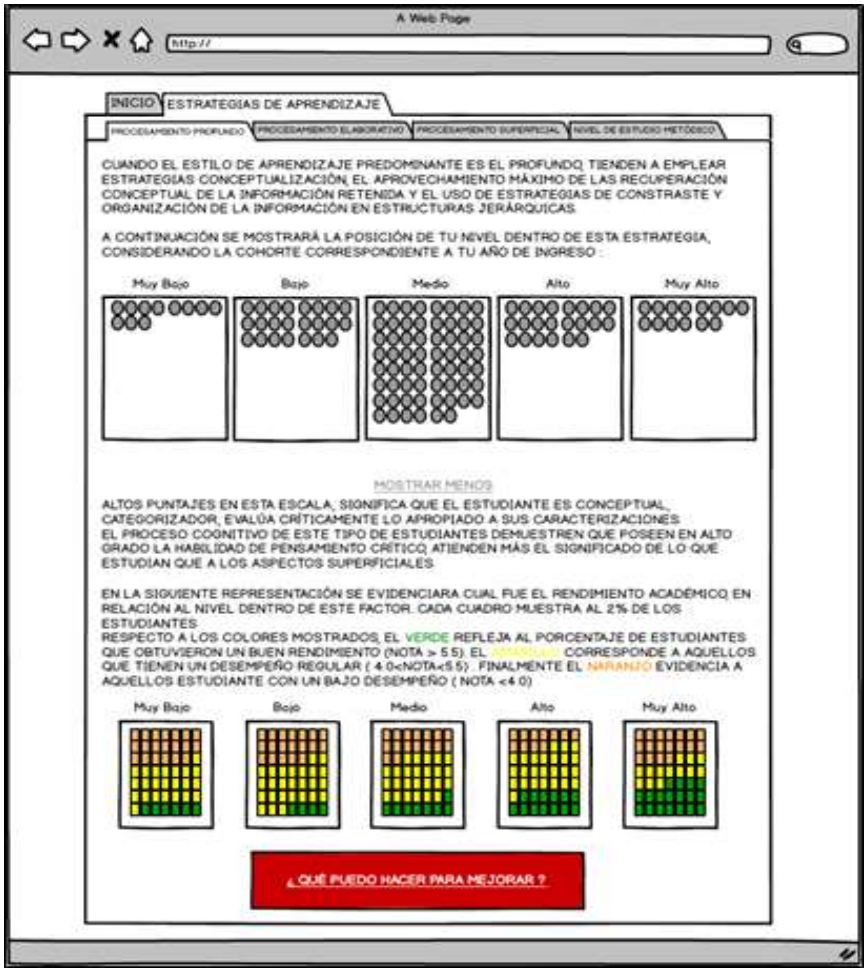


Figure 5.1.5 First Mockup: "tab with all the content"

After clicking on the "a little more" link, a second visualization will appear showing the results of the survey of students from previous years combined with their academic performance obtained in the first semester, specifically in percentage terms, a distribution of students from according to their average obtained (Figure 5.1.6).

Figure 5.1.6 Tab of recommendations in the learning strategies

The recommendations tab showed a series of tips that were the same regardless of the dimension in which you are browsing.

Once presented, a series of meetings were held to discuss the design. The first ones were carried out with the LALA team, who proposed using activatable items and deactivating them to give dynamism. In addition, he suggested adding smaller statements and indicating the level or result obtained using some distinctive element.

At this stage, there was little clarity as to which sections of the psychoeducational profile report would be used. It was until the first meeting with entities of the UAAEP, where it was confirmed that the results of the Self - concept and Learning Strategies sections would be used. In the same instance, they approved the combination of the results of the survey and academic performance, since, according to their statements, there is a correlation. Finally, it was determined that the data collected from 2014 to date would be used.

Within the framework of the Learning Strategies section, it was proposed to separate it into two sections: The first referred exclusively to the Methodical Study Level and the second level about the Information Processing Levels (Profound, Elaborative and Superficial). With this, I would change the design of this section. For its part, the Self - concept section will be made up of five dimensions and will, therefore, have five different tabs to show the information.

Second iteration

Considering the proposals and decisions described in the previous section, a new iteration was carried out. In Figure 5.1.7 the results obtained by adding new components are shown.

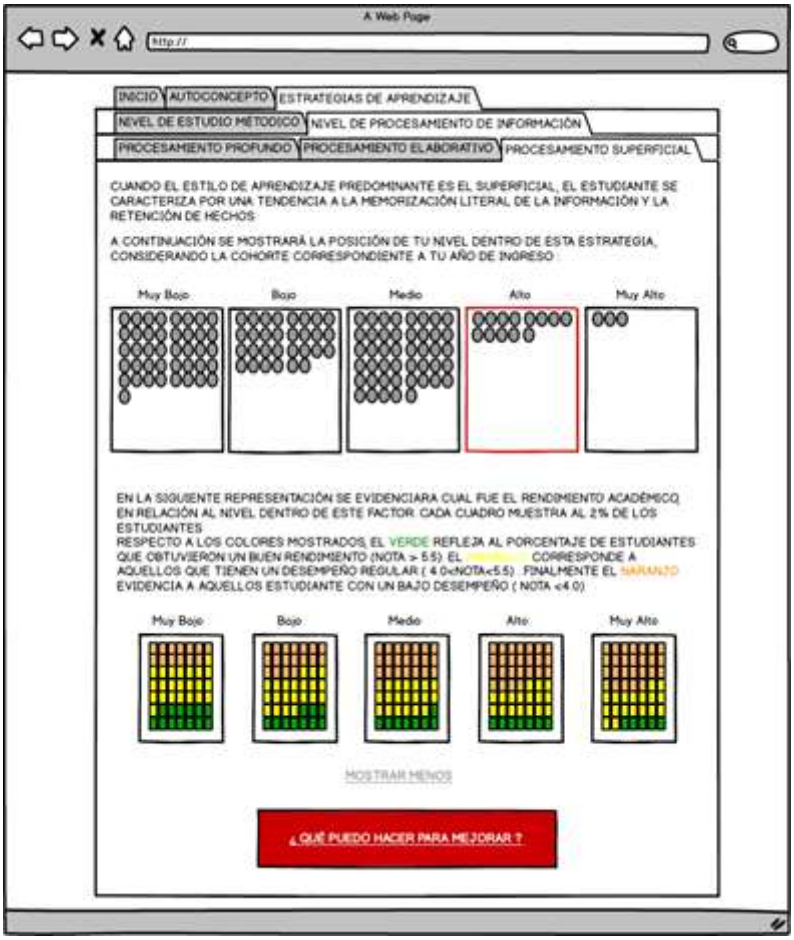


Figure 5.1.7 Second Mockup: "Surface Processing Level"

Figure 5.1.7 shows that the learning strategies section was separated into two levels. In addition, it is visible that the Self - concept section was added. Also, a red color was integrated at the edge of the container to indicate the level obtained when responding to the survey. Finally, the link was reordered to make disappear the elements that show the results of students of previous promotions (Figure 5.1.8)

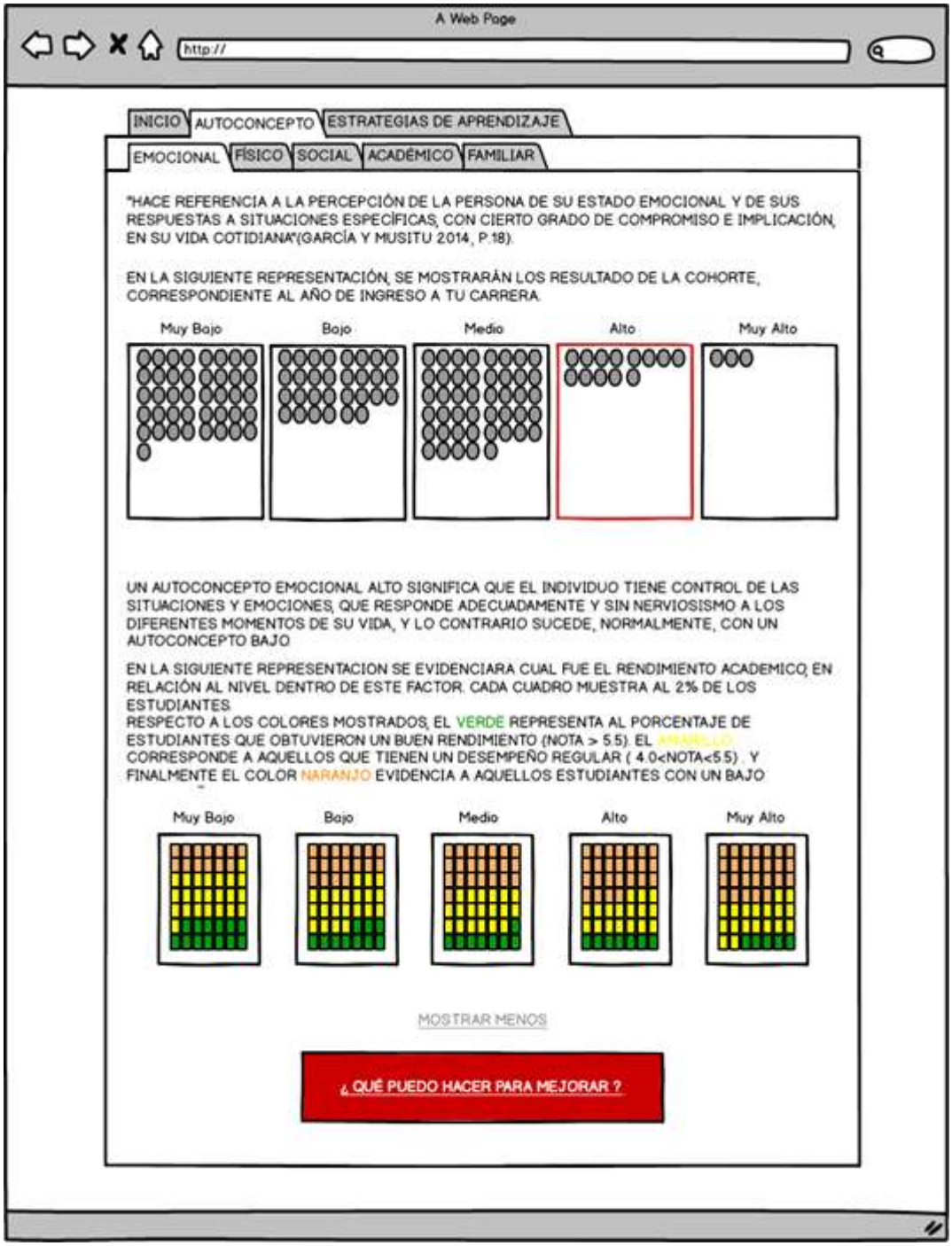


Figure 5.1.8 Second Mockup: "Emotional Self-concept"

Thus, once these mockups were finished, a new revision stage was carried out through meetings with the LALA team and UAAEP entities

The most significant changes arose in the meeting with entities of the UAAEP. One of the topics dealt with was how to show the results of the Autoconcept section, because it is **sensitive information**; it should not be delivered abruptly. This has to be treated responsibly. Students at 17 and 18 are beginning to form, they see certain features of their personality and the results of self - concept in the sense that they are very low, can engage in risk behaviors. An example of this is the interpretation that a student gives to his Self-concept Academic when this is Very Low, where I could point out that: «I do not have the necessary skills to study, I better not keep trying» or «I have to make an effort to improve.» Another example of this, if a student has a Very Low Social Self-concept, can increase that and become much more part of their result, feeling more excluded from the group. There is a variety of interpretations that depend clearly on the person, that is why the following proposal was made; instead of showing five different levels ranging from "Very Low" to "Very High", it was decided to group the results of the students in three different levels (Improve, Enrich and Persist), with the intention of giving a more subtle message, of a positive and actionable character.

Third iteration

The last iteration carried out, involved the change of the grouping of the results in the different dimensions of the Self - concept section, thus, it went from a total of five levels to only three. In Figure 5.1. 9, you can see the change made.

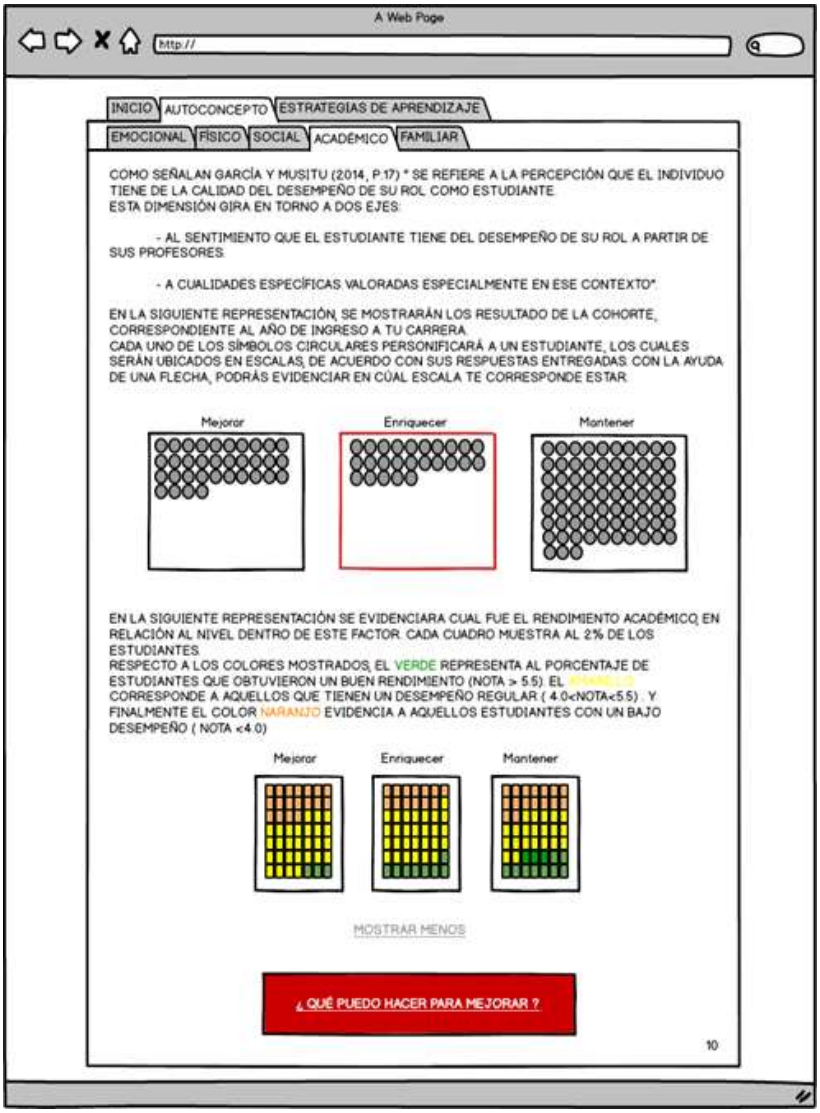


Figure 5.1.9 Third Mockup: "Academic Self-concept"

Study one, non-functional prototype

Method

To carry out the study session the following procedure was performed; First of all he had to set a time and place where the study will be made. In the second place, the first year students of the Civil Engineering in Computer Science career were contacted. For this, a teacher was asked to attend a class to invite university students to be part of this important stage of work. Finally, a protocol was defined to guide the session and set the times and set an order for the tasks. To begin the session, a brief introduction was made in which the tool and the objectives of the session were discussed. After a few minutes, each of the students was He gave a printed consent for them to sign it. Then, we proceeded to ask a series of questions with the idea of finding out if they remembered something from the survey they responded to when they enrolled. Subsequently, 13 sheets were distributed to each young person, which contained screenshots of the built mockup and additionally, instructions directed to the students so that they can carry out the simulation. In Figure 5.1.10, an example of the instructions inside the mockup is shown.

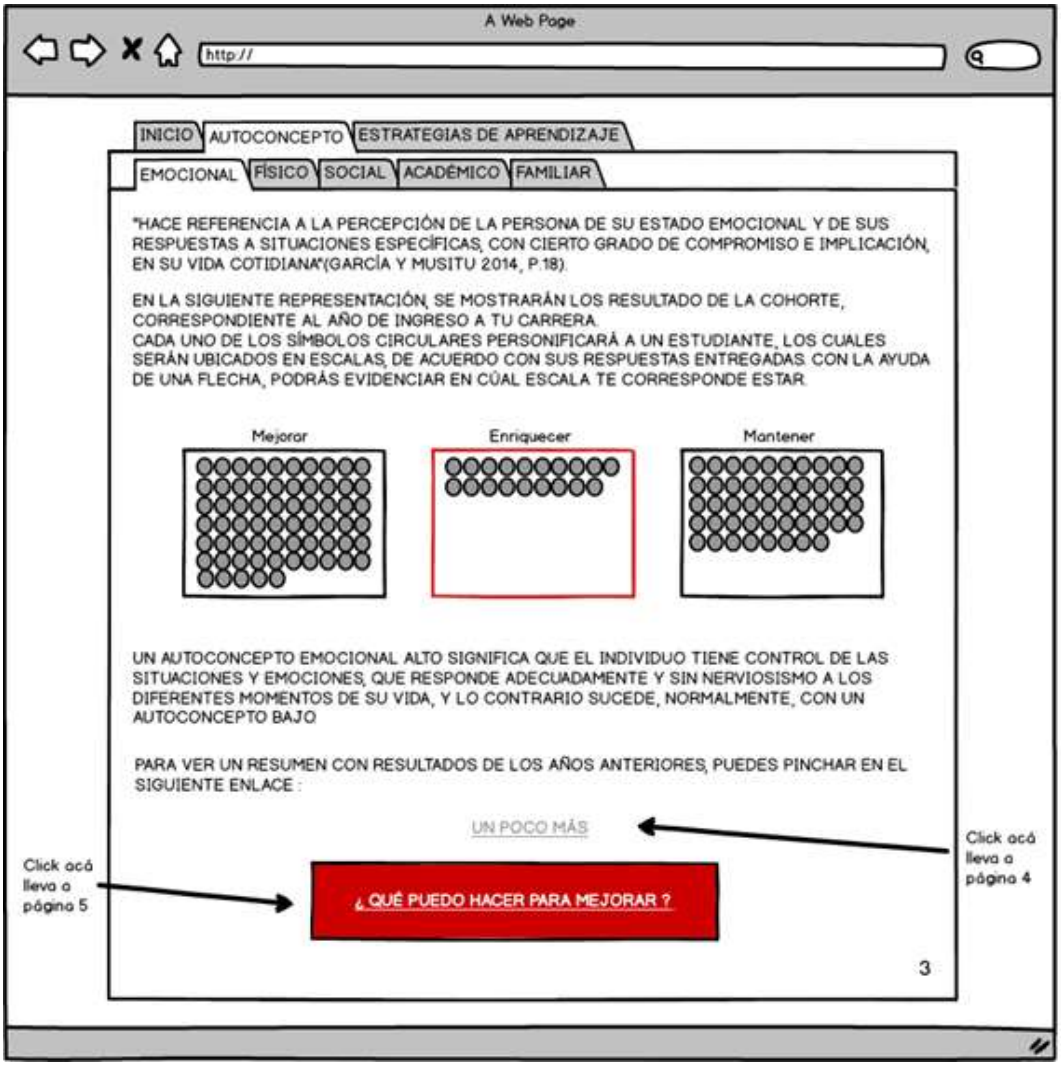


Figure 5.1.10 Part of the mockup used in the study session

In addition, they were given a questionnaire prepared with the Likert technique that they had to answer after interacting with the mockup. The Likert scale is a tool used to evaluate the opinions and attitudes of a person, specifying the level of agreement or disagreement with a statement (item or question). In the following figure it is presented the questionnaire used.

Afirmación	muy en desacuerdo	algo en desacuerdo	ni de acuerdo, ni en desacuerdo	algo de acuerdo	muy de acuerdo
Puedo ver claramente cuál fué mi respuesta en Nivel de Estudio Metódico					
Puedo reconocer fácilmente en qué nivel de Estudio Metódico está la mayoría de los estudiantes					
Es fácil entender la explicación en texto de nivel de Estudio Metódico					
Es fácil entender qué significan los cuadros con celdas grises en nivel de Estudio Metódico					
Es fácil entender qué significan los cuadros con celdas de colores en nivel de Estudio Metódico					
Puedo reconocer fácilmente qué nivel de Estudio Metódico se relaciona con peor rendimiento académico					
Puedo señalar qué Nivel de Estudio Metódico es "mejor" o más positivo					
Puedo ver claramente cuál fué mi respuesta en Autoconcepto Emocional					
Puedo reconocer fácilmente en qué nivel de Autoconcepto Emocional está la mayoría de los estudiantes					
Es fácil entender la explicación en texto de nivel de Autoconcepto Emocional					
Es fácil entender qué significan los cuadros con celdas grises en Autoconcepto Emocional					
Es fácil entender qué significan los cuadros con celdas de colores en Autoconcepto Emocional					
Puedo reconocer fácilmente qué nivel de A. Emocional se relaciona con peor rendimiento académico					
Puedo señalar qué nivel de Autoconcepto Emocional es "mejor" o más positivo					
Me gustaría usar la herramienta terminada para ver mi propia información					
En general creo que es fácil usar esta herramienta					
Creo que esta herramienta va a ser útil para los estudiantes					
En general los textos me parecieron fáciles de entender					
Creo que podría ver como estoy en todos los resultados de la encuesta en la herramienta terminada					

Questionnaire session one non-functional prototype

Then, as a preliminary instruction, they were asked to consider that the mockups show their information, with the idea of simulating that the system is being used with the actual results obtained in the survey.

Once they finished interacting with the mockups and answering the questionnaire delivered, a group conversation was held to have all the students mention the positive and negative aspects of the tool, in order to make future changes or simply keep the decisions of designs used.

Results

In this study session, 7 freshman students of the Civil Engineering degree in Computer Science attended. The approximate duration time was 50 minutes.

The most relevant comments issued by the students and other observations are listed below.
Initially, the instructions that were added in the mockups are understood.
Most remember that they were surveyed many times. Some expressed negativity about the fact that the survey was mandatory in order to access the system for the first time and that this caused some to answer it without the corresponding seriousness.
Generally, it is difficult to understand the statements used to describe and explain each component. They encourage to show texts as if they were intended for children (easy to understand).
They understand that the box with the red border indicates the level obtained, but suggest adding an explicit indicator.
They ask about the way to access the tool, for the reason that they are not shown a window to log in.
When asked about the purpose of the tool, they mention that it is an enhancer, because if in any dimension you have a low level they offer help to improve.

It was commented that the tool is well focused, for the reason that it shows you that in some aspect you are wrong and it helps you. In addition, they see very positively receiving feedback quickly after responding to surveys, since many times the results are not returned.
There was much uncertainty regarding the graphs that show the academic performance of the students of previous years. The representation is not understood, and they are confused when interpreting. In general, the majority did not understand the message.
They suggest changing the color used on the button to show the tips, for the reason that the red color represents danger or aggressiveness.

In the following table, you can see the average answers for each of the statements.

Average results questionnaire session one non-functional prototype

Affirmation	Average answer
1	Something of agreement
2	Very of agreement
3	Very of agreement
4	Very of agreement
5	Neither agree nor disagree
6	Something of agreement
7	Something of agreement
8	Very of agreement
9	Very of agreement
10	Very of agreement
11	Very of agreement
12	Neither agree nor disagree
13	Neither agree nor disagree

14	Something of agreement
15	Very of agreement
16	Something of agreement
17	Something of agreement
18	Something of agreement

Regarding the comprehension, clarity, and cohesion of the representations and statements, an average answer was obtained "neither in agreement nor in disagreement," this mainly because the statements were not adequate to explain each concept and also for the reason that there was much confusion when interpreting the second graph. On the other hand, in relation to the usability and future success of the tool, the average response was "Something agree." In general, the students gave their approval to the tool and believe that it will be a great support during their adaptation to university life.

Solution architecture

It consists of a *Front end [1]* and a *Back end [2]* that supports it. The application would make requests to an API developed in the *Back end* that extracts the data from an architecture that uses a *data warehouse [3]* fed by ETL operations [4] that act on databases (Figure 5.1.11 and Figure 5.1.12).

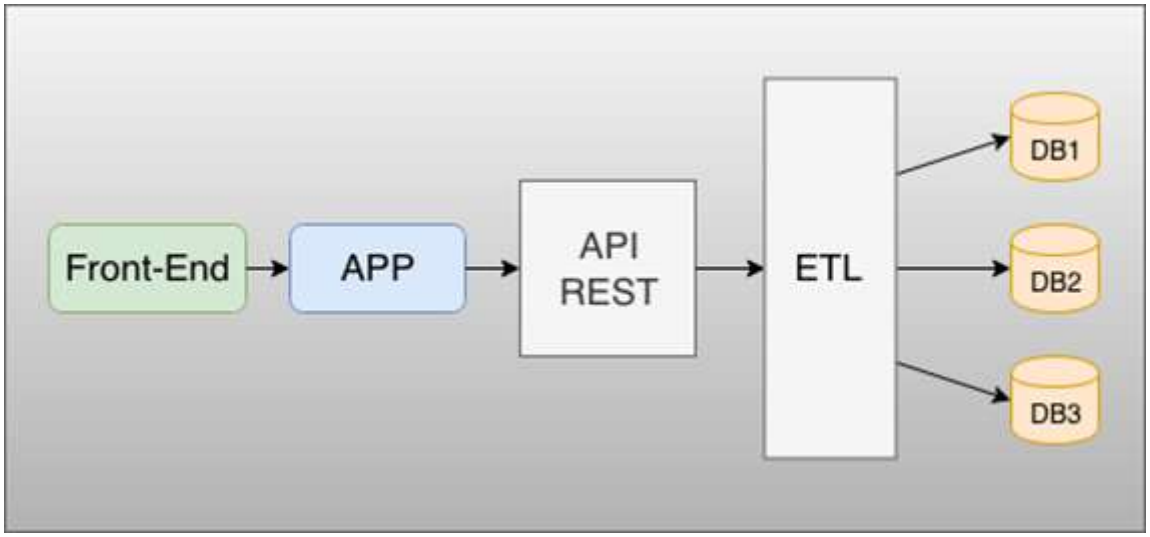


Figure 5.1.11 VERA Architecture

.

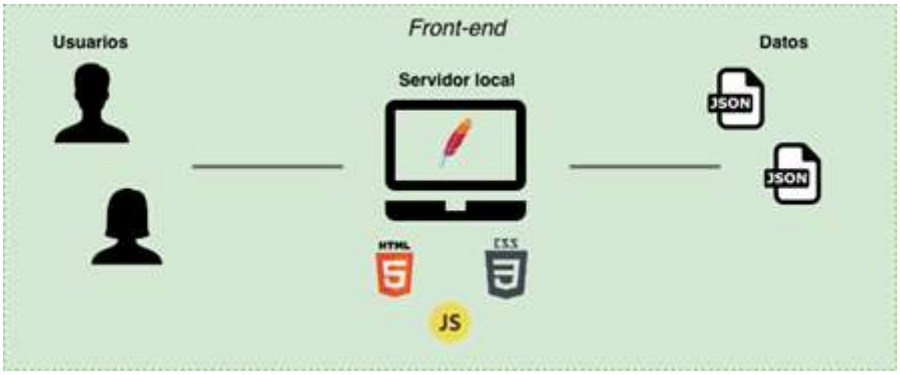


Figure 5.1.12 Frontend architecture

API data in JSON format
surveyText: has the information shown in the home tab and each of the definitions associated with the different dimensions of the survey and references

```
{
        "home" : String,
        "introduction" : String,
        "surveys" : [ Survey ],
        "references" : [ Reference ]
}

"Survey": {
        "id" : String,
        "label" : String,
        "introduction" : String,
        "factors" : [ Factor ]
}

"Factor" : {
        "id" : String,
        "label" : String,
        "text1" : String,
        "textGraphic1" : String,
        "textGraphic2" : String,
        "levels" : [ Level ]
}

"Level" : {
        "id" : String,
        "label" : String
}

"Reference" : {
        "reference" : String
}
```

surveyResult: Presents the results obtained by the students in the survey and the information associated with academic performance.

```
        {
          "studentID"       : String,
          "startYear"       : int,
          "studentResponses" : [ STUDENT_DATA ],
          "groupData"       : [ GROUP_DATA ]
}

        "STUDENT_DATA": {
                "surveyId"    : String,
                "factorId"    : String,
                "componentId"  : String,
                "levelId"     : String
        }

        "GROUP_DATA" : {
          "surveyId"    : String,
          "factorId"    : String,
          "componentId"  : String,
          "levelId"     : String,
          "cohortCount"  : int,
          "historic"    : [ double ]
        }
```

Recommendation:

```
{
                "surveys" : [Survey]
        }

        "Survey" : {
                "id" : String
                "factors" : [Factor]
        }

        "Factor" : {
                "id" : String,
                "high" : String,
                "low" : String
}
```

---

[1] Presentation layer
[2] Data access layer
[3] Integrated data collection, non-volatile and variable over time
[4] Extract, transform and load is the process that allows to move, reformat, clean and load data
[5] Are the URLs of an API or a backend that respond to a request

Final version

The identifier corresponds to an anonymization, and it is the subsequent task of the backend to resolve the access (Figure 5.1.13).

Figure 5.1.13 Identifier

The home tab is used to make known the origin of the tool, collaborators and data source. In addition, it briefly explains the different components of the tool. It consists exclusively of texts and images (Figure 5.1.14).

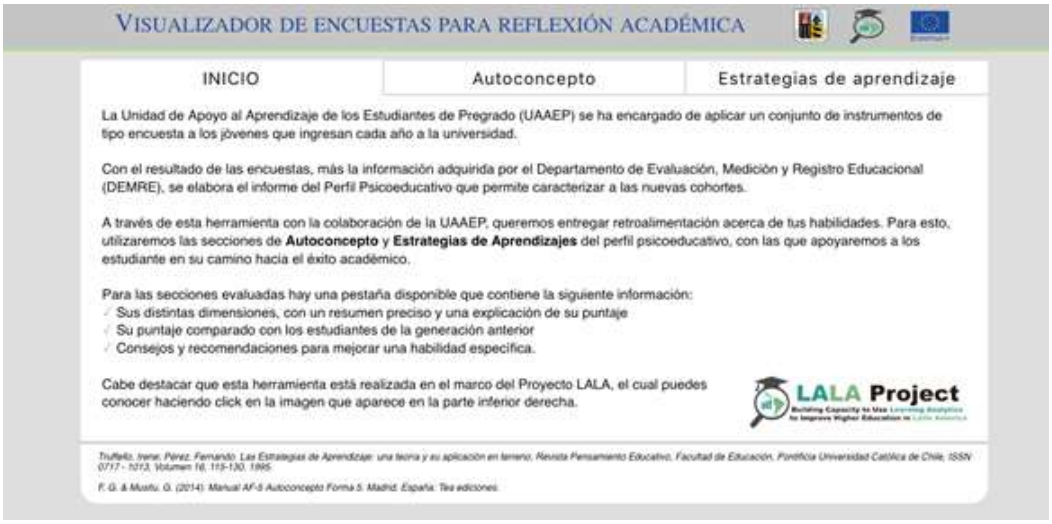


Figure 5.1.14 Home tab

Factors: a brief definition is shown that seeks the user to understand what was evaluated in the survey. Then, five buttons are shown that are associated with each of the dimensions that make up the self-concept. Each button has the function of displaying and displaying the results obtained in the survey (Figure 5.1.15).

Figure 5.1.15 Self-concept

In this part of the tool, an interface called «accordion» was used, which allows grouping the information in sections under a title so that each of the blocks can be expanded or contracted. In addition, it causes the user to click to see the information, which allows capturing these events for further analysis. In this way, the user interface is better exploited and, at the same time, the organization of the information is streamlined (Figure 5.1.16)

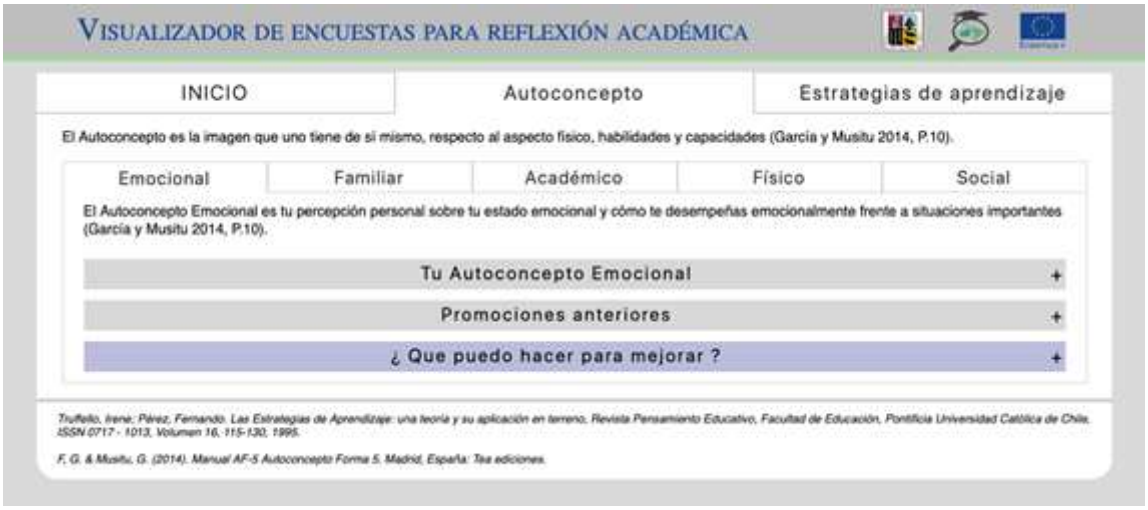


Figure 5.1.16 Self-concept with the accordion interface

Result: this visualization allows us to see the distribution of the results obtained by students belonging to the same cohort of the user. In addition, an arrow-shaped symbol is used to indicate the level obtained by the user (Figure 5.1.17)

Figure 5.1.17 Visualization of results

Historical information: shows a more complex visualization, which is characterized by showing information of students from previous years. According to the level obtained in the dimension of the survey (High, Medium or Low), and the number of failed subjects during the first semester after answering the survey type instruments, three different categories were formed within each level. Each category is represented with a certain color:

- Green: The student passed all his subjects.
- Yellow: The student failed a subject.
- Red: Failed two or more subjects

This representation is a percentage, which means that, if of 100 students who obtained a high level of emotional self-concept, 30 of them failed more than two subjects, then the number of frames in red at the high level (Persist) would be 30 (Figure 5.1.18)

Figure 5.1.18 Historical Information Display

Recommendations: The last section is used exclusively to show a recommendation, which depends only on the level obtained by the user. For each of the dimensions, there are two tips, one associated with the high level and the other linked to the medium and low level. The councils were prepared by the UAAEP, both for the self - concept survey and for the learning strategies (Figure 5.1.18).

The results of the learning strategies survey are composed of sections similar to those presented above.

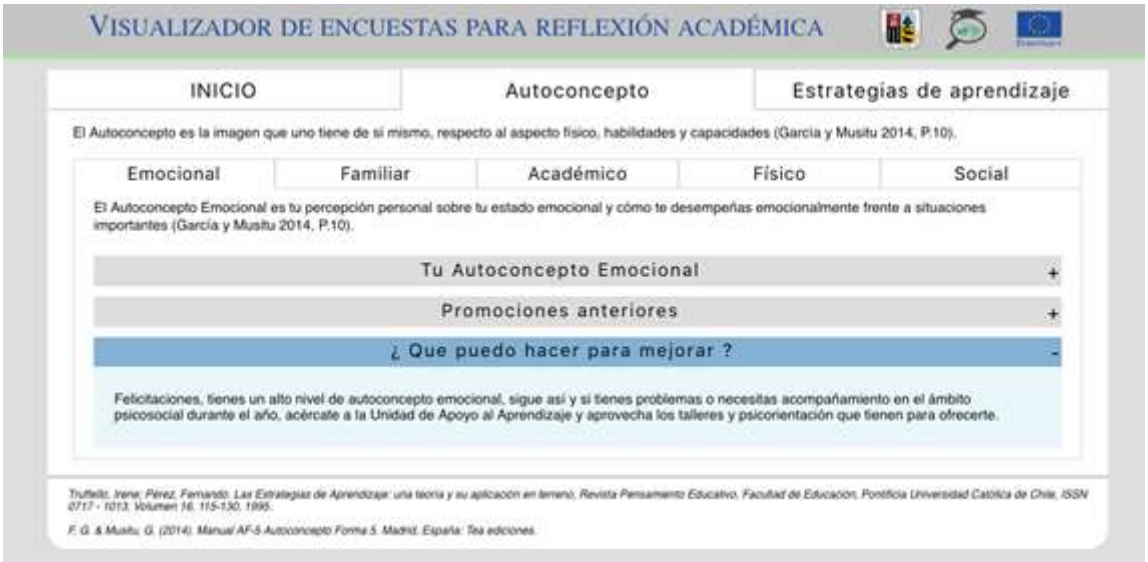


Figure 5.1.18 Visualization of recommendations

5.2 Ontask

OnTask is a tool that assists teaching staff to provide timely, personalized and actionable student feedback throughout their participation in a course in order to improve their academic experience. By providing frequent suggestions about specific tasks in the course, students will be able to quickly adjust their learning progressively. Figure 5.2.1 shows the flow of the data through OnTask. It receives data from different institutions' source files, like LMS or demographics, and generates a table with the attributes per student. These attributes can be used by the instructor to provide personalized feedback (action-out), and eventually, the students answer questions that are included on the initial table (action-in).

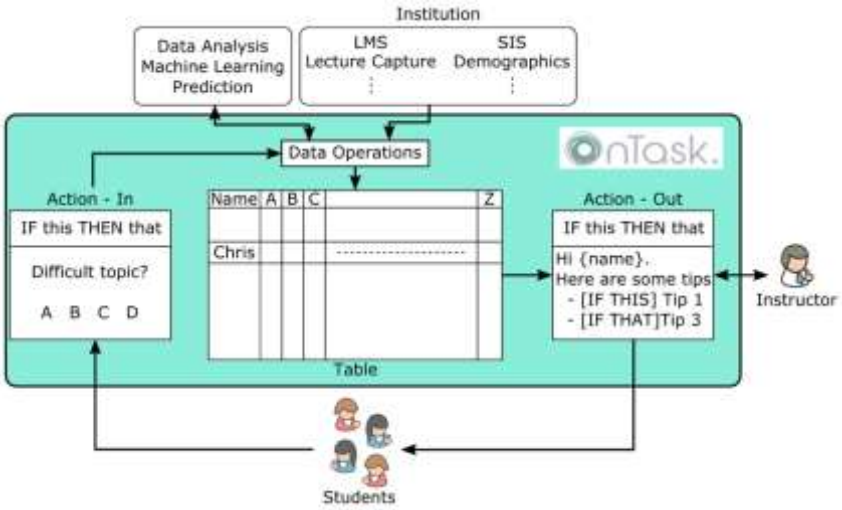


Figure 5.2.1 Flow of activities

The instructors can write personalized feedback through an interface that provides resources to create a set of ‘if…then’ rules. Figure 5.2.2 shows the interface to create a personalized email; it encapsulates general text, generic attributes like {{GivenName}} and the rules. The most significant benefit is to create messages to a large cohort of students without needed to write similar messages repetitively.
In order to adapt the tool, the universities in Latin America have to define which attributes would be used based on their LMS or SIS system. It means that the teacher will use the same user interface as shown in the figure below; however, the technical team should provide a service to integrate the data with OnTask to allow the use of it. This adaptation could be made by integrating the institution database with OnTask directly or by importing a CSV file into the system before use it.

5.2.2 Interface to create the personalized emails

It is important to mention that OnTask is an opensource tool, highly documented (https://github.com/abelardopardo/ontask_b) and we are using the Django version. Besides, the tool is LMS agnostic and receives its data from various sources such as video engagement, assessments, student information systems, electronic textbooks, discussion forums, etc. In order to set up the server to run OnTask, the partner should follow the step-by-step detailed on this link: https://abelardopardo.github.io/ontask_b/Install/index.html. As requirements, the server should have Python 2.7 and Python 3.6, Django 2.1.4, Redis, PostgreSQL (version 9.5 or later). The tool was developed using python based on Django and Pandas. Besides, it uses ORM and SQLAlchemy and a database decided by the institution that will adopt OnTask. Thus, in order to adopt it, the higher education institution needs to define which data source would be used as input, instantiate a database and a server (available here: https://github.com/abelardopardo/ontask_b), and finally integrate the uses to provide login accounts (Figure 5.2.3).

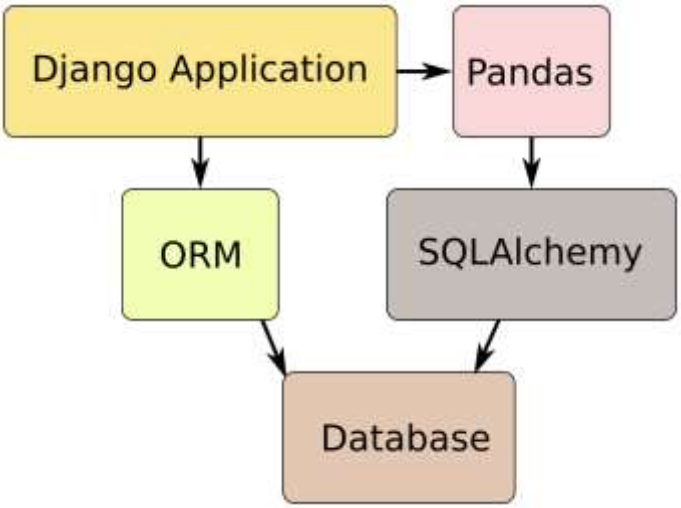


Figura 5.2.3 Backend Technologies

As part of LALA project, the University of Edinburgh is assisting the partners to adopt Ontask to their context. It involves both technical and pedagogical support. The technical support includes activities related to the instantiation of the tools (https://abelardopardo.github.io/ontask_b/Install/index.html) on the servers bought during the project, synchronization with the university data source (e.g., LMS) and the training for teaching staff that will use it.
On the pedagogical side, the team from the University of Edinburgh will assist the partners on the initial adoption by providing training on how to create the emails using the tools and what are the pedagogical issues that could happen during the use of OnTask.

Finally, the development process of the tool can be seen in https://www.ontasklearning.org/scenarios/ and https://abelardopardo.github.io/ontask_b/Introduction/index.html. Our focus on this document was on the adaptation of the tool to the Latin American partners.

6. Conclusions

This document presented the process that each university had in adapting the tools of counseling and early desertion, in its context. A common infrastructure was needed to separate privacy and security aspects according to the needs of the Latin American partners. Although a generic database was designed, it was changing depending on the institutional needs. The same applied to visualizations and intelligence algorithms. It is worth mentioning that all these processes were possible because they came from a previous survey of needs through the "LALA framework." Additionally, it was not started from scratch, but rather an adaptation of already created tools provided by the European partners. The inclusion of the end users of the tool in the design process was kept in consideration.

It is expected that the description of these steps for the four HEIs in Latin America will help other HEIs to do something similar. In this way, the process of adopting learning analytics tools for HEIs in Latin America is facilitated.

7. References

1. Bannan-Ritland, B.: “The role of design in research: the integrative learning design framework,” Educational Researcher, 32, 1 (2003), 21–24.
2. Brooke, John. "SUS-A quick and dirty usability scale." *Usability evaluation in industry* 189.194 (1996): 4-7.
3. Brusilovsky, P., Somyürek, S., Guerra, J., Hosseini, R. and Zadorozhny, V.: “The Value of Social: Comparing Open Student Modeling and Open Social Student Modeling”, in. Springer, Cham, (2015), 44–55.
4. Charleer, Sven, et al. "Learning analytics dashboards to support adviser-student dialogue." *IEEE Transactions on Learning Technologies* 11.3 (2018): 389-399.
5. Kizilcec, R. F., Pérez-Sanagustín, M. and Maldonado, J. J. (2017) “Self-regulated learning strategies predict learner behavior and goal attainment in Massive Open Online Courses”, Comp. & Education, 104, (2017), 18–33.
6. Kniberg, Henrik, and Mattias Skarin. *Kanban and Scrum-making the most of both*. Lulu. com, 2010.
7. Millecamp, Martijn, et al. "A qualitative evaluation of a learning dashboard to support advisor-student dialogues." Proceedings of the 8th international conference on learning analytics and knowledge. ACM, 2018.
8. Milligan, C. and Littlejohn, A.: “How health professionals regulate their learning in massive open online courses”, The Internet and Higher Educ., 31, (2016), 113–121.
9. Moreno-Marcos, P. M., Muñoz-Merino, P. J., Alario-Hoyos, C., Estévez-Ayres, I., & Delgado Kloos, C. (2018). Analysing the predictive power for anticipating assignment grades in a massive open online course. Behaviour & Information Technology, 37(10-11), 1021-1036.
10. Nielsen, J.: “10 Heuristics for User Interface Design: Article by Jakob Nielsen”, http://www.nngroup.com/articles/ten-usability-heuristics/. (1995)
11. Pérez-Álvarez, Pérez-Sanagustín, R. and Maldonado, J. J.: “How to design tools for supporting self-regulated learning in MOOCs? Lessons learned from a literature review from 2008 to 2016”, Proc. 42nd Latin American Computing Conf., CLEI 2016, (2016), 1–12.
12. Pérez-Álvarez, R., Pérez-Sanagustín, M., Maldonado-Mahauad, J. J.: “NoteMyProgress: Supporting learners’ self-regulated strategies in MOOCs”, in. Springer, Cham, (2017b), 517-520.
13. Pintrich, P., Boekaerts, M.: “The role of goal orientation in self-regulated learning”. In Pintrich, Paul R. (Ed); Zeidner, Moshe (Ed) Handbook of self-regulation. San Diego, CA, US: Academic Press, xxix, 783, (2000), 451-502.
14. R. Pérez-Alvarez, J. J. Maldonado-Mahauad, and M. Perez-Sanagustin, “Design of a Tool to Support Self-Regulated Learning Strategies in MOOCs,” J. Univers. Comput. Sci., vol. 24, no. 8, pp. 1090–1109, 2018.
15. Pérez-Sanagustín, M., Maldonado-Mahauad J.J, Pérez-Álvarez R., (2018): “UC Online Engineering”, XIII Conferência Latino-americana de Tecnologias de Aprendizagem LACLO 2018, pp. 460-446.
16. Reimann, P.: “Design-Based Research”, in Methodological Choice and Design. Dordrecht: Springer Netherlands, (2011), 37–50.
17. Ssemugabi, S., Villiers, R.: “A comparative study of two usability evaluation methods using a web-based e-learning application”. Proc. of conf. SAICSIT '07, (2007), 132-142.

18. Veletsianos, G., Reich, J. and Pasquini, L.: “The Life Between Big Data Log Events Learners, Strategies to Overcome Challenges in MOOCs”, AERA Open, 2(3), (2016), 1–10