Center for the Study of Western Hemispheric Trade
Texas A&M International University

WORKING PAPER SERIES
WP 2024-006
August 2024

Development of a Web-Based Timetabling Software for a Mexican University

Jesus Abraham Garcia Yañez
Felipe De Jesus Ramirez Saenz
Eva Litzary Rios Saldaña
Mauricio Alberto Rubio Aguilar
Itzia Esperanza Barrios Flores
Camila Lineth Garcia Flores
Ramon Estuardo Hernandez Cerda
Jose Julian Zacarias Ortiz

www.tamiu.edu/cswht

Development of a Web-Based Timetabling Software for a Mexican University

Jesus Abraham Garcia Yañez[1]
Universidad del Noreste A.C., Tampico, Tamaulipas, México

Felipe De Jesus Ramirez Saenz
Universidad del Noreste A.C., Tampico, Tamaulipas, México

Eva Litzary Rios Saldaña
Universidad del Noreste A.C., Tampico, Tamaulipas, México

Mauricio Alberto Rubio Aguilar
Universidad del Noreste A.C., Tampico, Tamaulipas, México

Itzia Esperanza Barrios Flores
Universidad del Noreste A.C., Tampico, Tamaulipas, México

Camila Lineth Garcia Flores
Universidad del Noreste A.C., Tampico, Tamaulipas, México

Ramon Estuardo Hernandez Cerda
Universidad del Noreste A.C., Tampico, Tamaulipas, México

Jose Julian Zacarias Ortiz
Universidad del Noreste A.C., Tampico, Tamaulipas, México

This article shows the significant progress of a technology project that is currently being developed at the Universidad del Noreste in Tampico, Tamaulipas, Mexico. In the near future, the project will generate automatic and intelligent schedule planning through algorithms specialized in resource management in the area of engineering and chemical sciences of said university.

As of the date this document, the system allows teachers responsible for scheduling to do so in an intuitive and practical way using a web application. The system allows professors to enter their work schedule preferences, such as available days and hours, and considers the university's requirements regarding working hours for each professor. It also considers various restrictions, for example, the subjects (courses) that can be taught, each teacher, schedules defined by the language area and programmed for students, availability of classrooms, etc. The manual management that was used until a few months ago produced many errors and conflicts in academic programming, affecting both the administration and the academy, since many man-hours were invested in this process, which, because it was not automated, had many weaknesses.

The web application created makes it easy to find viable scheduling solutions that satisfy the requirements and restrictions. It has also been prepared so that, when integrated with intelligent algorithms, it can propose optimal and efficient solutions in the generation of schedules.

[1] Address correspondence to Jesus Abraham Garcia Yañez, Universidad del Noreste A.C., Tampico, Tamaulipas, México. Email: a182048@une.edu.mx

I. Introduction

Efficient scheduling in academic institutions is crucial to ensure the optimal utilization of resources, meet the diverse needs of students, and facilitate the smooth functioning of educational programs. However, manual scheduling processes are often arduous and error-prone, particularly at large institutions with complex restrictions and preferences.

From the engineering point of view, the problem is traditionally solved through combinatorial optimization with operations research techniques and/or heuristic algorithms (Schaerf 1999; Phala 1988; Ernst et al. 2004). In the last two decades, metaheuristic algorithms have provided advances and improvements in the quality of the solution but at a much higher computational cost (Phala 1988; Burke and Petrovic 2002; Babaei, Karimpour, and Hadidi 2015; Bashab et al. 2020). With the advancement and development of the various branches of artificial intelligence, machine learning and genetic algorithms have emerged to provide very effective results with this type of problem (Chen et al. 2021). Recent publications have made an exhaustive review of the various methodologies and recent contributions reported for timetable design in school environments (Ceschia, Di Gaspero, and Schaerf 2023; Chen et al. 2021). Without a doubt, there is currently a growing interest in taking advantage of computational methods to automate the generation of academic schedules, with the aim of streamlining the scheduling process and improving overall efficiency.

Automated scheduling systems harness the power of computational algorithms to generate schedules that adhere to various constraints and objectives, such as room availability, faculty preferences, student course selections, and curriculum requirements. These systems offer the promise of reducing the time and effort invested in scheduling tasks while producing schedules that are optimized for key metrics such as minimizing conflicts and maximizing resource utilization.

This research aspires to contribute to the burgeoning field of automated academic scheduling by presenting a comprehensive study of a software project dedicated to this endeavor. Our software project aims to develop a robust and flexible scheduling solution tailored to the specific needs of Universidad del Noreste in Tampico, Tamaulipas, Mexico.

1.1 Motivation

This project addresses the problem of manual schedule timetable planning in the engineering area of the Universidad del Noreste in Tampico, Tamaulipas, Mexico. The current management is carried out manually by obtaining information from many sources (mainly excel spreadsheet files) and without order (Figure 1). The director of the engineering area and a support teacher oversee the collection of all the information needed to manually plan and create feasible timetable schedules of the engineering careers for each academic period. The above leads to errors and conflicts in academic programming, affecting both the administration and the experience of teachers and students, in addition to the large number of man-hours invested in this tedious work.

The proposal for the university consisted of developing a system with well-defined roles (Figure 2) and an intuitive user interface for an automated system for generating academic schedules. The global project will be developed in three stages. This document shows the results of the first stage. The objective of the development project is to automate the processes in the work of academics in addition to optimizing the allocation of courses, classrooms, and all resources involved, reducing conflicts and maximizing efficiency.

Figure 1: Planning timetables for the Universidad del Noreste (previous situation).

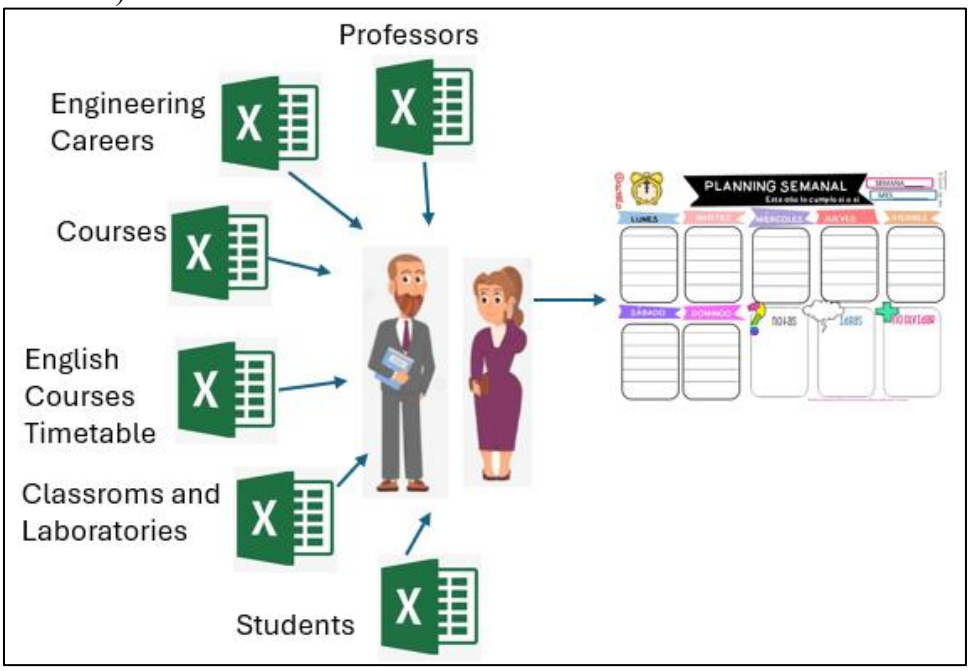
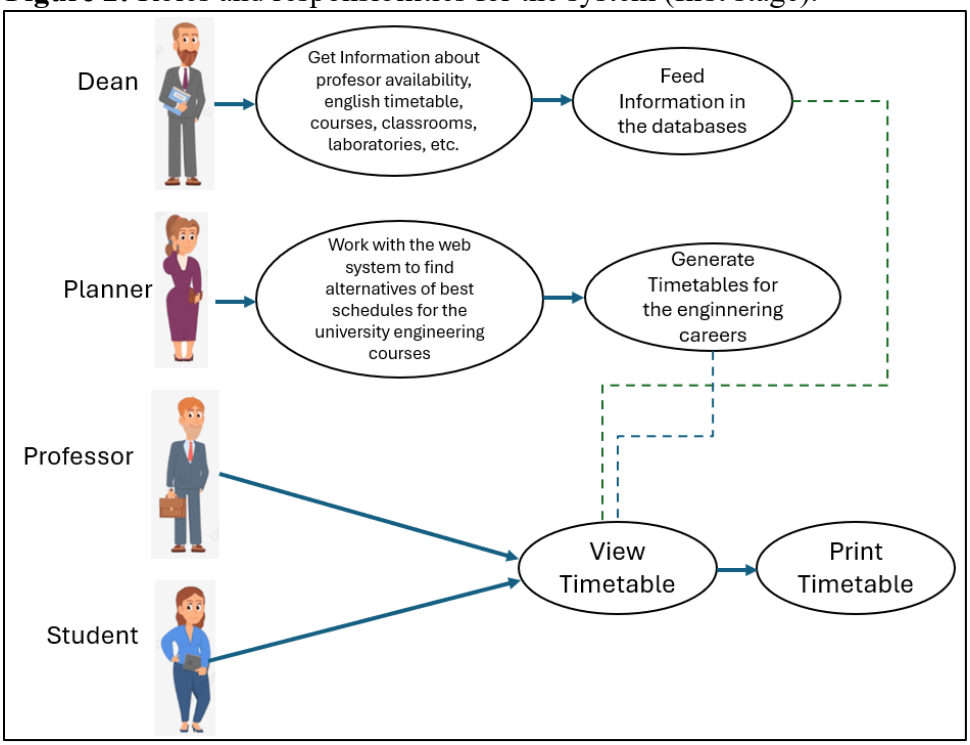


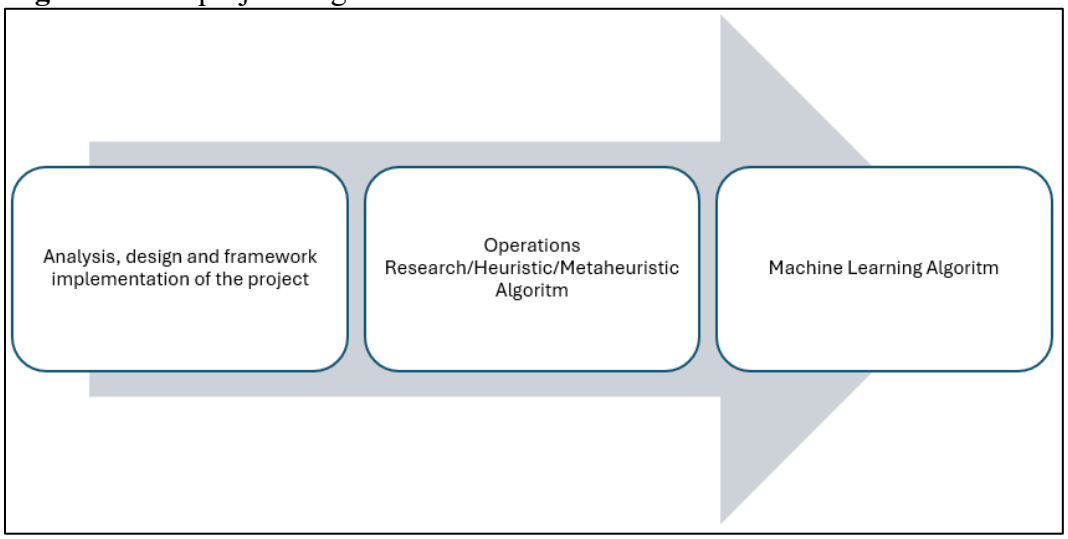
Figure 2: Roles and responsibilities for the system (first stage).



1.2 Expanded Project Vision

This project consists of three stages. The first is documented in this article, and it represents the backbone of the software which allows for the planning of schedules for the school periods of the engineering careers of the university in an easy, friendly way and with an intuitive interface. The plan for the second stage is to integrate with operations research methods or heuristic or metaheuristic algorithms. The third stage consists of integrating machine learning so that the system can learn to generate school schedules more efficiently.

Figure 3: Full project stages.



1.3 Outline of the Remaining Sections

In the following sections, we review the existing literature on automated scheduling algorithms and highlight the challenges and opportunities in applying these techniques to academic scheduling contexts. Then, we present the methodology employed in the development of the software project, including the analysis, design, and frontend and backend development, and we describe the implementation details of the software, outlining its architecture, functionalities, and user interface components.

II. Literature Review

A summary of the most relevant projects related to this article is shown in Table 1 (Ong 2020; Devin and Le Nir 2010; Priyadarshana 2021; Omoregbee et al. 2022; Raju and Mangal 2017; Kembuan et al. 2018); the main characteristics of the implementations are described there. Only the web based, or applications of the timetable educational system are shown since they are somehow similar to the work described in this paper.

Table 1: Implementations of timetable educational applications reported.

Author/Year	Title of the Article	Summary
Ong (2020)	Web-based Scheduling Software for a University	This thesis provides insight into the process of creating the scheduling system, including the connection to the database and the design of a backtracking algorithm and a genetic algorithm.
Devin and Le Nir (2010)	On-line Timetabling Software	The paper discusses the challenges of timetabling and presents a software solution that addresses data acquisition and timetable computation. The software uses advanced technologies, such as Rich Internet Application and constraint programming in SWI-Prolog, to automate the timetabling process. The use of web services allows for interaction with Google Calendar, and the software is built in the IT system. The paper also includes a real case study to demonstrate the effectiveness of the software.
Priyadarshana (2021)	Web Based Timetable Management System for University of Vocational Technology	This dissertation presents a web-based timetable management system for UNIVOTEC in Sri Lanka. The manual process of managing the university timetable is time-consuming and prone to errors. The system aims to automate the timetabling process and provide management information system reporting. The focus is on user login, course, program, lecture, staff, and student details, as well as time slot, room, lab, class, batch, semester, and department details. The system also includes timetable allocation and generation processes.
Omoregbee et al. (2022)	Web-based Student Time-Table Management System	The article discusses the development of a web-based student time-table management system using a genetic algorithm to solve the complex problem of scheduling lectures and practical timetables for many courses. The system was developed and verified using PHP and MySQL programming languages, and it allows for the generation of different schedules based on user-specified limits and requirements.
Raju and Mangal (2017)	Web-based Application for Automatic Timetable Generation	The paper discusses the development of a web-based application for automatic timetable generation in educational institutions. It addresses the challenges of timetabling problems being NP-hard and the limitations of existing software solutions. The system aims to simplify the process by generating high-quality timetables for two shifts, handling teacher information, feedback collection, defaulter lists, and performance graphs. The focus is on reducing manual work, improving efficiency, and providing a cost-effective solution for educational institutions.
Kembuan et al. (2018)	Development of Web based Timetabling System	The aim of this research was to develop a web-based timetabling system to optimize the resources using a genetic algorithm. The algorithm was tested with the real data containing 47 combined lessons data to be scheduled into 40 timeslots and 8 rooms. The research method used is a design or experimental method. Rapid application development (RAD) in a system development life cycle (SDLC) model was used as the system development methodology of this timetabling system. PHP programming language and MySQL were used in this timetabling application. The results showed that the proposed timetabling system successfully minimizes processing time and provides the optimal solution for the problem.

There are several methodologies which have been proposed for the development of the timetable system, depending on various parameters and requirements. Most researchers used metaheuristics or machine learning techniques for optimization. However, each proposal is for a specific academic institution and addresses different problems with diverse constraints. Furthermore, industry-based solutions are expensive and charge huge annual licensing fees. Therefore, in this paper, we present the web-based platform to develop an efficient timetable system with an attractive graphical user interface (GUI) which reflects and solves our institutional problems and issues. The design of the platform described has been carefully and meticulously created so that any future changes or restrictions can be integrated without major problems.

III. Methodology

3.1 Analysis

The analysis of information was carried out with the personnel involved in the timetable planning of the engineering area of the university. The most relevant information presented in this part of the system development process is shown in this section.

3.1.1 Requirements

In the realm of web application development, success hinges not only on the aesthetic appeal of the interface or the richness of features but also on the fulfillment of essential functional and non-functional requirements. These two categories of requirements play pivotal roles in shaping the functionality, performance, and user experience of web applications.

Functional Requirements
Functional requirements define the specific behaviors and functionalities that a web application must exhibit to satisfy user needs and achieve its intended purpose. These requirements encompass features such as user authentication, data input forms, search functionalities, content management systems, and interactive elements like chatbots or recommendation engines. Functional requirements serve as the building blocks of a web application's functionality, delineating the actions users can perform and the outcomes they can expect.

Non-functional Requirements
In contrast, non-functional requirements address the attributes that characterize the overall performance, reliability, security, and usability of a web application. These requirements are not concerned with specific functionalities but rather with qualities such as scalability, responsiveness, availability, accessibility, and security. Non-functional requirements ensure that the web application operates efficiently, reliably, and securely under varying conditions, catering to the diverse needs and preferences of users while adhering to industry standards and best practices.

Both types of requirements were collected at the time of conducting interviews with all those interested in the development of the system presented in this article.

Table 2: User Types, Functional Requirements, and Non-functional Requirements.

User Types	*Functional Requirements*	*Non-Functional Requirements*

Administrator: responsible for making all the information available for the system to work correctly; oversees contacting all the teachers involved to gather their information.
 	
Coordinator: Oversees all the planning for the system.
 	
Professor: Person who uploads his/her personal availability information to the system.
 	
Student: Future implementation
	• System home page
	  ○ The system home page displays only user login form and help option.
	  ○ System identifies correct user type and directs them to the appropriate pages with given user privileges.
	  ○ Login username will be the same as the student registration number.
	  ○ The default password will be the e-mail address, and students will be able to change their password as needed.
	  ○ If "Incorrect Username or Password" message appears three times, then the user will need to contact the administrator to reset their password.
	  ○ There is an accurate validation for username and password.
	  ○ Successful user login will show a welcome message and direct the user to the employee home page based on their user category.
	• The course registration home page must allow users to register for courses individually and in groups using an Excel file.
	• User registration home page must allow professors to register individually and in groups using an Excel file.
	  ○ The coordinator and teachers must be able to display their timetable and print it to a PDF file.
	  ○ The coordinator must be able to print the schedule of the races in the area in a period indicated in the interface.
	  ○ The web system must avoid conflicts in schedules, classrooms, teachers, excess or decrease of academic load assigned to each professor.
	  ○ Once all the information has been loaded into the databases, the coordinator will be able to create the plan in a completely graphical interface using the mouse and dragging function to organize the schedule.
	• English courses home page must allow the graphic integration of the timetables of the language courses so that these timetables are "blocked" in the scheduling of school timetables.
		• Shall be platform-independent, i.e., shall run on any platform (cellular, tablet, PC, etc.)
		• Shall maintain high security and be highly reliable
		• It must be compatible with the institutional colors of the university.
		• Except for the authorized user, no personal information shall be disclosed to system operators or other users.
		• The system shall respond quickly (0.5 seconds or faster) to user actions.
		• Shall provide a graphical user interface (GUI) for all end-users of the system
		• The system shall be easy to learn (and work with) and be able to learn by example.
		• Shall maintain a help facility for "how to use"
		• The application server should run on Google® technology (web server and database server).

3.1.2 Interviews

A set of interviews was conducted with the engineering professors responsible for the process of generating schedules to understand the mechanism used to generate them in school periods. The functional and non-functional requirements of the system were collected. The user types and system requirements for the developed stage are shown in Table 2.

3.1.3 Processes

There is a diversity of processes that are carried out when designing the schedules of the school periods within the engineering area of the university. To not make this article too long, the processes that are considered representative are shown and described in Figure 4. All the processes of the generation of academic schedules in the university were automated within the implementation of the web system.

The professor availability process (Figure 4a) refers to the collection and recording of a teacher's time availability for a specific school period. The process begins with the selection of the school year and periodicity by the professor, then s/he accesses a calendar that shows all the hours available during the selected school year. At this point, s/he proceeds to mark the hours in which s/he is available to lecture classes. Once the professor has completed the selection of his/her available hours in the calendar, the system provides the option to save the information.

The professor registration procedure (Figure 4b) is a procedure carried out by the person in charge of the engineering area (administrator). First, the person in charge of the area selects the corresponding type of professor, then proceeds to enter the professor's personal information, including his or her username, name, and password, which will allow the teacher to identify and access the platform. Subsequently, the career and school period to which the professor will be associated is selected. Once these parameters are defined, the person in charge accesses a set of available subjects (courses). At this point, the selection of subjects (courses) that the professor will be able to teach is carried out. Finally, the changes are saved in the system.

The schedule generation process (Figure 4f) is the responsibility of the administrator. Once inside the system, the coordinator selects the school year in which the classes will be held as well as the course and corresponding period. Next, s/he proceeds to select the subjects that will be part of the schedule. At this stage, the person in charge has the option of dividing a subject into cases where it is necessary.

Figure 4: Some of the processes in the development of the university timetable (Spanish version).

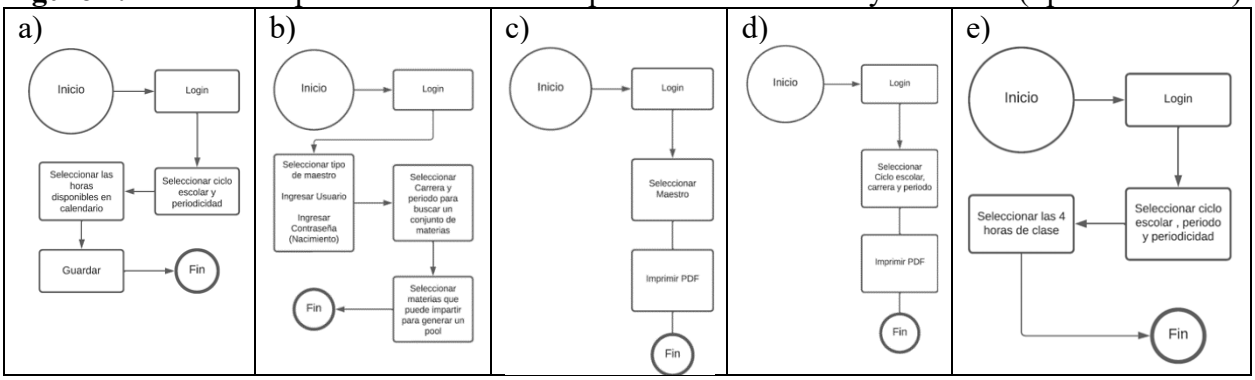
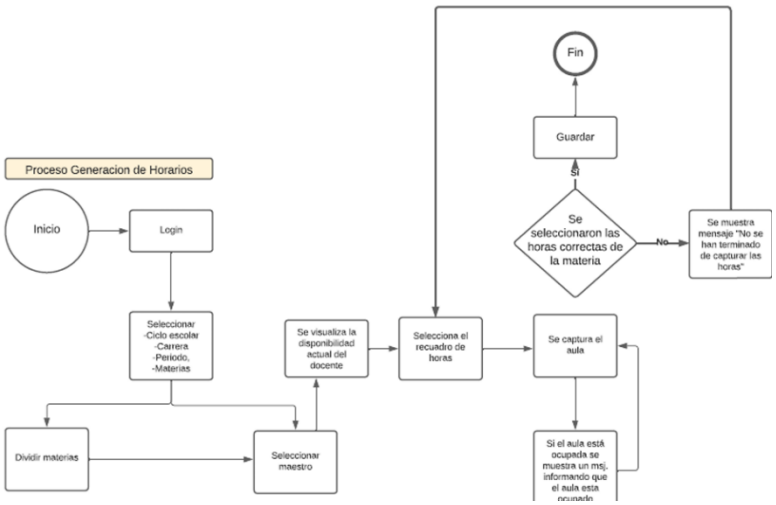
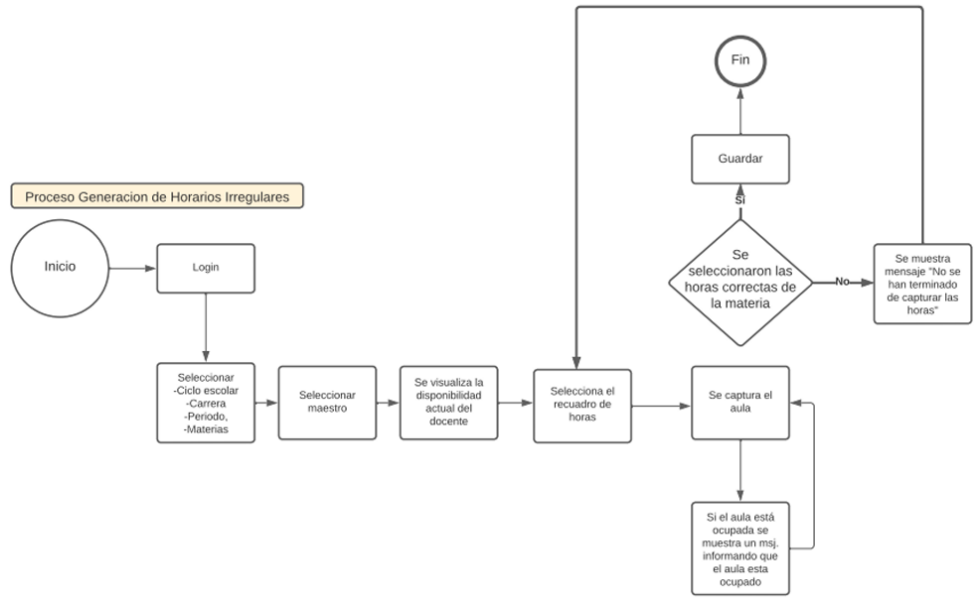


Figure 4. (continued)

f)



g)



Subsequently, the professor who will teach the subject is selected. At this point, the professor's availability is displayed, allowing the teacher to make informed decisions about class assignments. The specific times in which the classes will be held are then selected, and the classroom in which the classes will be held is also captured. It is important to mention that if the person in charge has not selected the required number of hours for a subject (course), the system will not allow them to progress to the next subject (course). Finally, once all the assignments are done correctly and the stipulated hours are met, the changes are saved.

The irregular schedule generation process (Figure 4g) is the responsibility of the administrator. Once inside the system, the administrator selects the school year in which the classes will be held as well as the course and corresponding period. Next, we proceed to select the subjects (courses)

that will be part of the schedule. Subsequently, the teacher who will teach the subject (course) is selected. At this point, the professor's availability is displayed, allowing the professor to make informed decisions about class assignment. The specific times in which the classes will be held are then selected, and the classroom in which the classes will be held is also captured.

It is important to mention that if the administrator has not selected the required number of hours for a subject (course), the system will not allow them to progress to the next subject. Finally, once all assignments are completed correctly and the stipulated hours are met, the changes are saved.

The schedule printing process for professors (Figure 4c) can be used by any system user type. Once inside the system, the user selects the teacher for whom he or she wishes to generate the schedule, which allows access to information related to that professor's class scheduling. Subsequently, the user activates the function by clicking on the "Print PDF" button. The print action generates a PDF file containing the specific schedule for the selected teacher. This PDF file is produced so that it is readable and suitable for printing.

The schedule printing process (Figure 4d) is exclusive to the administrator and planner. Once inside the system, the manager selects the school year as well as the course and corresponding period for which he or she wishes to generate the schedule, which allows access to information related to class scheduling for those parameters. Subsequently, the manager activates the function by clicking on the "Print PDF" button. The print action generates a PDF file containing the specific schedule for the selected period-year-career. This PDF file is produced so that it is readable and suitable for printing.

The English schedule generation process (Figure 4e) is the responsibility of the administrator. Once inside the system, the person in charge of the department selects the school year in which the classes will be held as well as the corresponding period and periodicity. Next, s/he proceeds to select the four-hour class that will be part of the English course schedule for the week.

3.2 System Design

The design of the web solution was divided in a very general way into two parts: the development of the project in (1) the front end and (2) the back end. Both projects from different student teams were integrated, and finally, the solution created was tested.

Front-end development and back-end development are two very important pieces of the software development world. These two components work together to improve the functionality of websites and web applications. While the front end and back-end work closely together in web development, these two kinds of development types involve different tasks.

Front-end development focuses on the client-facing, usability, and user experience aspects of a website or web application. This includes designing and optimizing the user interface and working on important visual aspects of web pages.

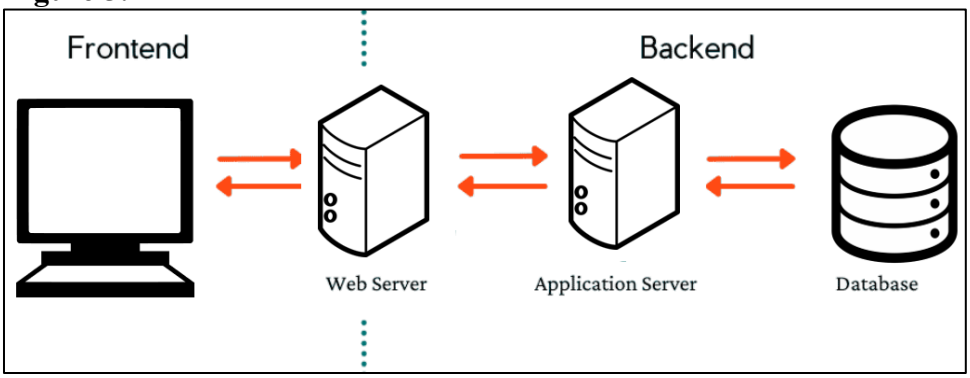
Back-end development focuses on the server-side aspects of a website or web application. This type of development is concerned with website architecture, scripting, and communication with databases. Back-end code enables the communication between browsers and information from databases. Back-end developers focus on how a website functions, which means they might work with application program interfaces (APIs), code that interacts with databases, libraries, data architecture, and more. Back-end development works together with front-end development to provide users with a functional and interactive experience. Table 3 outlines some of the key objectives and responsibilities of both front-end and back-end web development.

Table 3: Font-end and back-end tasks in web design.

Front-End Tasks	Back-End Tasks
Create user interfaces that are visually appealing and intuitive.	Develop server-side logic and databases to ensure the smooth functionality of web applications.
Implement responsive design to ensure compatibility across various devices and screen sizes.	Manage data storage and retrieval, including handling user inputs and database interactions.
Enhance the user experience through interactive elements and smooth navigation.	Optimize performance and scalability of the server-side code to handle increasing traffic and data loads.
Ensure that cross-browser compatibility and accessibility standards are met.	Implement security measures to protect sensitive data and prevent unauthorized access.
Collaborate with designers to translate mockups and wireframes into functional web interfaces.	Integrate third-party services and APIs for additional functionality and features.
Utilize HTML, CSS, and JavaScript frameworks/libraries for front-end development.	Use server-side languages such as Python, Ruby, PHP, or Node.js along with frameworks like Django, Rails, Laravel, or Express.js for back-end development.

Figure 5 shows a simplified graphical version of the boundaries of back end and front end.

Figure 5: Back end and front end.

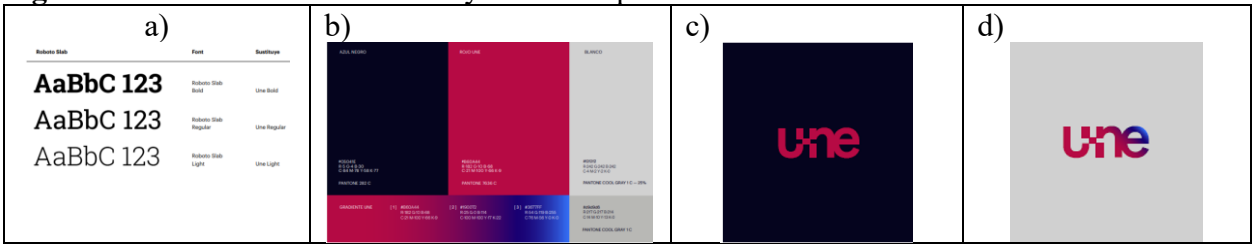


3.2.1 Front-end Development

Visual Identity and Color Palette

Universidad del Noreste has an official visual identity that must be respected in the development of any software system to be used within the institution. Figure 6 shows the fonts, color palette (a and b), and application to the university logo (c and d) used in the visual part of development.

Figure 6: Institutional visual identity and color palette of the Universidad del Noreste.



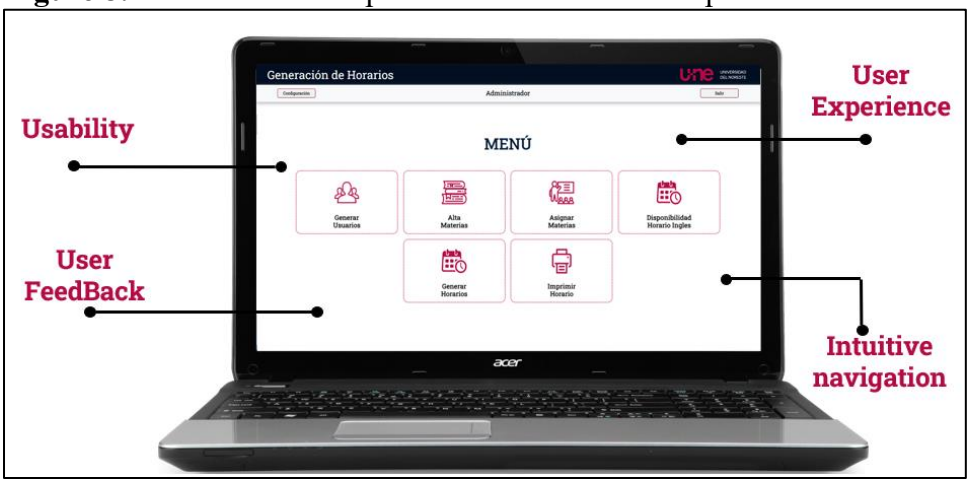
The visual identity was applied to all designed screens; this is shown in Figure 7.

Figure 7: Institutional visual identity and color palette applied in the interfaces.

User Interface (UI), Usability, and User Experience (UX)

User interface (UI) design focuses on creating visually appealing and functional interfaces that facilitate interaction between users and digital systems. It involves elements such as layout, typography, color schemes, and interactive components to ensure a seamless user experience. Usability, on the other hand, refers to the ease with which users can navigate and interact with a website or application. It encompasses factors like intuitive navigation, clear information architecture, and efficient task completion. User experience (UX) incorporates the overall perception, emotions, and feelings users derive from interacting with the application. It delves into understanding user needs, behaviors, and emotions to design experiences that are meaningful, enjoyable, and valuable. Together, UI, usability, and UX form the foundation of effective web front-end programming, guiding developers in creating interfaces that are both aesthetically pleasing and user-friendly and ultimately enhancing the overall user experience.

Figure 8: Foundation concepts of the front-end development.



To achieve an adequate design in the aspects of usability and UX in the developed web system, the following stages were followed:

i) Investigation: During this stage, the greatest amount of information possible and necessary for the project to be carried out was obtained through interviews with those involved. The information obtained in this stage included: definition of scenarios, understanding of the context, definition of user profiles, definition of the contents of the web application, definition of the general requirements of the project, and characterization of the typology and colors to be used.

ii) Organization: During this stage, it was possible to represent all the possible structures of the contents in accordance with the needs of the users and context of use, and the functional flows of the system (navigation map and flowcharts) were also defined.

iii) Prototyping: The screens of the web system were defined, proposing several versions of them and evaluating the perception of the end users; the final functionality of the product was also defined, and prototypes of the application interfaces were created.

Technologies Used in the Front End
The main technologies used for the front-end development were Github like version manager, the Angular framework from Google, HTML and CCS for the format and the page style, and some support libraries.

3.2.2 Back-end Development

The main task of the back end in the development carried out is to implement the interaction with the database of all the designed interfaces, maintain its integrity, and implement all the business logic of the problem being solved. Figure 9 shows the different actions taken in the navigation of the main particular tasks. Task 9i refers to the registration of teachers and courses; 9ii is the task of schedule generation; 9iii is the task of division of groups (when a group is too large to fit in a classroom or laboratory and must be divided into two, for example, Group A and Group B); 9iv refers to the English schedule, which is the schedule assigned for all students to study English at different levels; 9v is the registration of English teachers; 9vi is the schedule consultation per professor or by academic period; and 9vii refers to professor group registration.

Figure 9: Navigation tasks to perform actions in the backend of the system (Spanish).

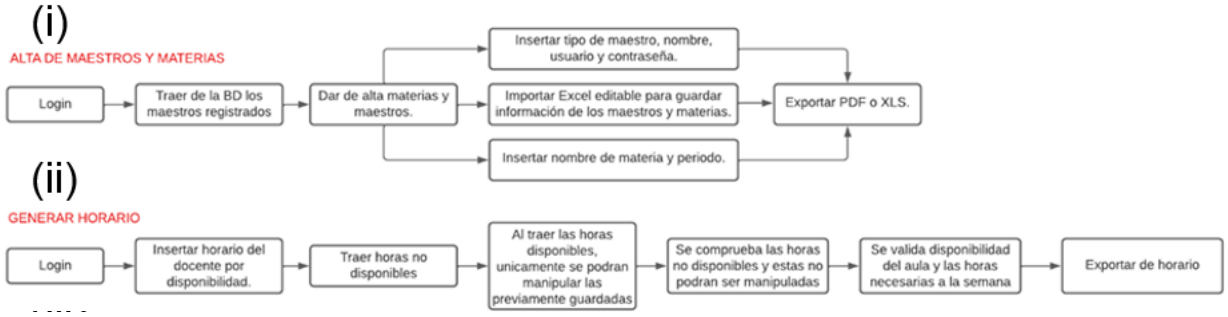
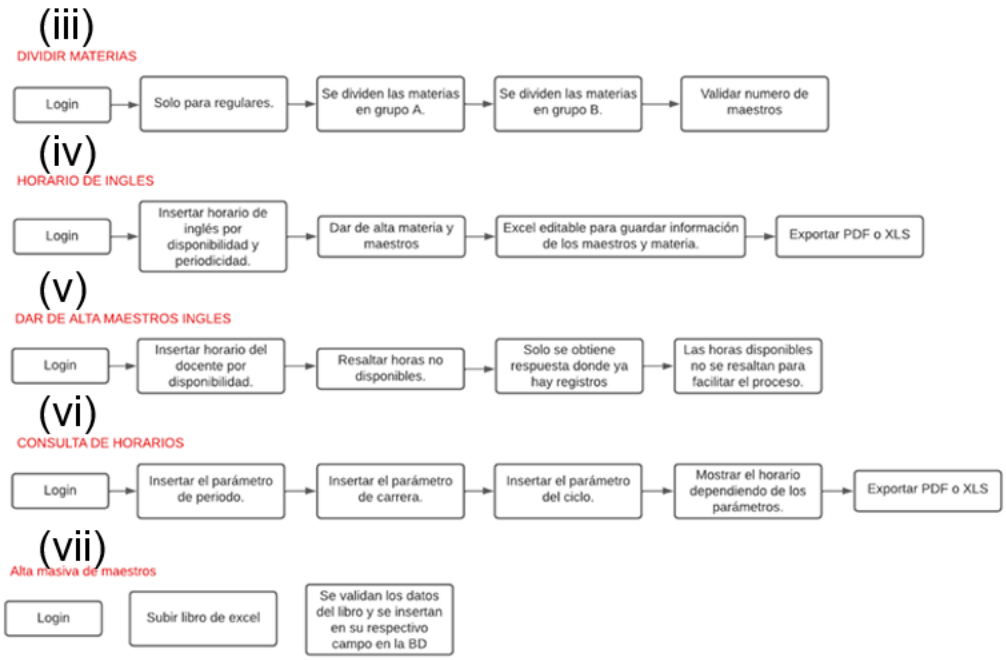


Figure 9. (continued)



The database used is a non-relational database called Firebase by Google; the different object structures stored are shown in Figure 10.

Figure 10: Structures stored in the non-relational database Firebase (Google).

User

Field	*Data Type*
IdUser	String
UserType	String
TeacherType	String
Username	String
Name	String
Mail	String
Password	String
Asignatures	String[]
Hours	Int
TotalHours	Int
RecentChanges	Bool

Career

Field	*Data Type*
IdCareer	String
Name	String
Periodicity	String
PeriodQuantity	Int

Asignature

Field	*Data Type*
IdAsignature	String
Name	String
Career	String
Period	Int
Hours	Int
AvailableHours	Int

Register

Field	*Data Type*
Field	Any[]

TeacherAvailability

Field	*Data Type*
IdAvailability	String
TeacherName	String
Cicle	String
Availability	Register[]

EnglishAvailability

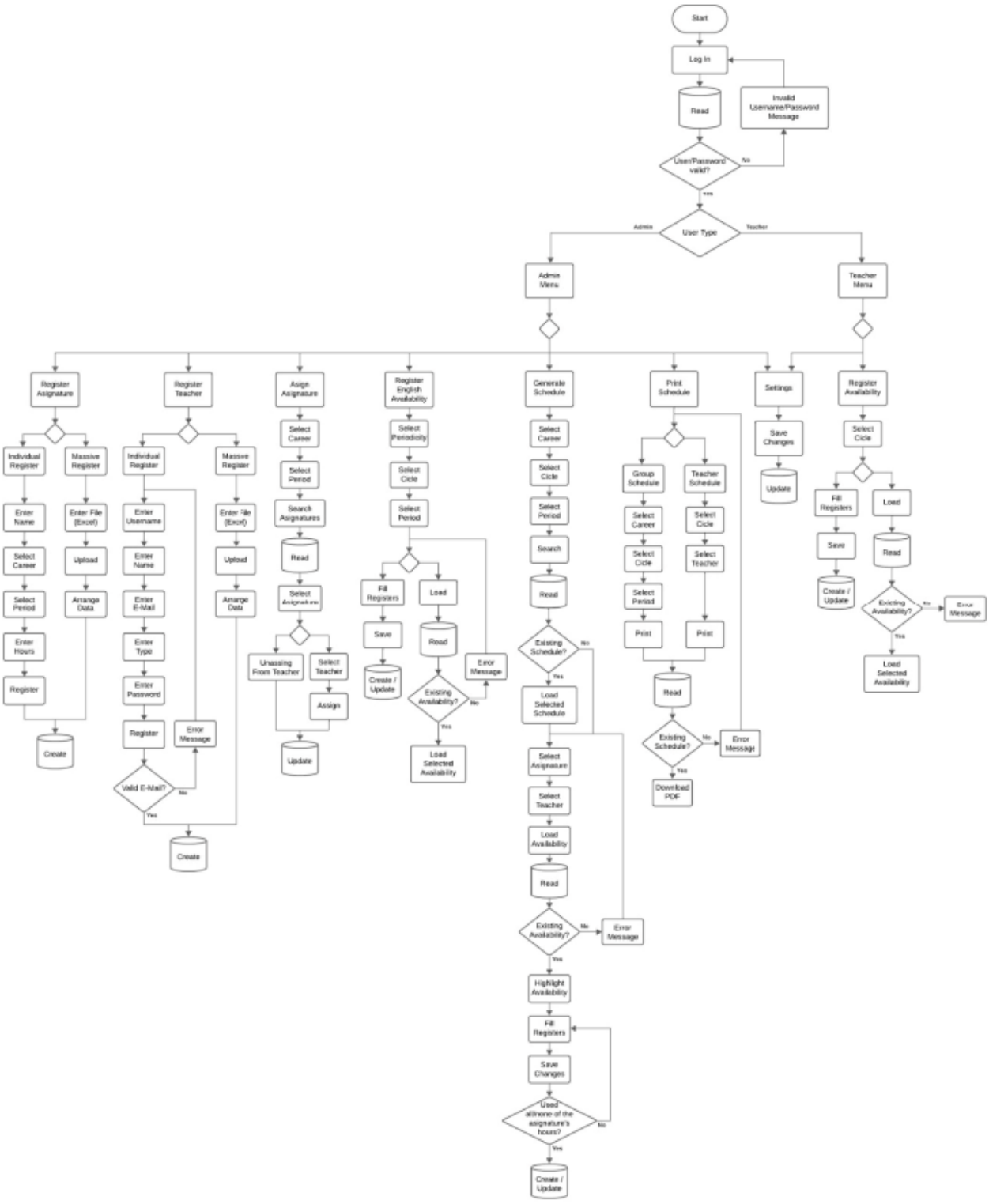
Field	*Data Type*
IdAvailability	String
Name	String
Cicle	String
Period	Int
Periodicity	String
Availability	Register[]

Schedule

Field	*Data Type*
IdSchedule	String
Career	String
Cicle	String
Period	Int
Teachers	String[]
Asignatures	String[]
Regular	Register[]
Irregular	Register[]

A graphical design in a flowchart of the algorithms used is shown in Figure 11.

Figure 11: Algorithms used in the backend.



IV. Results

4.1 Developed Interfaces

The main interfaces developed with which a user interacts with the web system are shown in this section. To optimize data entry and improve the fluidity of user interaction with the screen, various restrictions have been incorporated into the available fields. These constraints are designed to guide the user through a sequential data entry process. In this context, when completing a specific field, the next field in the sequence will be activated automatically, thus allowing the user to fill in the information progressively. Additionally, a logical and coherent arrangement of the fields on the screen has been arranged. This organization has been structured from left to right and top to bottom with the goal of providing an intuitive and easy-to-understand presentation.

Figure 12 shows the login screen. Users provide their user ID as well as their password. Once these data have been entered, the user must click the "LOGIN" button to access the software web platform.

Figure 12: Session login interface (Spanish version).

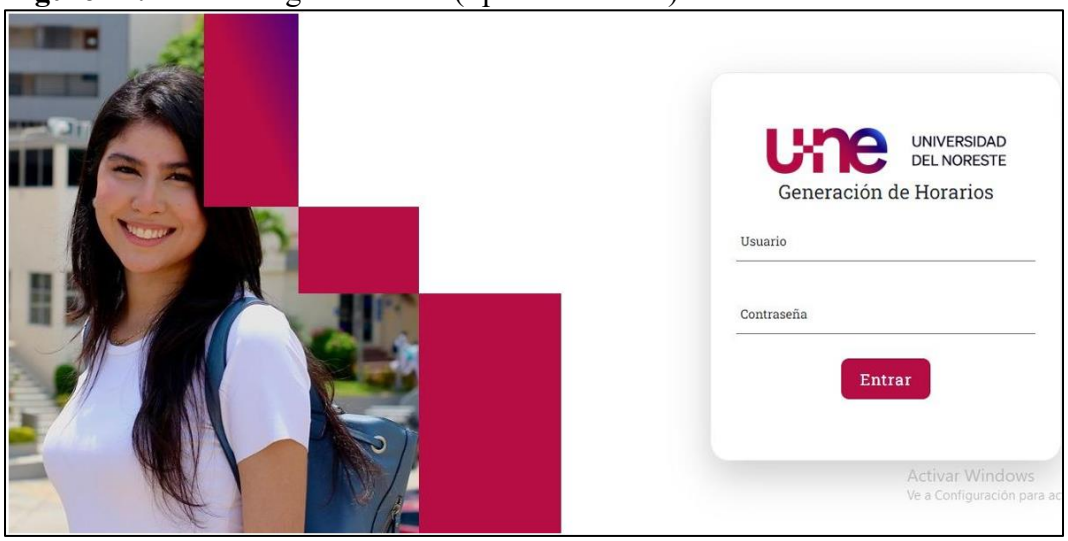


Figure 13: Administrator menu interface (Spanish version).

The “Administrator Menu” interface shown in Figure 13 presents a set of actions that are available to the administrator; these options include the ability to generate users, register subjects, assign subjects, assign the availability of the English schedule, generate schedules, and print schedules. Additionally, at the top of the screen, there is a navigation bar containing two additional buttons. One button allows the administrator to access their profile settings while the other button allows the user to log out of the application. This screen offers a clear and organized interface that makes it easy to manage the actions relevant to system administration.

The “Settings” interface shown in Figure 14 provides the user with the ability to modify certain elements of their profile. Options available for editing include username and password. However, it is important to note that the user has the ability to view their name and email address but does not have the power to make changes to these fields. This screen provides an efficient means of managing specific aspects of user profile information.

Figure 14: Configuration interface (Spanish version).

Figure 15: Registration of individual and group users’ interface (Spanish version).

The “User Registration” interface (Figure 15) allows the administrator to register teachers individually or in groups through an efficient process. For individual registration, the administrator must provide specific information for each teacher, including a username, name, email address, corresponding teacher type, and password. In addition, the screen offers the alternative of carrying

out a mass registration of teachers through the option of uploading a file in Excel format, which speeds up the process by allowing the incorporation of multiple teachers simultaneously. This screen offers flexibility and efficiency in the teacher registration process, resulting in the more effective management of users within the system.

The "Course Registration" interface (Figure 16) makes it easier for the administrator to register subjects individually for the different careers offered by the Universidad del Noreste. This process involves assigning each subject a specific name, indicating the corresponding major, specifying the relevant academic period, and defining the hours associated with the subject. In addition to the individual entry option, the system also provides the administrator with the ability to perform race registration efficiently and expeditiously. To do this, a functionality has been implemented that allows you to load an Excel file, which makes it possible to massively enter careers. This simplified approach significantly streamlines career management, ensuring greater efficiency and accuracy in the academic administration process.

Figure 16: Registration of courses interface (Spanish version).

Figure 17: Registration of course-teacher interface (Spanish version).

The “Assign Courses” interface (Figure 17) provides the planner with the ability to assign subjects to teachers based on the degree and academic period selected. Additionally, the screen includes a table at the bottom that shows a list of teachers with previously assigned subjects, thus providing a clear and detailed view of existing assignments. This screen is an effective tool that facilitates the management and monitoring of course assignments to teachers in the context of the system.

The interface for generating schedules allows one to obtain a map of the current configuration of schedules for a defined career and school period (Figure 18). It loads the availability of each teacher who teaches in that school period and is able to accommodate each period in a cell of a rectangular map, marking a selection with a different color when the hours are chosen (Figure 19) in such a way that the shape is more visually attractive to the user, in addition to being able to correct the schedules in a drag-and-move style with the computer mouse.

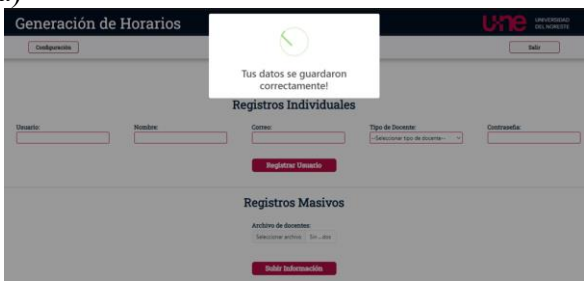
Figure 18: Timetable generation (Spanish version).

Figure 19: Selecting cells in the schedule (Spanish version).

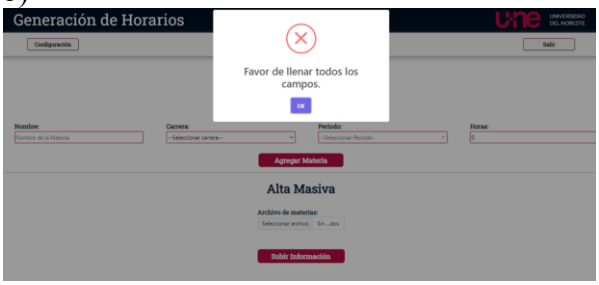
In different interfaces, the correct or incorrect completion of a task is confirmed. In case there is a problem, the user is given a report of what the problem was. Figure 20a shows a successful case for the user registration interface, and Figure 20b shows a failed case for the course registration interface; the latter cannot be done unless all the fields are selected or filled out correctly (period, major, subject, teacher, etc.)

Figure 20: Showing correct and incorrect procedure to the timetable generation (Spanish version).

a)



b)

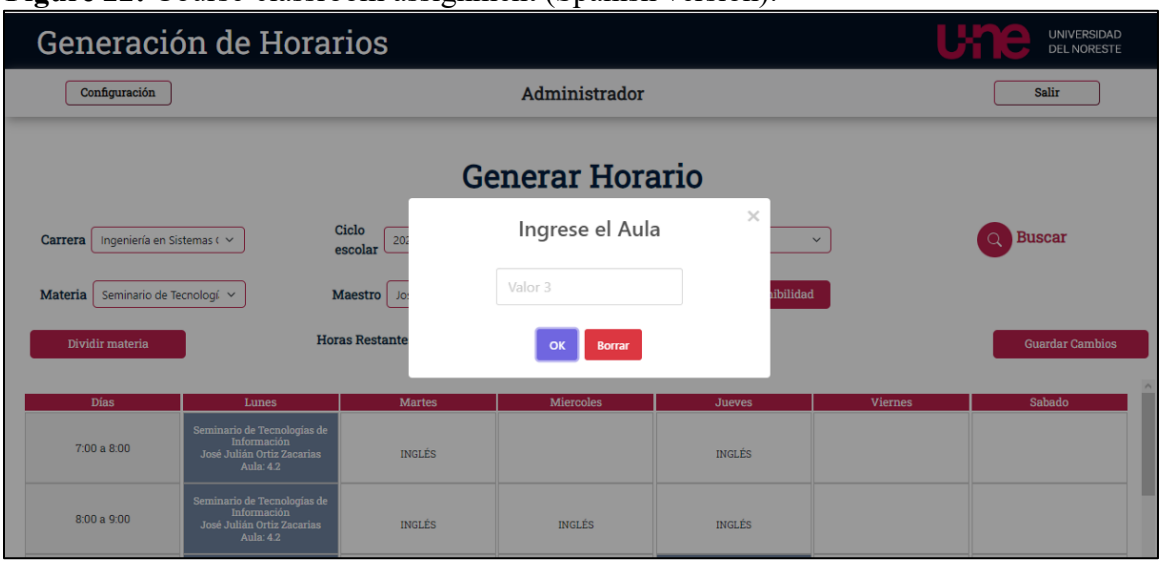


When planning a schedule for regular students, information on all courses for a school period is loaded. An irregular student is one who is behind the generation with which he/she entered the university because he/she failed one or more subjects. At this stage, generating your schedule is a little more complicated since you must load course by course those that the student can take and accommodate the schedule of the professors who can teach those courses (Figure 21).

Figure 21: Generation of irregular student schedule (Spanish version).

To optimize the classroom assignment process for scheduled classes, a functionality has been implemented that allows the administrator to enter the specific location by clicking on a previously selected cell. When activating this action, a window designed to facilitate the introduction of the necessary information (Figure 22) about the specific classroom where that course will be taught is displayed.

Figure 22: Course-classroom assignment (Spanish version).



An updated video showing the full functionality of the web system can be viewed on the YouTube platform (Horarios UNE 2023).

V. Conclusions

This project was undertaken to minimize the intensive manual effort that is being made to establish and build timetables for a private university in Tampico, Tamaulipas, Mexico. The web-based timetable system can produce intuitive and friendly manual designing of near-optimal schedules depending on several constraints. The timetable system allows multiple users access (administrators, planners, professors, and students) with a responsive design, provided there is an internet connected device. The login requirement secures the system from unauthorized users and modifications. With this approach, timetable management has been made simpler as a web application is used to generate and export the timetable in a PDF format that can be easily shared on internet enabled devices.

Our solution was developed using Google technology frameworks and libraries to simplify and boost the development process. Most of these well-established libraries and frameworks offer rich features and strong development benefits. This application successfully managed hard constraints to achieve a feasible and efficient timetable schedule for students and lecturers. Our system can be modified based on user needs and offers an attractive user interface with the best practices to design front-end and back-end web applications systems.

References

Babaei, H., J. Karimpour, and A. Hadidi. 2015. "A survey of approaches for university course timetabling problem." *Computers & Industrial Engineering* 86 (August): 43-59. https://doi.org/10.1016/j.cie.2014.11.010

Bashab, A., A.O. Ibrahim, E.E. AbedElgabar, M.A. Ismail, A. Elsafi, A. Ahmed, and A. Abraham. 2020. "A systematic mapping study on solving university timetabling problems using meta-heuristic algorithms." *Neural Computing & Applications* 32 (23): 17397–17432 (2020). https://doi.org/10.1007/s00521-020-05110-3

Burke, E.K., and S. Petrovic. 2002. "Recent research directions in automated timetabling." *European Journal of Operational Research* 140 (2): 266-280. https://doi.org/10.1016/S0377-2217(02)00069-3

Ceschia, S., L. Di Gaspero, and A. Schaerf. 2023. "Educational Timetabling: Problems, Benchmarks, and State-of-the-Art Results." *European Journal of Operational Research* 308 (1): 1-18. https://doi.org/10.1016/j.ejor.2022.07.011

Chen, M.C., S.N. Sze, S.L. Goh, N.R. Sabar, and G. Kendall. 2021. "A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities." *IEEE Access* 9: 106515-106529. doi: 10.1109/ACCESS.2021.3100613

Devin, F., and Y. Le Nir. 2010. "On-line timetabling software." In *Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling* (PATAT), Queen's University Belfast, Northern Ireland, August 10-13, 176-192.

Ernst, A.T., H. Jiang, M. Krishnamoorthy, and D. Sier. 2004. "Staff scheduling and rostering: A review of applications, methods and models." *European Journal of Operational Research* 153 (1): 3-27. https://doi.org/10.1016/S0377-2217(03)00095-X

Horarios UNE. 2023. "Sistema de Apoyo para la Generación de Horarios." YouTube, December 12. https://www.youtube.com/watch?v=yyb1tgJ0hw4

Kembuan, O., G.C. Rorimpandey, P.T.D. Rompas, and J.P.A. Runtuwene. 2018. "Development of Web based Timetabling System." In *Proceedings of the 7th Engineering International Conference on Education, Concept and Application on Green Technology*, Semarang, Central Java, Indonesia, October 18, 2018: 317-322. DOI: 10.5220/0009010403170322

Omoregbee, O. H., E.I. Ihama, L.E. Izogie, and F.M. Otamere. 2022. "Web-based Student Time-Table Management System." *Edo Poly Journal of Science, Technology and Management* 1 (1): 89–103. https://www.edopolyjournal.org.ng/index.php/home/article/view/15

Ong, B.C. 2020. "Web-based scheduling software for a university." Final Year Project, Universiti Tunku Abdul Rahman, Perak, Malaysia. http://eprints.utar.edu.my/id/eprint/3790

Phala, J.M. 1988. "A university course timetabling problem." *ORiON* 4 (2): 92-102. https://doi.org/10.5784/4-2-487

Priyadarshana, R.D.P.I. 2021. "Web Based Timetable Management System for University of Vocational Technology (UNIVOTEC)." Doctoral Dissertation. https://dl.ucsc.cmb.ac.lk/jspui/handle/123456789/4666

Raju, Y.R., and M. Mangal. 2017. "Web-Based Application for Automatic Timetable Generation." *International Journal of Interdisciplinary Innovative Research & Development* (*IJIIRD*) 02 (03): 23-28.

Schaerf, A. 1999. "A Survey of Automated Timetabling." *Artificial Intelligence Review* 13 (2): 87–127. https://doi.org/10.1023/A:1006576209967