ECORFAN Journal-Spain December 2024 Vol.11 No.21 8-20
Article

Development of a database for managing social service students using MATLAB

Desarrollo de una base de datos para gestión de alumnos de servicio social utilizando MATLAB

González-Galindo, Edgar Alfredo [*a], Luna-Alanís, Héctor Nathán[b], González-Ledesma, Alberto [c] and Castro-Pérez, Joseph Kevin[d]

[a] Universidad Nacional Autónoma de México • G-7927-2018 • 0000-0003-4654-9595 • 351785
[b] Universidad Nacional Autónoma de México • KZU-2476-2024 • 0009-0009-6326-2175 • 2053363
[c] Universidad Nacional Autónoma de México • KYR-9089-2024• 0009-0004-1719-7935 • 2051892
[c] Universidad Nacional Autónoma de México • HJY-8903-2023 • 0000-0001-6755-5260 • 1266037

CONAHCYT classification:

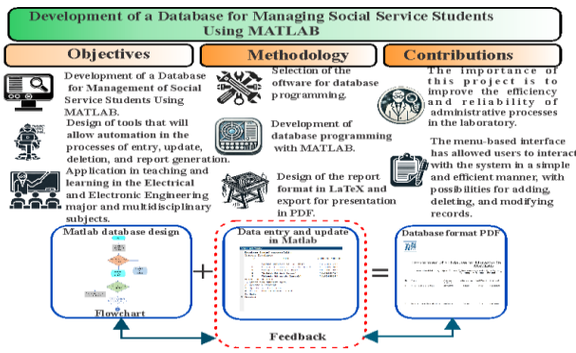
Area: Engineering
Field: Engineering
Discipline: System engineer
Subdiscipline: Computer Sciences

https://doi.org/10.35429/EJS.2024.21.11.8.20
History of the article:
Received: July 26, 2024
Accepted: December 07, 2024

* ✉ [unam_alf@comunidad.unam.mx]

Abstract

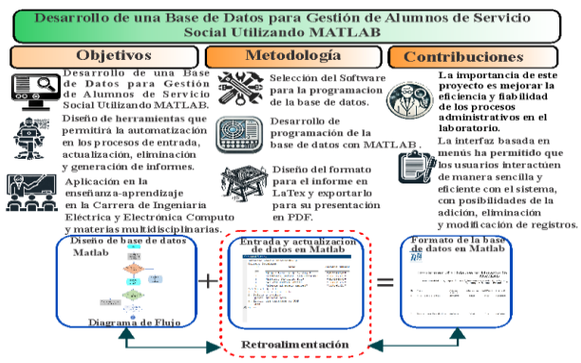
This work presents the development and implementation of a database for managing social service students using MATLAB. The system is designed to optimize and automate the processes of data entry, updating, deletion, and generation of student reports. The database includes essential fields such as student ID, name, account number, initial and final service periods, start and end dates, project title, and completion status. Users interact with the system through a menu-based interface in MATLAB, allowing easy addition, deletion, and modification of records. The system exports the database to a PDF report using MATLAB as a LaTeX editor, executing automatically. This solution reduces administrative workload in the laboratory, improving the efficiency and reliability of managing social service students.



Database, MATLAB, Latex

Resumen

Este trabajo presenta el desarrollo e implementación de una base de datos para la gestión de alumnos de servicio social utilizando MATLAB. El sistema está diseñado para optimizar y automatizar los procesos de entrada de datos, actualización, eliminación y generación de informes de los estudiantes. La base de datos incluye campos esenciales como ID del alumno, nombre, número de cuenta, periodos inicial y final del servicio, fechas de inicio y fin, título del proyecto y estatus de conclusión. Los usuarios interactúan con el sistema a través de una interfaz basada en menús en MATLAB, lo que permite la fácil adición, eliminación y modificación de registros. El sistema exporta la base de datos a un informe en PDF usando MATLAB como editor de LaTeX, ejecutándose automáticamente. Esta solución reduce la carga administrativa dentro del laboratorio, mejorando la eficiencia y fiabilidad en la gestión de alumnos de servicio social.



Base de Datos, MATLAB, Latex

Citation: González-Galindo, Edgar Alfredo, Luna-Alanís, Héctor Nathán, González-Ledesma, Alberto and Castro-Pérez, Joseph Kevin. Development of a database for managing social service students using MATLAB. ECORFAN Journal-Spain. 2024. 11-21:8-20.

Introduction

Efficient data management in educational settings is crucial for improving administration and monitoring of both academic and extracurricular activities. In the context of social service, it is essential to have systems that allow proper handling of student information, facilitating both administration and report generation, and storing it in the cloud for tracking student performance.

This project focuses on the development of a database for managing social service students, particularly in the Measurement, Instrumentation, and Control Laboratory at the Centro Tecnológico of the Facultad de Estudios Superiores Aragón of the Universidad Nacional Autónoma de México, using MATLAB as the primary tool. Known for its capabilities in numerical calculation and data visualization, MATLAB also offers functionalities for database management and manipulation, making it a viable option for such applications. According to Goranov (2023), advancements in technology have increased the complexity and size of data sets, necessitating programming tools for rapid data processing and multivariate statistical analysis in environmental research, with data processing capabilities, ease of use, and detailed tutorials.

This enhances research efficiency, and its free availability enables applications in UV-VIS spectroscopy, fluorescence spectroscopy, mass spectrometry, and nuclear magnetic resonance (NMR).

Another application is the Structural Health Monitoring Data Management System (SDMS) presented by Koo (2011), built on the MySQL database management system, an open-source relational database known for its ease of use and rapid deployment.

Koo developed a MATLAB-based interface using mYm that allows researchers to read and write data in the database using simple MATLAB commands, without needing advanced SQL knowledge.

They also developed a web interface with JavaScript and the Google Chart Tools API, allowing real-time data visualization in scatter plots.

This system has been implemented in various structures in the UK, including the Tamar Bridge and the Humber Bridge, demonstrating its ability to manage and analyze large volumes of structural data, providing valuable tools for maintenance and structural integrity research.

The article by MathWorks (2024), "MATLAB in the Cloud," describes how to run MATLAB and Simulink in cloud environments such as AWS (Amazon Web Services) and Azure, the latter being a cloud computing platform created by Microsoft for building, testing, deploying, and managing applications and services using its global infrastructure, allowing access to MATLAB directly from a web browser and facilitating the connection to data and computational resources for long-duration calculations, simulations, and data analysis without the need for powerful local hardware.

The "MATLAB Database Toolbox" resource by MathWorks (2024) explains how to use the MATLAB Database Toolbox to connect to SQL (Structured Query Language) and NoSQL (Not only Structured Query Language) relational databases and in the cloud, enabling import, export, and joining of tables without writing SQL queries.

This toolbox is compatible with popular databases like MySQL (My Structured Query Language), PostgreSQL, a relational database management system, and SQLite, the colloquial term 'Lite' referring to a 'reduced' version, providing native interfaces for greater efficiency. Additionally, the article "Connecting MATLAB to Cloud Databases" on MATLAB Central (MathWorks, 2024) explains how to connect MATLAB to Databricks and other cloud database services like Azure and Google BigQuery, the latter being a managed, serverless data storage product by Google that offers scalable analytics over large data volumes, enabling users to access and query large data sets remotely and run native MATLAB code on a Databricks cluster. Meanwhile, the article by Sandoval (2023) compiles the best articles on database design published in 2023, covering topics from entity-relationship diagrams (ERDs) to best practices for database schema design, providing essential information for professionals and enthusiasts in database design and management.

ISSN: 2444-3204

González-Galindo, Edgar Alfredo, Luna-Alanís, Héctor Nathán, González-Ledesma, Alberto and Castro-Pérez, Joseph Kevin. Development of a database for managing social service students using MATLAB. ECORFAN Journal-Spain. 2024. 11-21:8-20.
https://doi.org/10.35429/EJS.2024.21.11.8.20

On the other hand, the article "Database Management Trends in 2024" (Foote, 2023) explores emerging directions in database management, highlighting the use of cloud-based DBMS, autonomous and augmented databases, analytical, in-memory, and graph databases, and the integration of SQL and NoSQL through new databases like NewSQL, a type of database management system that combines the best of SQL and NoSQL databases, offering both transactional consistency and scalability. With the digitalization of today's society and the rise of IoT, people continuously interact with various user interfaces, which will be designed to be user-friendly, easy to use, and to provide the highest software product performance (Campos and Campos, 2023).

MATLAB is an interactive software for numerical calculation and data visualization, developed by MathWorks Inc. Widely used in fields of science and engineering, MATLAB is compatible with Unix, Macintosh, and Windows environments. The chapter covers how to create, edit, save, run, and debug M files (ASCII files with MATLAB instructions). Additionally, it explores the creation of matrices and vectors, and the use of linear algebra functions like matrix and vector multiplication, dot and cross products, transposition, determinants and inverses, and solving linear equations. MATLAB also allows for the programming of logical structures and loops, the use of subprograms and functions, and the generation of two and three-dimensional graphics. The symbolic math functions of MATLAB are also presented for performing symbolic operations and developing algebraic expressions. MATLAB is known for its highly optimized matrix and vector calculations and its intuitive language for expressing mathematical and visual problems and solutions, not forgetting that it can generate databases within the environment and visualize the information (Dukkipati, 2008).

MATLAB is a numerical computing environment and a scripting programming language. Numerical computing environments like MATLAB enable much more sophisticated and complex mathematical work than simple calculator programs or spreadsheet software packages can achieve. MATLAB stands for "matrix laboratory," and its ability to work with matrices was one of its original strengths, although it is now just one of many features.

Originally intended as a shortcut for students unfamiliar with FORTRAN, it now also interacts with programs written in C, C++, Java, and Python, and allows users to create their own user interfaces. MATLAB is integrated with Simulink, a graphical programming environment that can be driven by MATLAB scripts or used as the interface from which to run MATLAB. Simulink is used to simulate multi-domain dynamic systems. A multi-domain dynamic system is powered by more than one energy domain, including electrical, mechanical, hydraulic, pneumatic, and thermal. Therefore, designing a multi-domain dynamic system is a complex engineering task. Complementary software for Simulink includes software for generating C source code for real-time implementation, real-time testing of models, and formal verification and validation tools, along with graphical block construction libraries (Kte, 2023).

The term "informatics" arises from combining the words information and automatic. Informatics is the science that studies the automatic processing of information within a computer. Moreover, the term "information" is defined as a set of organized data that, collectively, conveys some meaning. Therefore, based on the previous definitions, an informatic system is an information system that enables automatic information processing and consists of a set of interconnected devices. Informatics is prevalent in society, both in personal and professional domains, and is one of the fastest evolving sectors. There are numerous informatic systems; thus, it is imperative that personnel managing these systems are qualified and specialized. Initially, informatic systems were located in a single space, but it is now common to find them distributed across various physical locations. For instance, in a company, not all computer devices are in the same place; several systems are distributed across different rooms or departments, as is the case with other computer devices that are part of an informatic system. This topic addresses the physical structure of an informatic system and its primary functions.

Programming languages have evolved significantly from their origins, as shown in the progression from their inception to the present: first-generation languages in the 1940s, machine code was the representative of this era, with the machine language ENIAC being a pioneer.

ISSN: 2444-3204

González-Galindo, Edgar Alfredo, Luna-Alanís, Héctor Nathán, González-Ledesma, Alberto and Castro-Pérez, Joseph Kevin. Development of a database for managing social service students using MATLAB. ECORFAN Journal-Spain. 2024. 11-21:8-20.
https://doi.org/10.35429/EJS.2024.21.11.8.20

They required encoding instructions using numerical codes. These languages were very difficult to use, and programming was highly complicated. Second-generation languages: In the early 1950s, the second evolution appeared; this was when assembler languages (specific to each machine) emerged, allowing mnemonic rules to be later transformed into machine code. Initially, this process was manual and was later performed by a program called an assembler.

Third-generation languages: In the 1950s, shortly after the appearance of assembler languages, the first compilers emerged, which added block structures and procedure calls, as well as control instructions in a compilable language. These first languages were FORTRAN, COBOL, and LISP. An algorithm is a sequence of necessary steps to solve a problem or perform a task within a finite time. Algorithms are independent of both the programming language in which they are expressed and the computer.

Algorithms can be translated into a programming language and then executed on a machine. The characteristics that an algorithm must have: Precise: it must indicate the order in which the tasks are performed. Defined: if an algorithm is executed twice, it should yield the same result each time. Finite: it must conclude at some point; it must have a finite number of steps (López et al., 2023). The resource "Save a figure as pdf - MATLAB Answers - MATLAB Central" explains how to use the exportgraphics command in MATLAB to create PDF files directly from MATLAB-generated figures. From version R2021b, it is possible to create PDF files containing multiple figures using a for loop (MathWorks, 2023). Furthermore, "PDF Latex file using Matlab - MATLAB Answers - MATLAB Central" describes how to export a LaTeX file to PDF using MATLAB. In the Live Editor, you can select Export to LaTeX, which also creates a LaTeX style file "matlab.sty" in the same folder as the output document (MathWorks, 2020). Another resource, "Convert live script or function to standard format - MATLAB export", provides details on how to convert live MATLAB scripts to standard formats such as PDF, Microsoft Word, and LaTeX. It describes additional options such as including outputs, running code, and hiding code during conversion (MathWorks, 2023).

Additionally, the script "matlabfrag to pdf - File Exchange - MATLAB Central" allows for creating PDF files formatted in LaTeX using matlabfrag to generate .eps and .tex files, then using pdflatex to create the PDF file (Martin, 2024).

Finally, the "Plot2LaTeX - File Exchange - MATLAB Central" tool exports a MATLAB figure as a vector-format PDF file for inclusion in LaTeX, requiring the open-source vector graphics editor Inkscape (Jan de Jong, 2024).

In today's world, the integration of advanced technologies in education is crucial for enhancing the quality and accessibility of learning across various academic contexts.

The incorporation of Big Data tools in business analytics courses, as discussed by Zadeh et al. (2021), highlights the need to develop competencies in handling massive data sets, preparing students to face challenges in data-driven decision-making.

Similarly, the use of open-source platforms like SageMath in teaching operational research in Brazil, presented by Assumpção et al. (2024), demonstrates how economic limitations can be overcome to provide quality education in engineering. Additionally, the application of backpropagation neural networks to personalize learning in open universities, as explored by Wang (2024), underscores the potential of machine learning to transform higher education. In parallel, the Evaluation, Supervision, and Control (ESC) strategies to mitigate student dropout in Islamic higher education, discussed by Agus (2024), reveal the importance of a proactive and collaborative approach to student retention.

Finally, the development of remotely accessible and fully controllable fiber optic systems laboratories, described by Aslan et al. (2024), shows how virtualization and remote control can expand educational opportunities, allowing students to conduct practical experiments without the constraints of time and space.

Collectively, these studies highlight how technological innovation can be a key catalyst in improving education across various fields and contexts.

ISSN: 2444-3204

González-Galindo, Edgar Alfredo, Luna-Alanís, Héctor Nathán, González-Ledesma, Alberto and Castro-Pérez, Joseph Kevin. Development of a database for managing social service students using MATLAB. ECORFAN Journal-Spain. 2024. 11-21:8-20.
https://doi.org/10.35429/EJS.2024.21.11.8.20

Objetive

Develop and implement a MATLAB-based database for efficient management and control of students undertaking their social service at the Measurement and Instrumentation Control Laboratory. This tool will enable the automation of data entry, updating, deletion, and report generation processes.

Hypothesis

Frequently, administrative tasks involve the submission of reports on students engaged in social service within the laboratory. Implementing a database in MATLAB for managing these social service students will facilitate the automation and optimization of control and administrative processes. Time optimization is pertinent as it enables the automatic generation of PDF reports, thereby minimizing the time spent on repetitive administrative tasks. This will significantly reduce the workload on staff and enhance the accuracy and reliability of the records.

Methodology and development

This work was developed at the Faculty of Higher Studies Aragón, within the Technological Center, specifically in the measurement and instrumentation laboratory. It is important to mention that having a database is crucial for maintaining an efficient record of students performing social service. This allows for the immediate presentation of administrative reports in PDF format when required. The file we need to create is named alumnos.mat. This extension describes a database stored in MATLAB format and generates a table where the students' data are stored. It must be loaded at the start of the program to maintain data persistence between sessions; if it does not exist, a new empty table would be created. The database columns include the following fields: ID (unique identifier of the student), Name (student's name), Account_Number (student's account number), Initial_Period (initial period of social service), Final_Period (final period of social service), Start_Date (start date of social service), End_Date (end date of social service), Project_Title (title of the student's project), and Concluded (status of social service completion: Concluded / Not concluded).

The file alumnos.tex is a LaTeX document generated by the program, containing the representation of the student table in LaTeX format and is used as input for the pdflatex compiler to generate a PDF file, including the header, footer, and the table format. The file alumnos.pdf is the result of compiling alumnos.tex using pdflatex, and it is the final version of the report in PDF format, created for viewing and distribution. Additionally, three auxiliary files are generated during the creation of the PDF file: alumnos.aux (an auxiliary file generated by pdflatex during compilation), alumnos.log (a log file containing messages and errors from the pdflatex compilation), and alumnos.out (an additional output file generated by pdflatex).

1.- Verification and loading of the database from the alumnos.mat file

Figure 1 shows a block diagram of the procedure for verifying the alumnos.mat file, which is the database, to ensure it loads correctly. If it does not exist, the file alumnos.mat will be created, and the following is the MATLAB code generated.

Box 1

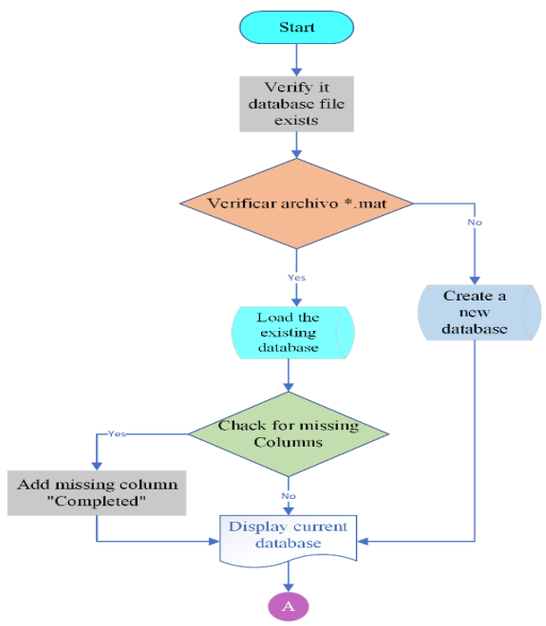


Figure 1

Load the database, and if it does not exist, create a new database file

MATLAB code where the student.map file is generated, and the table header is created.

ISSN: 2444-3204

González-Galindo, Edgar Alfredo, Luna-Alanís, Héctor Nathán, González-Ledesma, Alberto and Castro-Pérez, Joseph Kevin. Development of a database for managing social service students using MATLAB. ECORFAN Journal-Spain. 2024. 11-21:8-20.
https://doi.org/10.35429/EJS.2024.21.11.8.20

```
% Verify if the database file exists
if isfile('students.mat')
    % Load the existing database
    load('students.mat', 'students');
    disp('Database loaded successfully.');

    % Check if any column is missing and add it if necessary
    if ~ismember('Completed',
students.Properties.VariableNames)
        students.Completed = strings(height(students), 1);
    end
else
    % Create an empty table with the necessary fields
    students = table('Size', [0 9], 'VariableTypes', {'int32', 'string',
'string', 'string', 'string', 'string', 'string', 'string', 'string'}, ...
                     'VariableNames', {'ID', 'Name', 'Account_Number',
'Initial_Period', 'Final_Period', 'Start_Date', 'End_Date',
'Project_Title', 'Completed'});
    disp('New database created.');
end

% Display the database at the start
disp('Current Database:');
disp(students);
```

2.- Function and Operations

Figure 2 shows a flowchart presenting a menu to the user for performing various operations on the database. These operations include entering data, deleting data, updating data, exporting to PDF, and exiting.

Box 2

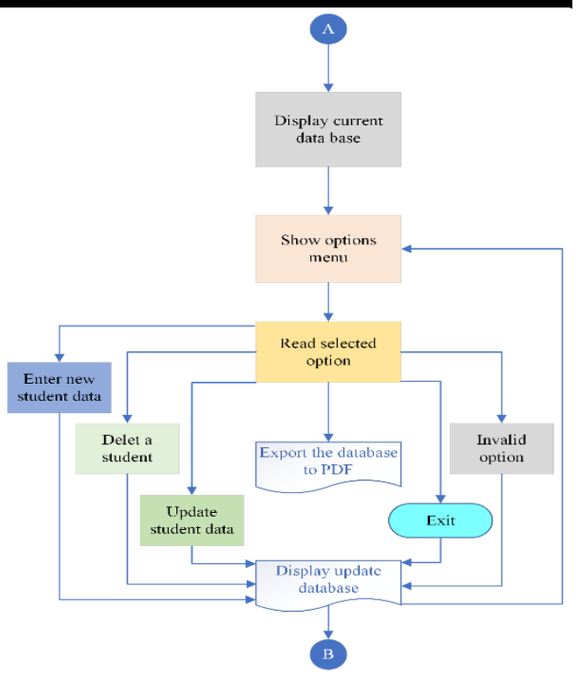


Figure 2

Flowchart visualizing the functions of the operators

MATLAB code displays the menu for selecting required actions, whether it is to input information, delete, update, export, or exit the iteration.

```
while true
    disp('Select an option:');
    disp('1. Enter new student data');
    disp('2. Delete a student');
    disp('3. Update student data');
    disp('4. Export the database to PDF');
    disp('5. Exit');
    option = input('Option: ', 's');
    switch option
        case '1'
            % Call the function to enter data
            students = enter_data(students);
            % Save the database
            save('students.mat', 'students');
        case '2'
            % Call the function to delete data
            students = delete_data(students);
            % Save the database
            save('students.mat', 'students');
        case '3'
            % Call the function to update data
            students = update_data(students);
            % Save the database
            save('students.mat', 'students');
        case '4'
            % Export the table to a LaTeX file
            export_to_latex(students, 'students.tex');
            % Compile the LaTeX file to PDF using pdflatex
            if isfile('students.tex')
                disp('Compiling the LaTeX file to PDF...');
                % Set an explicit time limit
                [status, cmdout] = system('pdflatex -
interaction=nonstopmode -halt-on-error students.tex');
                % Check if the compilation was successful
                if status == 0
                    disp('PDF generated successfully.');
                    disp('The PDF file is located in the following
directory:');
                    disp(pwd);
                    % Automatically open the PDF file
                    open_pdf('students.pdf');
                else
                    disp('Error generating the PDF.');
                    disp(cmdout);
                    % Display the contents of the log file
                    if isfile('students.log')
                        fid = fopen('students.log', 'r');
                        while ~feof(fid)
                            tline = fgetl(fid);
                            disp(tline);
                        end
                        fclose(fid);
                    end
                end
                % Delete intermediate files only if they exist
                delete_files({'students.aux', 'students.log',
'students.out', 'students.tex'});
            else
                disp('The file students.tex could not be created.');
            end
        case '5'
            break;
        otherwise
            disp('Invalid option. Please try again.');
    end
    % Display the database after each operation
    disp('Updated Database:');
    disp(students);
end
```

3.- Data Entry

Figure 3 displays a flowchart where a menu is presented to the user for entering data of the students.

ISSN: 2444-3204

González-Galindo, Edgar Alfredo, Luna-Alanís, Héctor Nathán, González-Ledesma, Alberto and Castro-Pérez, Joseph Kevin. Development of a database for managing social service students using MATLAB. ECORFAN Journal-Spain. 2024. 11-21:8-20.
https://doi.org/10.35429/EJS.2024.21.11.8.20

Box 3

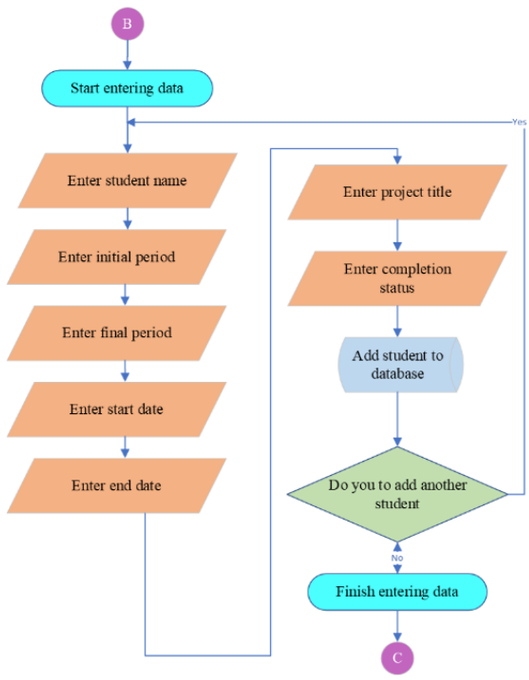


Figure 3

Flowchart enabling student data entry

The following code prompts the user to enter the data of a new student and adds it to the table. It allows entering multiple students in a single execution through iterations.

```
while true
      Name = input('Enter the student name: ', 's');
      Account_Number = input('Enter the account number (9 digits): ',
's');
      Initial_Period = input('Enter the initial period (e.g., 2024-1): ', 's');
      Final_Period = input('Enter the final period (e.g., 2024-2): ', 's');
      Start_Date = input('Enter the start date (dd-mm-yyyy): ', 's');
      End_Date = input('Enter the end date (dd-mm-yyyy): ', 's');
      Project_Title = input('Enter the project title: ', 's');
      Completed = input('Enter Yes if completed or No if not completed:
', 's');
      students = add_student(students, Name, Account_Number,
Initial_Period, Final_Period, Start_Date, End_Date, Project_Title,
Completed)
      continue_entry = input('Do you want to add another student? (y/n):
', 's');
      if lower(continue_entry) ~= 'y'
         break;
      end
   end
```

4.- Data Deletion

The user can delete data that has already been entered, removing the entire row, as the row contains all the information of the students (see Figure 4).

Box 4

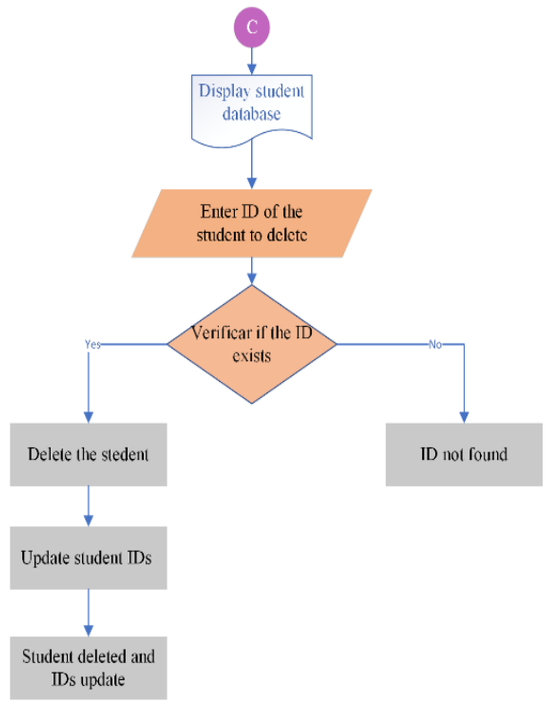


Figure 4

Flowchart facilitating the deletion of student data

The subsequent code allows the deletion of a student from the table based on the ID entered by the user and reorganizes the IDs to be consecutive following the deletion of a student.

```
disp(students);
   ID = input('Enter the ID of the student to delete: ');
   if any(students.ID == ID)
      students(students.ID == ID, :) = [];
      % Update IDs
      for i = 1:height(students)
         students.ID(i) = i;
      end
      disp('Student deleted and IDs updated.');
   else
      disp('ID not found.');
   end
```

5.-Updating Data:

The menu features a key option that allows modification of student information, presenting a menu to select the information requiring alterations as shown in Figure 5.

ISSN: 2444-3204

González-Galindo, Edgar Alfredo, Luna-Alanís, Héctor Nathán, González-Ledesma, Alberto and Castro-Pérez, Joseph Kevin. Development of a database for managing social service students using MATLAB. ECORFAN Journal-Spain. 2024. 11-21:8-20.
https://doi.org/10.35429/EJS.2024.21.11.8.20

Box 5

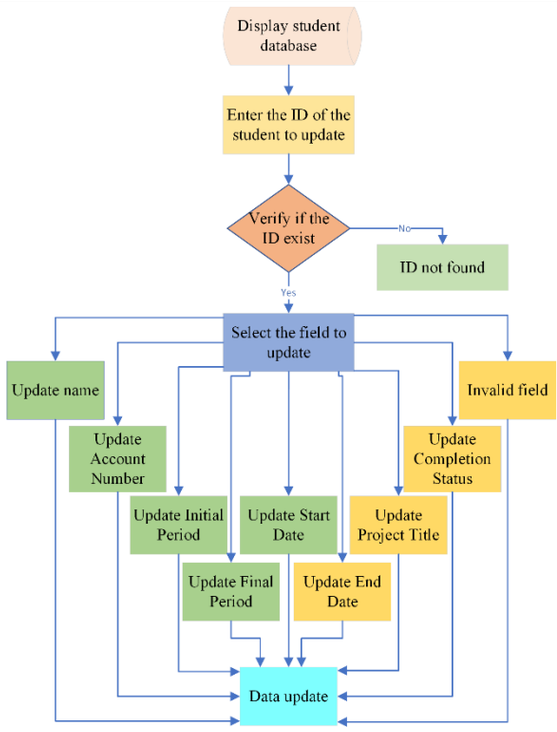


Figure 5

Flowchart for updating student data based on their ID

The code displayed below enables the updating of data for a specific student based on their ID.

```
disp(alumnos);
   disp(students);
   ID = input('Enter the ID of the student to update: ');
   if any(students.ID == ID)
      disp('Select the field to update:');
      disp('1. Name');
      disp('2. Account Number');
      disp('3. Initial Period');
      disp('4. Final Period');
      disp('5. Start Date');
      disp('6. End Date');
      disp('7. Project Title');
      disp('8. Completed');
      field = input('Field: ', 's');
      switch field
         case '1'
            students.Name(ID) = string(input('Enter the new name: ', 's'));
         case '2'
            students.Account_Number(ID) = string(input('Enter the new account number: ', 's'));
         case '3'
            students.Initial_Period(ID) = string(input('Enter the new initial period: ', 's'));
         case '4'
            students.Final_Period(ID) = string(input('Enter the new final period: ', 's'));
         case '5'
            students.Start_Date(ID) = string(input('Enter the new start date: ', 's'));
         case '6'
   students.End_Date(ID) = string(input('Enter the new end date: ', 's'));
         case '7'
            students.Project_Title(ID) = string(input('Enter the new project title: ', 's'));
         case '8'
            students.Completed(ID) = string(input('Enter Yes if completed or No if not completed: ', 's'));
         otherwise
            disp('Invalid field.');
      end
      disp('Data updated.');
   else
      disp('ID not found.');
   end
```

6.- Exporting to LaTeX and PDF:

This function creates a LaTeX (.tex) file from the alumnus table data, setting up the document class, used packages, and defining the page and table styles. It employs geometry to establish custom margins, fancyhdr to define headers and footers, datetime to include the date and time of update, setspace mainly used to adjust line spacing, and array and graphicx to define the table format and include graphics, respectively. The document title is defined, centered with a 1.5 line spacing, while the student table is formatted with centered columns using the previously defined C column type. Each table row is written using a fprintf within a for loop, open_pdf, this function opens the generated PDF file in the system's default PDF viewer, and delete_files, this function deletes intermediate files generated during LaTeX compilation to keep the directory clean. Figure 6 shows the flowchart of the process for exporting and viewing in PDF.

Box 6

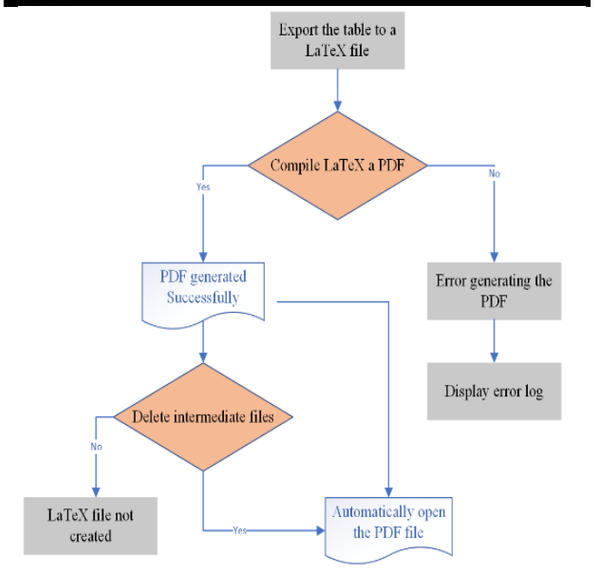


Figure 6

Flowchart for exporting the database to LaTeX and viewing in PDF

ISSN: 2444-3204

The following code demonstrates how the file can be exported to LaTeX and viewed in PDF.

```
    fid = fopen(filename, 'w');
    if fid == -1
        error('Could not create the LaTeX file.');
    end
    fprintf(fid, '\\documentclass{article}\n');
    fprintf(fid, '\\usepackage[landscape, left=0.5in, right=1in, top=1in, bottom=1in]{geometry}\n'); % Adjust margins
    fprintf(fid, '\\usepackage{booktabs}\n');
    fprintf(fid, '\\usepackage{fancyhdr}\n');
    fprintf(fid, '\\usepackage{datetime}\n'); % Package for date and time
    fprintf(fid, '\\usepackage{setspace}\n');
    fprintf(fid, '\\usepackage[english]{babel}\n');
    fprintf(fid, '\\usepackage{array}\n');
    fprintf(fid, '\\usepackage{graphicx}\n');
    fprintf(fid, '\\newcolumntype{P}[1]{>{\\raggedright\\arraybackslash}p{#1}}\n');
    fprintf(fid, '\\pagestyle{fancy}\n');
    fprintf(fid, '\\fancyhead[L]{\\includegraphics[width=2cm]{Aragon.jpg}}\n'); %Author: Edgar Alfredo Gonzalez Galindo
    fprintf(fid, '\\fancyhead[R]{FES Aragón Centro Tecnologico Aragón}\n');
    fprintf(fid, '\\fancyfoot[L]{\\fontsize{8}{12}\\textbf{Responsible: Edgar Alfredo Gonzalez Galindo}}\n');
    fprintf(fid, '\\fancyfoot[C]{\\thepage}\n');
    fprintf(fid, '\\fancyfoot[R]{Update date and time: \\today{} \\currenttime}\n'); % Date and time
    fprintf(fid, '\\renewcommand{\\headrulewidth}{0pt}\n');
    fprintf(fid, '\\renewcommand{\\footrulewidth}{0pt}\n');
    fprintf(fid, '\\setlength{\\headsep}{1.8cm}\n');
    fprintf(fid, '\\begin{document}\n');
    fprintf(fid, '\\begin{center}\n');
    fprintf(fid, '\\vspace*{1cm}\n'); % Space to move the title down
    fprintf(fid, '\\setstretch{1.5}\n');
    fprintf(fid, '\\textbf{\\LARGE Development of a Database for Managing Social Service Students Using MATLAB}\\\\[0.5cm]\n');
    fprintf(fid, '\\setstretch{-1}\n');
    fprintf(fid, '\\fontsize{11}{12}\\textbf{ González-Galindo, Edgar Alfredo *, Luna-Alanís, Héctor Nathán, González-Ledesma, Alberto and Castro-Pérez, Joseph Kevin}\\\\[0.5cm]\n');
    fprintf(fid, '\\setstretch{-1}\n');
    fprintf(fid, '\\end{center}\n');
    fprintf(fid, '\\begin{center}\n');
    fprintf(fid, '\\begin{tabular}{p{0.5cm}p{4.5cm}p{2cm}ccccp{4cm}p{2.7cm}}\n');
    fprintf(fid, '\\toprule\n');
    fprintf(fid, '\\fontsize{7}{12}\\textbf{ID} &
\\fontsize{7}{12}\\textbf{Name} &
\\fontsize{7}{12}\\textbf{Account Number} &
\\fontsize{7}{12}\\textbf{Initial Period} &
\\fontsize{7}{12}\\textbf{Final Period} &
\\fontsize{7}{12}\\textbf{Start Date} &
\\fontsize{7}{12}\\textbf{End Date} &
\\fontsize{7}{12}\\textbf{Project Title} &
\\fontsize{7}{12}\\textbf{Completed} \\\\\n');
    fprintf(fid, '\\midrule\n');
    fprintf(fid, '\\fontsize{7}{12}\\selectfont\n'); % Change the font size of the table
    for i = 1:height(students)
        fprintf(fid, '\\fontsize{7.5}{12}\\textbf{%d} &
\\fontsize{7.5}{12}\\textbf{%s} &
\\fontsize{7.5}{12}\\textbf{%s} &
\\fontsize{7.5}{12}\\textbf{%s} &
\\fontsize{7.5}{12}\\textbf{%s} &
\\fontsize{7.5}{12}\\textbf{%s} &
\\fontsize{7.5}{12}\\textbf{%s} &
\\fontsize{7.5}{12}\\textrm{%s} &
\\fontsize{7.5}{12}\\textbf{%s} \\\\\n',  students.ID(i), students.Name{i}, students.Account_Number{i}, students.Initial_Period{i}, students.Final_Period{i}, students.Start_Date{i}, students.End_Date{i}, students.Project_Title{i}, students.Completed{i});
        fprintf(fid, '\\hline\n'); % Add horizontal line after each row
        if mod(i, 4) == 0
            fprintf(fid, '\\end{tabular}\n');
            fprintf(fid, '\\newpage\n'); % Insert page break
            fprintf(fid, '\\begin{tabular}{p{0.5cm}p{4.5cm}p{2cm}ccccp{4cm}p{2.7cm}}\n'); % Reopen the table environment
            fprintf(fid, '\\toprule\n');
fprintf(fid, '\\fontsize{7}{12}\\textbf{ID} &
\\fontsize{7}{12}\\textbf{Name} &
\\fontsize{7}{12}\\textbf{Account Number} &
\\fontsize{7}{12}\\textbf{Initial Period} &
\\fontsize{7}{12}\\textbf{Final Period} &
\\fontsize{7}{12}\\textbf{Start Date} &
\\fontsize{7}{12}\\textbf{End Date} &
\\fontsize{7}{12}\\textbf{Project Title} &
\\fontsize{7}{12}\\textbf{Completed} \\\\\n');
            fprintf(fid, '\\midrule\n');
        end
    end
    fprintf(fid, '\\bottomrule\n');
    fprintf(fid, '\\end{tabular}\n');
    fprintf(fid, '\\end{center}\n');
    fprintf(fid, '\\end{document}\n');
    fclose(fid);
```

Results

Upon running the program for the first time, the existing database is successfully loaded, and the following output is displayed in the Command Window as shown in Figure 7.

Box 7

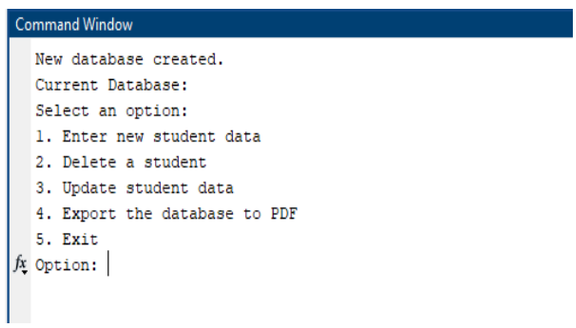


Figure 7

The Command Windows displays how a new database is created and requests the 5 options to execute; at the end, an iteration is generated

Figure 8 shows how the student data is entered, and if required, it restarts the menu to enter the next student.

ISSN: 2444-3204

Box 8

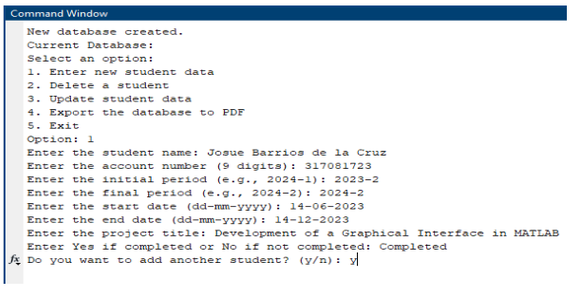


Figure 8

At the end of data entry, we can continue entering, giving the option of Y; if N is typed, it exits this iteration and returns to the initial menu.

Figure 9 shows the Command Windows window typing N to finish entering the student data.

Box 9

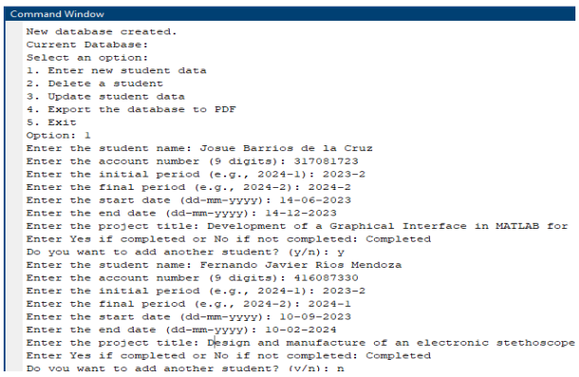


Figure 9

N is entered to end the iteration and return to the main menu.

Figure 10 shows the Command Windows waiting to select options, and it can be seen that a table of data for the students generated is displayed.

Box 10

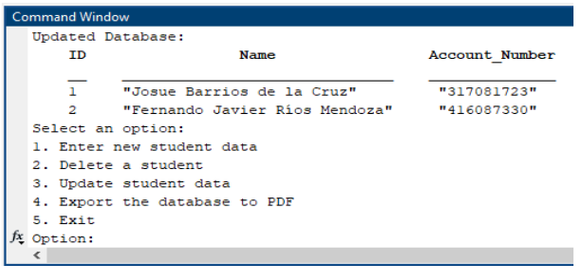


Figure 10

The MATLAB Command Windows shows the options for starting and exiting the database where two users are registered

Figure 11 shows the Command Windows as it looks before and after finishing entering the student data; it can continue and displays the list at the top and bottom with the updated data. This output indicates that the database has been updated correctly and presents the current state of the student records, including their identifiers, names, account numbers, service periods, start and end dates, titles of their works, and completion status.

Box 11

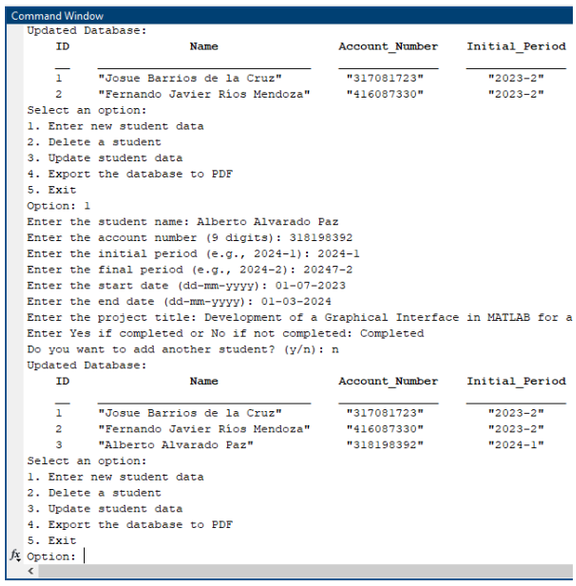


Figure 11

Displays the print in the Command Windows of the previous list and the updated list of student information.

Conclusions

The development and implementation of a database for managing social service students using MATLAB has proven to be an effective tool for improving the efficiency and reliability of administrative processes in the Measurement and Instrumentation Laboratory at the Facultad de Estudios Superiores Aragón. Through the automation of entry, update, deletion, and report generation processes, there has been a significant reduction in the workload of the staff assigned to the Laboratory. The system, by allowing the creation and management of detailed student records, facilitates the immediate generation of reports in PDF format, improving the precision and accessibility of the information.

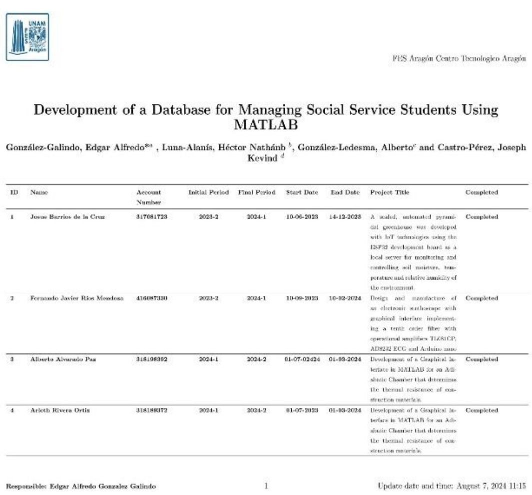
ISSN: 2444-3204

González-Galindo, Edgar Alfredo, Luna-Alanís, Héctor Nathán, González-Ledesma, Alberto and Castro-Pérez, Joseph Kevin. Development of a database for managing social service students using MATLAB. ECORFAN Journal-Spain. 2024. 11-21:8-20.
https://doi.org/10.35429/EJS.2024.21.11.8.20

The integration of MATLAB with LaTeX for generating PDF documents has proven to be a powerful combination, leveraging MATLAB's data management capabilities and the flexibility to export the format to LaTeX and view the file in PDF. Additionally, the MATLAB menu-based interface has allowed the responsible person to interact simply and efficiently with the system, enabling the addition, deletion, and modification of records without the need for advanced programming knowledge. This accessibility and ease of use are essential to ensure the adoption and continuous use of the system by the laboratory staff. In summary, the implementation of this database has significantly optimized the administrative processes within the laboratory related to managing social service students, providing a precise and efficient solution that can serve as a model for other laboratories and/or departments requiring similar management and reporting systems.

Annexes

Box 12



FES Aragón Centro Tecnologico Aragón

Development of a Database for Managing Social Service Students Using MATLAB

González-Galindo, Edgar Alfredo*a , Luna-Alanís, Héctor Nathánb b, González-Ledesma, Alberto c and Castro-Pérez, Joseph Kevind d

ID	Name	Account Number	Initial Period	Final Period	Start Date	End Date	Project Title	Completed
1	Josue Barrios de la Cruz	317081723	2023-2	2024-1	10-06-2023	14-12-2023	A scaled, automated pyramidal greenhouse was developed with IoT technologies using the ESP32 development board as a local server for monitoring and controlling soil moisture, temperature and relative humidity of the environment.	Completed
2	Fernando Javier Rios Mendoza	416087330	2023-2	2024-1	10-09-2023	10-02-2024	Design and manufacture of an electronic stethoscope with graphical interface implementing a tenth order filter with operational amplifiers TL081CP, AD8232 ECG and Arduino nano	Completed
3	Alberto Alvarado Paz	318198302	2024-1	2024-2	01-07-02424	01-03-2024	Development of a Graphical Interface in MATLAB for an Adiabatic Chamber that determines the thermal resistance of construction materials.	Completed
4	Arleth Rivera Ortiz	318188372	2024-1	2024-2	01-07-2023	01-03-2024	Development of a Graphical Interface in MATLAB for an Adiabatic Chamber that determines the thermal resistance of construction materials.	Completed

Responsible: Edgar Alfredo Gonzalez Galindo 1 Update date and time: August 7, 2024 11:15

Figure 12

PDF format as shown in the database record of social service students

Acknowledgements

The authors would like to thank the Coordination of the Technology Centre and the Electrical and Electronic Engineering Department of the Facultad de Estudios Superiores Aragón of the Universidad Nacional Autónoma de México, for the facilities in the use of tools provided through agreements with different software companies, We would like to thank Francisco Javier Domínguez Romero for his technical advice

Conflict of interest

The authors declare that they have no conflicts of interest. They have no known competing financial interests or personal relationships that might have appeared to influence the article reported in this paper.

Author contribution

González-Galindo, Edgar Alfredo: Collaborated in the initial conceptualization of the project, highlighting the need to integrate a menu-based interface in MATLAB to facilitate the entry, update, and deletion of student data. His contribution was crucial in defining the general structure of the system and ensuring that the database covered all essential fields such as student ID, name, and social service details, providing a solid foundation for the development and implementation of the system.

Luna-Alanís, Héctor Nathán: Played a key role in the technical part of the project, specifically in the development of the MATLAB code that generates and manages the database. He was responsible for implementing functions that allow dynamic interaction with the database, including adding and deleting records, as well as exporting these data to a PDF format through MATLAB using LaTeX, which significantly increased the efficiency of the administrative process.

González-Ledesma, Alberto: Focused on system optimization, working on process automation via MATLAB. His contributions included updating and consequently improving the database and integrating tools to generate automatic reports. These improvements ensured that the system was efficient and capable of minimizing time spent on repetitive tasks, a key goal of the project.

Castro-Pérez, Joseph Kevin: Contributed his expertise in user interface and end-user experience. He designed the MATLAB menu interface, ensuring it was intuitive and accessible to users without advanced programming knowledge. Moreover, he supervised the system integration with LaTeX for report creation, ensuring that the documents generated were both aesthetically pleasing and functionally complete, essential for the presentation of administrative reports.

Availability of data and materials

The data for this research is available according to the sources consulted.

Funding

Work carried out with the support of the UNAM-DGAPA-PAPIME Program, project PE102823

Abbreviations

NMR (Nuclear Magnetic Resonance)
SHM-SDMS (Structural Health Monitoring Data Management System)
MySQL (My Structured Query Language)
ERDs (Entity-Relationship Diagrams)
NoSQL (Not Only SQL)
SQL (Structured Query Language)
PostgreSQL (Relational Database Management System)
SQLite (Lightweight Structured Query Language)
ASCII (American Standard Code for Information Interchange)
ENIAC (Electronic Numerical Integrator and Computer)

References

Antecedents

Koo, K. Y., Battista, N. D., & Brownjohn, J. M. (2011, December). SHM data management system using MySQL database with MATLAB and web interfaces. In *5th international conference on structural health monitoring of intelligent infrastructure (shmii-5), cancún, méxico* (pp. 589-596).

Dukkipati, R. V. (2008). *Matlab: An introduction with applications*. New Age International Ltd.

López Querol, J., Campos Monge, E., & Campos Monge, M. (2023). *Algoritmia y Bases de Datos: Temario Oficial de Informática, Volumen II*. Rama Editorial.

Zadeh, A. H., Zolbanin, M., & Sharda, R. (2021). Incorporating Big Data tools for social media analytics in a business analytics course.

Assumpção, G. D. S., Santos, C. M. D., Campello, D. D. L., de Lima, L. S., & Castro, A. D. C. (2024). A proposal of teaching operational research in online contexts: An experience with SageMath in Brazil. *Engineering Reports*, e12863.

Wang, J. (2024). Towards an open university based on machine learning for the teaching service support system using backpropagation neural networks. *Soft Computing*, 28(5), 4531-4549.

Basics

MathWorks. (2024). MATLAB in the cloud - MATLAB & Simulink. MathWorks.

MathWorks. (2024). Database Toolbox - MATLAB. MathWorks.

MathWorks. (2024). Connecting MATLAB to cloud databases - MATLAB Answers. MathWorks.

MathWorks. (2023). Save a figure as pdf - MATLAB Answers - MATLAB Central.

MathWorks. (2020). PDF Latex file using Matlab - MATLAB Answers - MATLAB Central.

MathWorks. (2023). Convert live script or function to standard format - MATLAB export.

Martin. (2024). matlabfrag to pdf. MATLAB Central File Exchange.

Jan de Jong. (2024). Plot2LaTeX. MATLAB Central File Exchange.

Supports

Goranov, A. I., Sleighter, R. L., Yordanov, D. A., & Hatcher, P. G. (2023). TEnvR: MATLAB-based toolbox for environmental research. *Analytical Methods*, *15*(40), 5390-5400.

Sandoval, J. (2023, December 14). Year in review: Our best database design articles of 2023. Vertabelo.

Foote, K. D. (2023, December 19). Database management trends in 2024. DATAVERSITY.

ISSN: 2444-3204

González-Galindo, Edgar Alfredo, Luna-Alanís, Héctor Nathán, González-Ledesma, Alberto and Castro-Pérez, Joseph Kevin. Development of a database for managing social service students using MATLAB. ECORFAN Journal-Spain. 2024. 11-21:8-20.
https://doi.org/10.35429/EJS.2024.21.11.8.20

Kte, pi, B. M. (2023). MATLAB (matrix laboratory). *Salem Press Encyclopedia of Science*.

Discussions

Campos Monge, M., & Campos Monge, E. (2023). Lenguaje C, Bases de Datos y Aplicaciones Informáticas: Sistemas y aplicaciones informáticas (F.P.). Volumen III. Rama Editorial.

Agus Zaenul Fitri, Hosaini. (2024). Evaluation, Supervision, and Control (ESC) strategies in student dropout management in Islamic higher education. *Vol. 48 No. 1*, ISSN:1000-3673, pp.1589-607.

Aslan Akyol, Z., & Yucel, M. (2024). Remote accessible/fully controllable fiber optic systems laboratory design and implementation. *Computer Applications in Engineering Education*, 32(1), e22694.

ISSN: 2444-3204

González-Galindo, Edgar Alfredo, Luna-Alanís, Héctor Nathán, González-Ledesma, Alberto and Castro-Pérez, Joseph Kevin. Development of a database for managing social service students using MATLAB. ECORFAN Journal-Spain. 2024. 11-21:8-20.
https://doi.org/10.35429/EJS.2024.21.11.8.20