

University of Guadalajara

University Center of the Valleys



Software Configuration Management

**Academic-Administrative Management System for Postgraduate
Studies at the University Center of the Valleys (CUValles). Specific
case: Master's in Software Engineering**

Teacher: Omar Ali Zatarain Duran

Author: Elizabeth Baños Arias

Ameca, Jalisco, 2025

ÍNDICE DE CONTENIDOS

1. INTRODUCTION	1
2. JUSTIFICATION.....	1
3. FUNCIONAL REQUIREMENTS	1
4. MODULES.....	3
5. NON-FUNCTIONAL REQUIREMENTS	3
6. METHODOLOGY	4
7. LAYERED ARCHITECTURE:.....	4
8. PROJECT EXPENSE ESTIMATE (3 MONTHS).....	5

1. INTRODUCTION

The purpose of this project is to develop a web system for the academic and administrative management of the postgraduate program in Software Engineering at CUValles. This system wants to centralize and improve the processes related to the management of students, professors, and academic procedures. It will provide a technological platform that makes access, organization, and control of information easier. With this, the project looks to modernize the management tools and improve the experience of both administrative staff and academic users.

2. JUSTIFICATION

The implementation of this system is very important because academic and administrative management is currently done with Excel sheets. A stronger and more efficient solution will help to save time, avoid human errors, and assure the security and reliability of the data. In addition, this solution will support more efficient administration, better decision-making, and the improvement of the academic quality of the postgraduate program.

Objectives: Develop a web system for the academic and administrative management of the postgraduate program in Software Engineering at CUValles, for the management of students, professors, and academic procedures

3. FUNCIONAL REQUIREMENTS

NO.	Requirements	Description
RF_1	Authenticate User	Allows a user to log in to the system using valid credentials (username and password).
RF_2	Recover Password	Allows a user to recover or reset their password if they forget it, usually via email.
RF_3	Register User	Allows a new user to be registered in the system.
RF_4	Edit User	Allows an existing user to be modified.
RF_5	Delete User	Allows a user to be deleted from the system.
RF_6	Search User	Allows you to search for users in the system.
RF_7	View User	Allows you to view the details of a registered user

RF_8	Register Student	Allows you to register a new student in the system with personal and academic information.
RF_9	Edit Student	Allows you to modify the information of a registered student.
RF_10	Delete Student	Allows you to delete a student from the system.
RF_11	Search Student	Allows you to search for students
RF_12	View Student	Allows you to view a student's details without.
RF_13	Register Teacher	Allows you to register a new professor with their personal and professional information.
RF_14	Edit Teacher	Allows you to modify the information of an existing professor.
RF_15	Delete Teacher	Allows you to delete a professor from the system.
RF_16	Search Teacher	Allows you to search for professors.
RF_17	View Teacher	Allows you to view the details of a registered professor.
RF_18	Register Procedure	Allows you to register a new academic or administrative procedure, such as documents such as a Letter of Acceptance, Travel Permit, or Scholarship Commitment Letter.
RF_19	Edit Procedure	Allows you to edit an academic or administrative procedure, such as documents such as an Acceptance Letter, Travel Permit, or Scholarship Commitment Letter.
RF_20	Delete Procedure	Allows you to eliminate an academic or administrative procedure
RF_21	Search Procedure	Allows you to search for academic or administrative procedures such as an Acceptance Letter, Travel Permit, or Scholarship Commitment Letter.
RF_22	View Procedure	Allows you to view academic or administrative procedure details without.
RF_23	Export Procedure	Allows you to export documents

Table 1: FUNCIONAL REQUIREMENTS

4. MODULES

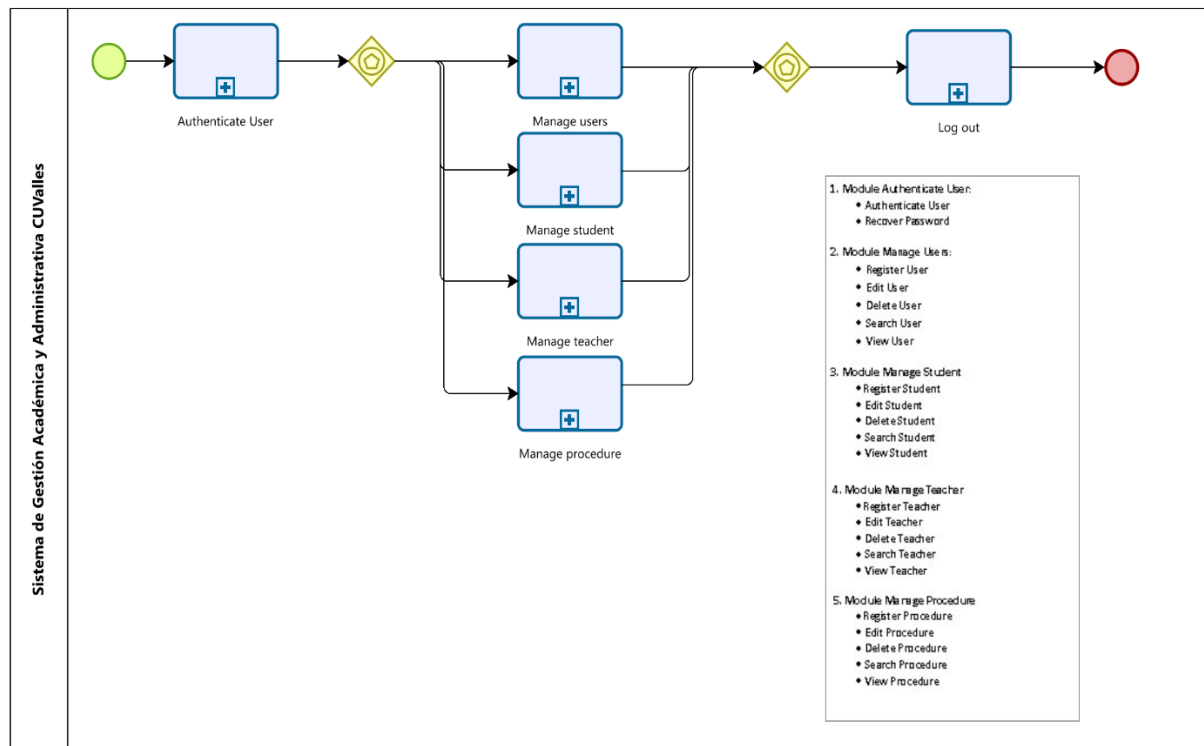


Figure 1: MODULES

5. NON-FUNCTIONAL REQUIREMENTS

1. Performance

- The system must authenticate users within a maximum of 2 seconds.

2. Usability

- The interface must be intuitive.
- Registration, editing, and deletion forms must have real-time validation (e.g., required fields).

3. Security

- Password recovery must be done via a secure email and a single-use token.

4. Compatibility

- The system must be accessible from modern browsers (Chrome, Firefox, Edge).
- It must be adaptable to different devices (PC, tablet, mobile).

6. METHODOLOGY

For this project, we use Scrumban because it combines the best of Scrum and Kanban. This methodology helps to improve the organization and the workflow of the team. With the visual board, it is easy to see all the tasks, their progress.

Another reason to use Scrumban is its flexibility. If the project needs changes, tasks can be adjusted without problems and without affecting the final product. Also, the meetings are simple and useful: planning for updates, daily meetings for coordination, sprint reviews, and retrospectives. These meetings are not too long, so they save time but still help the team to stay connected.

Scrumban is a good choice for this project because it allows better control, adaptation to changes, and good team communication, which are important to finish the system successfully.

7. LAYERED ARCHITECTURE:

1. Presentation Layer (Frontend)

- It is responsible for showing information and receiving data entered by the user.
- It communicates only with the business logic layer (backend).
- Example: forms for registering users, students, teachers, login.

2. Business Logic Layer (Backend / Application)

- This defines business rules: who can access, how data is validated, how permissions are managed.
It communicates with the database to read and write information.
- Once validated, it communicates with the data layer (database) to save or find information.
- Example: manage authentication and data validation.

3. Data Layer (Database)

- This is where system data.
- It answers the backend when it asks for data.
- Example: tables for users, students, teachers.

8. PROJECT EXPENSE ESTIMATE (3 MONTHS)

Category	Description	Unit Cost (MXN)	Quantity/Time	Subtotal (MXN)
Human Resources	Full Stack Developer	\$18,000/month	3 month	\$54,000
	Analyst	\$14,000/month	2 month	\$28,000
	Tester / QA	\$12,000/month	2 month	\$24,000
Infrastructure	Hosting + Domain (1 year)	\$2,500	1	\$2,500
	Cloud server (3 months)	\$1,200/month	3 month	\$3,600
Tools	IDEs / Support Software	\$2,000	1	\$2,000
Teams	Programmer's laptop	\$18,000	1	\$18,000
Operations	Internet (3 months)	\$600/month	3 month	\$1,800
	Energy / basic services	\$800/month	3 month	\$2,400
Training	Courses / workshops	\$3,000	1	\$3,000
Subtotal				\$139,300
Contingency (10%)				\$13,930
Project cost				\$153,230
Sale price				\$183,876 (20%)

Profit				\$30,646
---------------	--	--	--	-----------------

Table 2: PROJECT EXPENSE ESTIMATE