

**DESARROLLO DE UN APLICATIVO WEB PARA LA GESTIÓN DE TUTORÍAS DE
ESTUDIANTES DEL PROGRAMA DE INGENIERÍA DE SISTEMAS EN LA
UNIVERSIDAD POPULAR DEL CESAR, SECCIONAL AGUACHICA.**

CAMILO ANDRÉS ARANGO REYES

LUIS FERNANDO MEJÍA MARTÍNEZ

**UNIVERSIDAD POPULAR DEL CÉSAR
SECCIONAL AGUACHICA
FACULTAD DE INGENIERÍAS Y TECNOLOGÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS**

2025

**DESARROLLO DE UN APLICATIVO WEB PARA LA GESTIÓN DE TUTORÍAS DE
ESTUDIANTES DEL PROGRAMA DE INGENIERÍA DE SISTEMAS EN LA
UNIVERSIDAD POPULAR DEL CESAR, SECCIONAL AGUACHICA.**

**CAMILO ANDRÉS ARANGO REYES
LUIS FERNANDO MEJÍA MARTÍNEZ**

**LA EJECUCIÓN DEL PRESENTE PROYECTO DE INVESTIGACIÓN ES UN
REQUISITO PARA OBTENER EL TÍTULO COMO INGENIERO DE SISTEMAS.**

DIRECTOR(A)

Dr(c). LINA MARCELA ARÉVALO VERGEL

CODIRECTOR(A)

Dr(c). ERNEY ALBERTO RAMÍREZ CAMARGO

UNIVERSIDAD POPULAR DEL CÉSAR

SECCIONAL AGUACHICA

FACULTAD DE INGENIERÍAS Y TECNOLOGÍAS

PROGRAMA DE INGENIERÍA DE SISTEMAS

2025

Nota de aceptación:

LINA MARCELA ARÉVALO VERGEL

Director

ERNEY ALBERTO RAMÍREZ CAMARGO

Codirector

LIZETH BADILLO DURÁN

Jurado

CARLOS ALBERTO MEJIA RODRIGUEZ

Jurado

Aguachica, Día _____ Mes _____ Año _____

AGRADECIMIENTOS

Expresamos nuestro más sincero agradecimiento a la Universidad Popular del Cesar, Seccional Aguachica, por brindarnos la formación integral durante todo nuestro recorrido académico en el programa de Ingeniería de Sistemas, permitiéndonos alcanzar esta meta profesional.

A nuestros docentes de Ingeniería de Sistemas, quienes con su compromiso, conocimientos y disposición formaron parte esencial de nuestro proceso de formación, aportando significativamente al desarrollo de este trabajo.

Un reconocimiento especial a nuestra directora de proyecto, Lina Marcela Arévalo Vergel, por su apoyo, motivación y entrega, cualidades que fueron fundamentales para avanzar con determinación y culminar esta investigación.

De igual manera, agradecemos a nuestro codirector, Erney Alberto Ramírez Camargo, por su valiosa colaboración y acompañamiento durante este proceso.

A nuestras familias, por su amor, paciencia y respaldo incondicional durante cada etapa de nuestra carrera. Su apoyo constante ha sido nuestro impulso y razón para seguir adelante, con la esperanza y alegría de vernos alcanzar el título como Ingenieros de Sistemas.

Finalmente, extendemos nuestro agradecimiento a todas las personas que, de alguna manera, contribuyeron al desarrollo de esta investigación, ofreciendo su tiempo, recursos, conocimientos y atención, dejando una valiosa huella en este proceso académico.

CONTENIDO

1.	PROBLEMA	19
1.1	DEFINICION DEL PROBLEMA.....	19
1.2	JUSTIFICACIÓN	22
1.3	OBJETIVO GENERAL	23
1.3.1	Objetivo General	23
1.3.2	Objetivos específicos	23
1.4	DELIMITACIÓN.....	24
1.4.1	Temporal	24
1.4.2	Espacial.	25
1.4.3	Contextual.....	25
2	MARCO REFERENCIAL.....	28
2.1	MARCO HISTÓRICO.....	28
2.1.1	Antecedentes Internacionales	28
2.1.2	Antecedentes Nacionales.	31
2.1.3	Antecedentes Locales	33
2.2	MARCO TEÓRICO.....	34
2.3	MARCO LEGAL	38
2.3.1	Ley 23 de 1982	38
2.3.2	Ley 1581 de 2012	39
2.3.3	Ley 1915 de 2018.	41
2.3.4	Norma técnica colombiana 5854.....	43
2.3.5	Ley 1341 de 2009 (modificada por la Ley 1978 de 2019) – Ley TIC.....	44
2.4	MARCO CONCEPTUAL	45
3	DISEÑO METODOLÓGICO	50
3.1	TIPO Y ENFOQUE DE INVESTIGACIÓN.....	50
3.2	POBLACIÓN Y MUESTRA.	52
3.2.1	POBLACIÓN	52
3.2.2	MUESTRA	52
3.3	INSTRUMENTOS DE RECOLECCIÓN DE DATOS	54
3.4	HIPÓTESIS.....	55
3.4.1	Hipótesis de la investigación.....	55

3.5 VARIABLES	55
3.5.1 Variable independiente	55
3.5.2 Variable independiente	56
4 ESQUEMA TEMATICO	56
4.1 Fase 1: Obtener los requisitos funcionales y no funcionales para el desarrollo del aplicativo web para la gestión de tutorías.....	56
4.2 Fase 2: Diseñando las vistas del aplicativo web para la gestión de tutorías, en base a los requisitos previamente obtenidos.....	80
4.3 Fase 3: Construyendo el aplicativo web para la gestión de tutorías de los estudiantes del programa de ingeniería de sistemas en la Universidad Popular del Cesar, Seccional Aguachica.	92
4.3.1 Pasar wireframes a prototipo.....	92
4.3.2 Codificación del frontend y conexiones básicas con el Backend	103
4.3.3 sacar versión inicial del frontend con funcionalidad básica.....	129
4.3.4 Codificación del backend.....	147
4.4 Fase 4: Realizando pruebas de usabilidad y funcionalidad del aplicativo web para la gestión de tutorías, involucrando activamente a estudiantes, tutores y administradores del sistema.	170
4.4.1 Pruebas Unitarias.....	170
4.4.1.1 Prueba Automatizada: Creación de Cuentas Docente	170
4.4.1.2 Prueba Automatizada: Login del Docente	172
4.4.1.3 Prueba Automatizada: Login del Estudiante.....	174
4.4.1.4 Prueba Automatizada: Login del Administrador.....	176
4.4.1.5 Prueba Automatizada: Modificar datos y roles del usuario	178
4.4.1.6 Prueba Automatizada: Matricular estudiante a curso.	180
4.4.1.7 Prueba Automatizada: Eliminación de Estudiante del Curso por un Docente	182
4.4.1.8 Prueba Automatizada: Estudiante visualiza tutorías activas	184
4.4.1.9 Prueba Automatizada: El administrador activa o desactiva un curso	186
4.4.2.10 Prueba Automatizada: Estudiante visualiza tutorías activas matriculadas	188
4.4.2 Realizar pruebas de usabilidad.....	190
4.4.3 Análisis de resultados.	193
4.4.1 Realizar ajustes finales.	194
4.4.1.1 El colapsarlo de todos los cursos se pasó a una opción y página independiente.....	194
4.4.1.2 Se mejoraron los estilos del login	196

	4.4.1.3 Mejoras en los estilos de las tarjetas de los cursos (Administrador)	199
5	CONCLUSIONES.....	203
6	RECOMENDACIONES	205
7	BIBLIOGRAFIA	206
8	ANEXOS	211

LISTA DE FIGURAS

Figura 1: LoginGeneral	75
Figura 2: RegistroSistema	75
Figura 3: ActivarCuentaUsuario.....	75
Figura 4: ModificarDatosUsuario	76
Figura 5: ActivarDesactivarCurso	77
Figura 6: VisualizarUsuarios.....	77
Figura 7: AgregarEstudianteCurso	78
Figura 8: EliminarEstudianteCurso	78
Figura 9: VisualizarEstudiantes	79
Figura 10: VisualizarCursosActivos	79
Figura 11: VisualizarCursosMatriculados	80

LISTA DE TABLAS

Tabla 2: Requisitos no funcionales	59
Tabla 3: RF01 LoginGeneral.....	60
Tabla 4: RF02 RegistroSistema.....	62
Tabla 5: RF03 ActivarCuentaUsuario.....	63
Tabla 6: RF04 ModificarDatosUsuario.....	64
Tabla 7: RF05 ActivarDesactivarCurso.....	65
Tabla 8: RF06 VisualizarUsuarios.....	67
Tabla 9: RF07 AgregarEstudianteCurso.....	68
Tabla 10: RF08 EliminarEstudianteCurso.....	69
Tabla 11: RF09 VisualizarEstudiantes.....	70
Tabla 12: RF10 VisualizarCursosActivos.....	72
Tabla 13: RF11 VisualizarCursosMatriculados.....	73
Tabla 14: Roles y Redirecciones	107
Tabla 15: Roles y Estado del Usuario.....	113
Tabla 16: Campos del Curso	123

LISTA DE ILUSTRACIONES

Ilustración 1: Reportes estadísticos de deserción y reprobación generados por PEBI.	23
Ilustración 2: Planos geográficos del municipio de Aguachica - Cesar.....	27
Ilustración 3: Tabla de asistencia de estudiantes a las tutorías en el periodo 2023-2	53
Ilustración 4: Vista Previa – ÍndexHome.....	81
Ilustración 5: Vista Previa – Login	82
Ilustración 6: Vista Previa – Índex Admin	83
Ilustración 7: Vista Previa – Add User	83
Ilustración 8: Vista Previa – Edit User	84
Ilustración 9: Vista Previa – Gestión User	85
Ilustración 10: Vista Previa – Gestión Cursos.....	85
Ilustración 11: Vista Previa – Gestión Cursos / Cursos individuales.....	86
Ilustración 12: Vista Previa – Índex Estudiante.....	87
Ilustración 13: Vista Previa – Cursos (Vista de estudiante).....	88
Ilustración 14: Vista Previa – Cursos Individuales (Vista de estudiante)	89
Ilustración 15: Vista Previa – Índex Docentes	90
Ilustración 16: Vista Previa – Cursos individuales de asignaturas (Docente)	91
Ilustración 17: Login	99
Ilustración 18: Índex	100
Ilustración 19: Agregar usuarios	102
Ilustración 20: Gestión de cursos	102

Ilustración 21: Matricular usuarios	103
Ilustración 22: Código del formulario del login en el frontend.	105
Ilustración 23: Lógica de Login en el Backend	106
Ilustración 24: Código del formulario para Agregar Usuario	109
Ilustración 25: Código del controlador para Agregar Usuario	111
Ilustración 26: Código del formulario para Editar Usuario	115
Ilustración 27: Código del controlador para editar usuario	117
Ilustración 28: Código del formulario para crear curso	121
Ilustración 29: Código del controlador para crear curso	122
Ilustración 30: Código de la pantalla “Gestión de cursos”	125
Ilustración 31: Código de la lógica de la consulta de cursos.	126
Ilustración 32: Captura de la pantalla “índex”	129
Ilustración 33: Captura de la pantalla “login”.	131
Ilustración 34: Captura de la pantalla “vista principal admin”.	132
Ilustración 35: Título: Captura de la pantalla “agregar usuario”.	133
Ilustración 36: Captura de la pantalla “Modificar usuario”	134
Ilustración 37: Captura de la pantalla “Gestión de usuarios”	136
Ilustración 38: Captura de la pantalla “gestión de cursos”	139
Ilustración 39: Captura de la pantalla “gestión de un curso en específico”	141
Ilustración 40: Captura de la pantalla “índex estudiante”	143
Ilustración 41: Captura de la pantalla “cursos de un estudiante”	143
Ilustración 42: Captura de la pantalla “Detalle del curso”.	145
Ilustración 43: Captura de la pantalla “índex docente”.	146
Ilustración 44: Captura de la pantalla “gestión de un curso del docente”.	146

Ilustración 45: Código de la lógica del login.....	148
Ilustración 46: Código de la lógica de listar usuarios.....	150
Ilustración 47: Código de la lógica de “agregar usuario”.....	152
Ilustración 48: Código de la lógica de “editar usuario”.....	154
Ilustración 49: Código de la lógica de “agregar usuario”.....	155
Ilustración 50: Código de la lógica de “listar cursos”.....	157
Ilustración 51: Código de la lógica de” listar estudiantes matriculados a un curso”.	158
Ilustración 52: Código de la lógica de “matricular estudiante a un curso”.....	160
Ilustración 53: Código de la lógica de “Mostrar docente (sí hay) asignado a un curso”.	161
Ilustración 54: Código de la lógica de "eliminar docente de un curso"	163
Ilustración 55: Código de la lógica de "asignar docente a un curso"	165
Ilustración 56: Código de la lógica "visualizar los cursos en los que está matriculado un estudiante"	169
Ilustración 57: captura del test “Creación de Cuentas Docente”.....	172
Ilustración 58: captura del test “login del docente”.....	174
Ilustración 59: captura del test “login del estudiante”.....	176
Ilustración 60: captura del test “login del administrador”.....	177
Ilustración 61: Captura del test “Modificar datos y roles del usuario”.....	180
Ilustración 62: Captura del test “Matricular estudiante a curso”.....	182
Ilustración 63: Captura del test “Eliminación de Estudiante del Curso por un Docente”	184
Ilustración 64: Captura del test “Título: captura del test”.....	186

Ilustración 65: Captura del test “El administrador activa o desactiva un curso”.	188
Ilustración 66: Captura del test “Estudiante visualiza tutorías activas matriculadas”.	
.....	190
Ilustración 67: Prueba de usabilidad docente.	191
Ilustración 68: Prueba de usabilidad estudiantes.	191
Ilustración 69: Dirección de Programa.	192
Ilustración 70: Captura de la pantalla a modificar en el sidebar estudiante.	195
Ilustración 71: Captura de la pantalla del sidebar de estudiante modificado ...	196
Ilustración 72: Captura de la pantalla del “login” antes de modificar	198
Ilustración 73: Capturas de la pantalla del “login” modificado y mejorado en estilos	
.....	199
Ilustración 74: Captura de las tarjetas de los cursos (Administrador) antes de ajustes.	
.....	202
Ilustración 75: Captura de las tarjetas de los cursos (Administrador) después de ajustes.	
.....	202

LISTA DE ANEXOS

Anexo 1: Cronograma de actividades.....	211
Anexo 2: Fotos entrevista a representantes, docentes de tutorías.....	212
Anexo 3: Entrevista a docentes representantes.	213

RESUMEN.

El presente proyecto consistió en el desarrollo de un aplicativo web orientado a gestionar las tutorías académicas en el programa de Ingeniería de Sistemas de la Universidad Popular del Cesar, Seccional Aguachica. Este sistema fue creado con el propósito de optimizar la organización, planificación y seguimiento de las tutorías, facilitando el acceso de estudiantes y docentes a un entorno eficiente y tecnológicamente actualizado.

Para su desarrollo, se identificaron y analizaron los requisitos funcionales y no funcionales del sistema mediante entrevistas estructuradas, observación directa y análisis documental. Posteriormente, se diseñaron las interfaces del aplicativo considerando criterios de usabilidad y accesibilidad. El sistema se construyó e implementó utilizando herramientas modernas de desarrollo web, lo que permitió garantizar un desempeño óptimo en diferentes plataformas. Durante el proceso, se realizaron pruebas de funcionalidad y usabilidad involucrando a estudiantes, tutores y administradores, lo que permitió identificar y corregir errores, asegurando la calidad del producto final.

El aplicativo permite realizar funciones como la programación de tutorías, el registro de estudiantes y docentes, la asignación de horarios, y la gestión de reportes académicos. Como resultado, se logró reducir los problemas relacionados con cruces de horarios, interrupciones y dificultades logísticas que anteriormente complicaban las tutorías. Además, se observó una mejora en la comunicación entre tutores y estudiantes, así como un incremento en la participación en las tutorías, lo que contribuyó a abordar el problema

de los altos índices de reprobación en asignaturas críticas como Programación II y Cálculo II.

El proyecto alcanzó con éxito su objetivo principal al desarrollar e implementar un aplicativo web que mejora significativamente la gestión de las tutorías académicas. Este sistema no solo resuelve los problemas logísticos previamente existentes, sino que también refuerza la calidad de los procesos educativos, beneficiando tanto a estudiantes como a docentes. Con un mantenimiento periódico y capacitación continua de los usuarios, este aplicativo tiene el potencial de convertirse en una herramienta fundamental para optimizar el rendimiento académico y reducir la deserción estudiantil en la universidad.

ABSTRACT

This project focused on the development of a web application aimed at managing academic tutoring in the Systems Engineering program at the Universidad Popular del Cesar, Aguachica branch. The system was designed to optimize the organization, planning, and monitoring of tutoring sessions, providing students and tutors with an efficient and technologically updated platform.

To achieve this, the functional and non-functional requirements of the system were identified and analyzed through structured interviews, direct observation, and document analysis. Subsequently, the application's interfaces were designed with usability and accessibility in mind. The system was developed and implemented using modern web development tools, ensuring optimal performance across various platforms. Functionality and usability tests were conducted with the participation of students, tutors, and administrators, which allowed for error correction and ensured the final product's quality.

The application includes features such as tutoring scheduling, student and tutor registration, timetable management, and academic report generation. The implementation successfully addressed issues like scheduling conflicts, interruptions, and logistical difficulties that previously hindered tutoring sessions. Moreover, the system improved communication between tutors and students and increased participation in tutoring, helping to mitigate high failure rates in critical subjects such as Programming II and Calculus II.

The project successfully achieved its main objective by developing and implementing a web application that significantly enhances the management of academic tutoring. This system not only resolves pre-existing logistical challenges but also strengthens the quality

of educational processes, benefiting both students and tutors. With periodic maintenance and continuous user training, the application has the potential to become a fundamental tool for improving academic performance and reducing student dropout rates at the university.

INTRODUCCIÓN

El desarrollo tecnológico ha transformado diversos sectores sociales, siendo la educación uno de los más impactados. En este contexto, las instituciones de educación superior enfrentan el desafío de integrar soluciones innovadoras que permitan optimizar sus procesos académicos y administrativos. La Universidad Popular del Cesar, Seccional Aguachica, no es ajena a estas necesidades, especialmente en el programa de Ingeniería de Sistemas, donde se evidencian altas tasas de reprobación y deserción en asignaturas de línea ubicada en el pensum académico a la ciencias básicas y ingeniería aplicada críticas como Programación II y Cálculo II, según reportes del Programa de Permanencia y Bienestar Institucional (PEBI), durante el periodo 2023-2, Programación II registró 21 estudiantes reprobados entre los semestres segundo y octavo, con un mayor número en quinto semestre (12 estudiantes). De igual manera, en Cálculo II se reportaron 8 reprobados, siendo también el quinto semestre el más afectado.

La dificultad principal se encuentra en la gestión manual y desorganizada de las tutorías académicas, lo que genera conflictos de horarios, falta de seguimiento y limitaciones en la accesibilidad para los estudiantes que más lo necesitan. Ante esta situación, surge la necesidad de desarrollar un aplicativo web que permita gestionar de manera eficiente las tutorías, garantizando la organización, accesibilidad y seguimiento, y promoviendo una interacción más efectiva entre estudiantes y tutores.

El presente proyecto tiene como objetivo principal diseñar y construir un sistema que no solo facilite la programación de tutorías, sino que también automatice procesos clave,

optimizando así la experiencia académica de los estudiantes y fortaleciendo el apoyo pedagógico ofrecido por la institución. Este desarrollo se enmarca en el compromiso de la universidad por brindar herramientas tecnológicas que contribuyan al mejoramiento de los indicadores de permanencia y calidad educativa. A través de una metodología estructurada en fases, que incluye desde el análisis de requisitos funcionales hasta la implementación de la solución, este proyecto no solo busca responder a las necesidades identificadas, sino también aportar una herramienta replicable que pueda servir de modelo para otras instituciones educativas. En este sentido, el desarrollo de este aplicativo no solo responde a un problema local, sino que también se inserta en el contexto más amplio de las tendencias actuales en innovación tecnológica aplicada a la educación.

1. PROBLEMA

1.1 DEFINICION DEL PROBLEMA

Tal y como señala (Colcha Toscano, 2023) de la Universidad Politécnica Salesiana Ecuador en su tesis de grado titulada “Desarrollo de una plataforma web para el registro de tutorías de la Carrera de Ciencias de la Computación de la Universidad Politécnica Salesiana”; En la actualidad las universidades dentro de sus carreras ofertadas requieren y necesitan una solución tecnológica que automatice y optimice de una manera eficiente las tutorías académicas de las materias que vean con mayor índice de reprobación, materias que son requeridas por los mismos estudiantes de estas carreras. Este sistema de tutorías académicas, Como indica (Analuisa Medina, 2019), en su proyecto de investigación para obtención de título, al ser aún un proceso no automatizado y optimizado tecnológicamente se llegan a generar problemas de incomodidad por los estudiantes al encontrarse con gran pérdida de tiempo de estos espacios, por interrupciones de otras personas o estudiantes no propios de las tutorías o en el caso más frecuente y común, los cruces de horario al efectuarse estos apoyos académicos dentro del mismo horario de otras actividades académicas u otras clases. Como afirman de manera similar (Escobar, Isasi Olmedo, & Ayala, 2022) en su trabajo final de grado titulado “Aplicación web para gestión de solicitudes de agendamiento de tutoría para la FP-UNE”.

Por otra parte, como dice (Villa & Atehortua, 2023) en su informe final de práctica titulado “Aplicación Web De Tutorías Académicas”, la reprobación académica es algo común, es un problema real y frecuente desde la educación primaria, seguido por la secundaria y

observada aún en la educación superior dejando como huella una consecuencia negativa en el desarrollo académico y personal de los estudiantes, ya que, a pesar del esfuerzo de las instituciones educativas por reducir estos altos índices de reprobación, el problema sigue siendo notable, por lo tanto la integración de tutorías académicas es una de las estrategias más comunes para prevenir y corregir los factores que influyen en la reprobación académica. Seguidamente (Padilla & Mayoral, 2017) en su artículo de investigación denominado “Las tutorías académicas en el fortalecimiento del álgebra en estudiantes de octavo grado en una escuela distrital de Barranquilla”, hace referencia a que no solo la problemática radica en las indicadas anteriormente, sino que, encontramos influyente las diversas quejas de alumnos con respecto a las metodologías de enseñanza usada por los docentes de ciertas asignaturas llevando esto a la falta de confianza docente-alumno, generando no solo problemas específicamente académicos sino también emocionales como lo son una baja autoestima. Asimismo como lo expresa (Barragán & Barrera, 2020), la recomendación por parte del MEN (Ministerio de Educación Nacional) es que las estrategias para favorecer la permanencia sean implementados, en este caso las tutorías, sean implementadas con motivos preventivos, de una manera que impacten estos altos índices de deserción y reprobación, mejorando así la calidad de los procesos educativos; Dándole así una particular importancia en las universidades a esos estudiantes en situaciones de riesgo académicos, sean ya repitentes, desertores, bajo rendimiento y/o propiamente reprobados, para que sean ellos quienes con mayor urgencia acudan a estas tutorías académicas.

A pesar de los múltiples esfuerzos institucionales dentro de la universidad popular del cesar seccional Aguachica, el programa de ingeniería de sistemas y de la existencia de

tutorías académicas de manera presencial como estrategia para reducir la reprobación, las soluciones actuales evidencian claras deficiencias operativas y tecnológicas. Como se ha señalado, el proceso continúa siendo manual, propenso a interrupciones, pérdidas de tiempo y conflictos de horarios, lo que lo hace ineficiente e incluso desmotivador tanto para estudiantes como para docentes. Estas limitaciones no solo afectan el aprovechamiento académico, sino que también inciden negativamente en la salud emocional del estudiante, al fomentar sentimientos de frustración, baja autoestima y falta de confianza en el acompañamiento docente.

En este contexto, la implementación de un aplicativo web para la gestión de tutorías de la Universidad Popular del Cesar, Seccional Aguachica, en el programa de Ingeniería De Sistemas no es un lujo, sino una necesidad urgente. Esta aplicación se convierte en un aliado imprescindible al permitir una programación clara, oportuna y personalizada de las tutorías, garantizando así un acceso más equitativo y eficiente al acompañamiento académico, donde no solo el alumno podrá asistir a presencial, sino que encontrará un apoyo más de manera virtual. Además, su integración permite responder de forma proactiva a las recomendaciones del MEN (Ministerio de Educación Nacional) sobre estrategias de permanencia, enfocándose de manera prioritaria en estudiantes en riesgo académico.

Así mismo se convierte en una herramienta clave para transformar las tutorías en una experiencia efectiva, organizada y alineada con los objetivos institucionales de calidad y permanencia educativa

1.2 JUSTIFICACIÓN

En la actualidad, se reconoce que la sociedad contemporánea se desenvuelve en un entorno marcado por un ritmo vertiginoso de cambios, particularmente impulsados por avances tecnológicos que ocurren de forma exponencial. Estos cambios tienen repercusiones significativas en la estructura misma de la sociedad, lo que a su vez influye en la evolución de los sistemas educativos tanto en el ámbito o rol de docente como en el rol de estudiantes, necesitando tutorías.

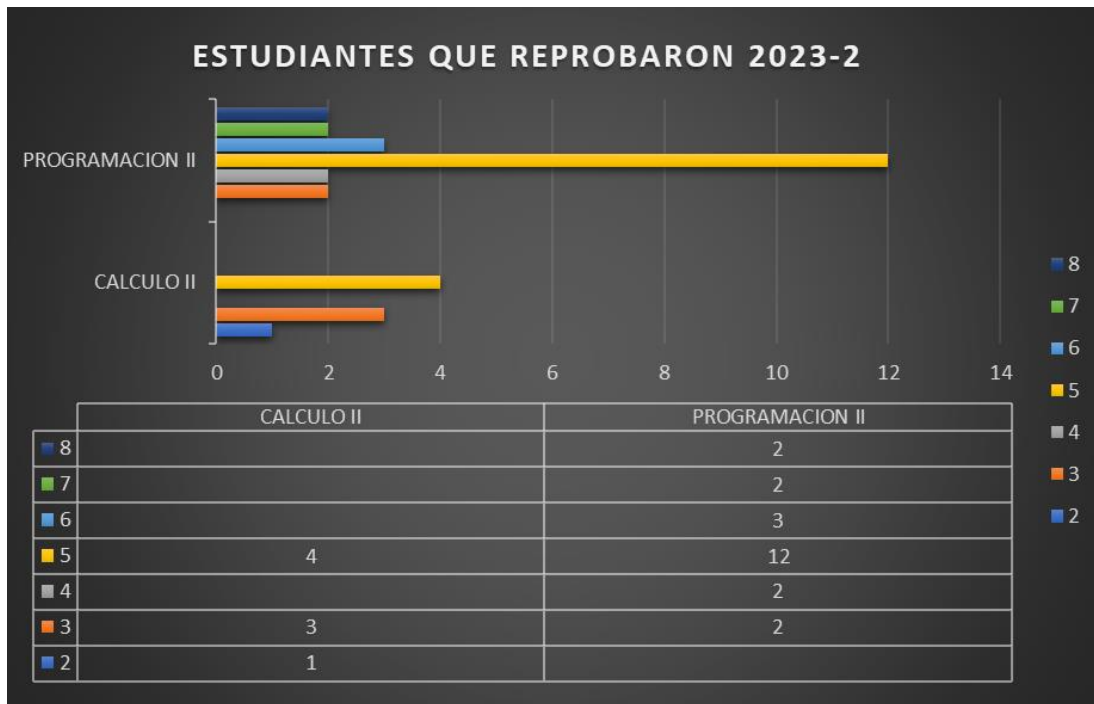
Sin embargo, como expresan Díaz y Svetlichich (2015), no es de ignorar el hecho de que la educación ha sido una de las actividades más afectadas por los cambios tecnológicos a través de la historia, haciendo que los docentes vayan incorporando permanentemente diferentes herramientas tecnológicas y digitales para así mitigar el número de estudiantes que se encuentran dentro de una zona de no aprendizaje directo.

En ese sentido, las tutorías se han consolidado como una estrategia pedagógica clave para reducir los índices de deserción y reforzar los procesos formativos, permitiendo un acompañamiento más personalizado y cercano entre docente y estudiante. Esto resulta particularmente importante en programas académicos de alta exigencia como la ingeniería de sistemas, donde los contenidos técnicos suelen requerir un refuerzo adicional que va más allá del aula tradicional.

Al interior de la Universidad Popular del Cesar, Seccional Aguachica, en el programa de Ingeniería de Sistemas, las asignaturas con un mayor índice estadístico de deserción y

reprobación son Programación II y Cálculo II, según lo reporta el programa Permanencia y Bienestar Institucional (PEBI), donde se encuentran los siguientes datos:

Ilustración 1: Reportes estadísticos de deserción y reprobación generados por PEBI.



Fuente: PEBI (programa Permanencia Y Bienestar Institucional)

1.3 OBJETIVO GENERAL

1.3.1 Objetivo General

Desarrollar un aplicativo web para la gestión de tutorías de estudiantes del programa de ingeniería de sistemas en La Universidad Popular del Cesar, seccional Aguachica.

1.3.2 Objetivos específicos

- Obtener los requisitos funcionales y no funcionales para el desarrollo del aplicativo web para la gestión de tutorías.

- Diseñar las vistas del aplicativo web para la gestión de tutorías, en base a los requisitos previamente obtenidos.
- Construir el aplicativo web para la gestión de tutorías de los estudiantes del programa de ingeniería de sistemas en la Universidad Popular del Cesar, Seccional Aguachica.
- Realizar pruebas de usabilidad y funcionalidad del aplicativo web para la gestión de tutorías, involucrando activamente a estudiantes, tutores y administradores del sistema.

1.4 DELIMITACIÓN

1.4.1 Temporal

La delimitación temporal para este proyecto es el tiempo neto de nuestro semestre académico, el cual consta de CUATRO MESES (4) desde el primer y último día, estipulado y asignado por nuestro calendario académico.

1.4.2 Espacial.

El proyecto se desarrollará en el campus de la Universidad Popular del Cesar, Seccional Aguachica, Colombia, en el programa de ingeniería de sistemas, dicha institución tiene sus raíces en la transformación del antiguo Instituto Tecnológico Universitario del Cesar, proceso que culminó con su reconocimiento como universidad en 1976. Tras varios trámites administrativos, en el año 2002 obtuvo la autorización oficial para operar como la Seccional Aguachica de la Universidad Popular del Cesar (UPC). Esta sede se encuentra ubicada en el municipio de Aguachica, considerado la segunda ciudad más importante del departamento del Cesar, tanto por su población como por su relevancia regional, después de la capital, Valledupar. Actualmente, la institución acoge a más de 1.200 estudiantes provenientes de municipios cercanos como Rio de Oro, San Martín, La Gloria, Gamarra, Pelaya y Pailitas. (LinkedIn, Universidad Popular del Cesar | Seccional Aguachica). (Rincón Páez, 2020)

1.4.3 Contextual.

Aguachica es un municipio con un entorno sociocultural y económico encaminado hacia el sector agropecuario, la agroindustria, así como el comercio, lo cual ha permitido el surgimiento de una serie de servicios de apoyo como los agrotecnicos, los financieros, el transporte junto con otros servicios empresariales; Además de personales dirigidos a los diferentes sectores económicos y a la población regional (Alcaldía de Aguachica Cesar, 2017).

1.4.3.1 Aspecto Geográfico

Aguachica, Cesar.

- **Fecha de fundación:** 16 de agosto de 1748
- **Fundador:** Don José Lázaro de Rivera
- **Descripción Física:**

El municipio de Aguachica está localizado al sur del departamento del Cesar a 301 Km. de Valledupar. Su cabecera municipal está localizada a los 08° 45” de latitud norte y 73°37’ 37” de longitud oeste del meridiano de Greenwich a 190 metros sobre el nivel del mar (msnm); Según las coordenadas X1:1.375.000, X2: 1.430.000; Y1: 1.035.000, Y2: 1.065.000. El municipio se localiza en la zona intertropical ecuatorial, con una extensión total de 876,26 Km², temperatura media de 28°C, y precipitación media anual de 1 835 mm, limita al norte con los municipios de La Gloria (Cesar) y El Carmen (Norte de Santander), por el Este con el municipio de Río de Oro (Cesar), por el sur con San Martín (Cesar) y Puerto Wilches (Santander), por el Oeste con el municipio de Gamarra (Cesar) y Morales (Bolívar).

Al municipio se puede acceder desde la Costa Atlántica colombiana a través de la vía Santa Marta – Aguachica que comunica los departamentos de Magdalena y Cesar; desde el sur del país se llega a través de la Troncal del Magdalena Medio, o a través de la vía Bogotá – Bucaramanga – Aguachica que comunica los departamentos de Cundinamarca, Boyacá, Santander y Cesar; de esta vía se desprende la carretera Aguachica – Ocaña,

que comunica el municipio con el nororiente y la carretera Aguachica - Gamarra que conecta al municipio con el río Magdalena y la región occidental, existen además carreteables que unen a Aguachica con la mayoría de las veredas, solo que se encuentran en estado de abandono y deterioro.

Otra vía de acceso importante es la aérea, el municipio cuenta con el aeropuerto Hacaritama, sin embargo, no se encuentra prestando el servicio óptimo y adecuado de comunicación con aeropuertos vecinos ni con las capitales departamentales más cercanas.

- **Extensión total:** 876,26 Km²
- **Altitud de la cabecera municipal (metros sobre el nivel del mar):** 50 y 200 msnm
- **Temperatura media:** 28°C° C
- **Distancia de referencia:** Dista de Valledupar, la capital a 301 Km.

Ilustración 2: Planos geográficos del municipio de Aguachica - Cesar.



Fuente: (Alcaldía de Aguachica Cesar, 2017)

2 MARCO REFERENCIAL.

El análisis del estado del arte del proyecto se basa en la importancia de la gestión de tutorías de la Universidad Popular del Cesar Seccional Aguachica que ha tenido una grande necesidad ya que el aula web solo era un apoyo generado para el soporte de evidencias académicas de las asignaturas ya matriculadas dentro de la carrera en cada semestre, dejando a un lado las tutorías. Esto conlleva a una falta de participación de los estudiantes a las diferentes tutorías por el problema de horarios de las materias matriculadas. Asimismo, en el proyecto de investigación se referencian monografías que tienen relación y que les valen como referentes teóricos, como se muestran a continuación:

2.1 MARCO HISTÓRICO.

A partir de una exhaustiva investigación, se identificaron varios estudios relacionados con la problemática abordada, los cuales comparten el mismo propósito. A continuación, se presentará una breve reseña de cada uno.

2.1.1 Antecedentes Internacionales

A nivel internacional (Zulay, 2023) realizó un proyecto de pregrado titulado “Aplicación web para gestionar la solicitud de tutorías académicas por los estudiantes en la Universidad Iberoamericana del Ecuador...” que tiene como objetivo Centrarse en resolver una necesidad que surge en la Universidad Iberoamericana del Ecuador, ya que no se cuenta actualmente con una aplicación web para gestionar el proceso de tutorías.

Teniendo en cuenta la situación actual respecto a la pandemia causada por Covid19, se debería pensar en diferentes versátiles de camino al aprendizaje, aunque la pandemia necesariamente no se consideró como un problema para el aprendizaje si genero un gran cambio a la docencia y se resalta la necesidad de impulsar actividades de compañía y recomendación en el trayecto de la carrera.

El autor optó por una metodología ágil SCRUM, además de una arquitectura de software M.V.C (Modelo Vista Controlador) como también de un frontend que es un framework de JavaScript y en parte del backend usaron firebase como servicio donde este forma parte de Google cloud platform, otro recurso que fue utilizado es el lenguaje de programación TYPESCRIPT llegando a la teoría de que la aplicación web se adapta a las modalidades acogidas por el departamento de titulación.

Este proyecto nos indica que la metodología ágil SCRUM es una muy buena alternativa para llevar a cabo el desarrollo de un software de estas características, acompañada del conocimiento y aplicación adecuada de los mismos, se pueden obtener resultados sumamente importantes y correctos para un proyecto de titulación. Además, también amplia el conocimiento acerca de esta metodología las más populares hoy en día para el desarrollo de software.

Otro estudio (Dick & Bryan, 2023) en su trabajo de titulación “Sistema web de gestión de tutorías pedagógicas para el control y seguimiento de actividades de los estudiantes de la Carrera de Software de la Universidad de Guayaquil” la intención del proyecto es permitirle a la carrera de software que contenga un sistema informático para poder

obtener un seguimiento y control de los estudiantes que asisten a las tutorías, también crear publicaciones que contenga recursos para apoyar las actividades realizadas dentro de la tutoría y, a partir de esto, ser posible realizar una evaluación. El uso de este sistema tiene sentido porque integra los datos de tutorías brindadas por cada tutor durante el año en curso. Al final del semestre, se genera un informe que contiene datos estadísticos sobre el número de estudiantes supervisados, el número de estudiantes con mejor desempeño y el número de estudiantes con peor desempeño.

El autor eligió la metodología SCRUM con el desarrollo de 6 Sprint y un total de 11 historias de usuarios, se empleó la tecnología: Node.js v18.8.0, Vue.js v5.0.4, Hyper v3.4.1, como framework Laravel v.8.75, y a su vez la base de datos MySQLv8.0.32 a la par con Navicat Premium v16.1.9. Definitivamente, los resultados obtenidos en la encuesta de criterios de validación de expertos conformados por el personal docente y director de la Carrera de Software de la Universidad de Guayaquil, muestran que con un 97.5 % de aceptación avalan la funcionalidad del sistema.

En definitiva, encontramos que una buena implementación de sistemas web como el desarrollado dentro de este proyecto de titulación optimizan y mejoran de manera muy detallada e incrementada los procesos o problemáticas a mejorar, en este caso usando una medición cuantitativa por medio de encuestas validadas y muestran que el resultado de la implementación fue correcta al encontrar un índice alto de aceptación dentro del mismo cuerpo estudiantil y cuerpo docente-administrativo.

El estudio (Mendoza, 2022) en su trabajo de titulación “implementación de un sistema para la gestión de tutorías para los estudiantes de la unidad educativa particular Jonh Pierre del hierro Melchiade” la finalidad del proyecto es que el sistema beneficie a la institución y a sus estudiantes con ayudas continuas para los estudiantes. A través del sistema se puede comprobar la eficacia de los profesores en cuanto a adherencia a las tutorías, provisión de temas tratados, periodo durante el cual se puede seguir el progreso de los estudiantes, y también es posible utilizar la tecnología como herramienta didáctica

La metodología que optó fue mediante un análisis aplicando técnicas de recolección de datos como son las encuestas y entrevistas y se utilizaron lenguajes de programación como (HTML, PHP, JavaScript, CSS) y en la parte de base de datos se utilizó (SQL).

Seguidamente, aunque optaron por una metodología de solamente análisis mediante herramientas de recolección de datos, el desarrollo del software fue muy práctico y estructurado, las tecnologías y los lenguajes utilizados dentro de ese mismo desarrollo al ser muy comunes y usados en la comunidad de programadores hace que sea optima y adecuada para obtener un buen producto como resultado final.

2.1.2 Antecedentes Nacionales.

A nivel nacional (Quimbayo, Llanos, Alarcón, & Borrero, 2019) realizó un proyecto de investigación titulado “Desarrollo de una aplicación web de seguimiento y deserción para los estudiantes de la corporación universitaria del Huila – CORHUILA” que tiene como objetivo adoptar estrategias que permitan realizar un seguimiento oportuno antes de que

el estudiante renuncie a sus aspiraciones, identificando factores claves socio-económicos, demográficos, psicográficos y de proceso de aprendizaje.

Todo lo anterior optando una metodología no experimental (pues en ningún momento se realizó intervención alguna en los sujetos o lugares de estudio); transversal (ya que se la información recolectada corresponde a un momento preciso en el tiempo) y descriptiva (pues tiene como objetivo precisar las características de un grupo dentro de una población).

Cabe destacar que en su proyecto no se encuentran indicios de un lenguaje de programación netamente utilizados, más sin embargo dentro de toda la estructura del documento podemos encontrar las varias y muchas formas de usar una herramienta de recolección de datos con el fin de estipular los fines de una idea centrada de desarrollo, fines como población, rango de edades y tipo de metodología que podría ser mejor usada dentro del software.

Así mismo (Higuera & Moreno, 2014) en su trabajo titulado “Implementación de aplicativo web 2.0 con fines Educativos para el área de biotecnología en el colegio la Amistad I.E.D.” como objetivo buscan brindar un sistema de educación completo y mejor aprendizaje en el área de la Biotecnología ya que esta es un énfasis en la Institución Educativa Distrital La Amistad, en donde los estudiantes podrán interactuar con los docentes y demás compañeros fuera de las aulas de clase por medio de la web y aprovechando de igual manera las redes sociales.

Para el desarrollo del aplicativo web, como metodología, hicieron una extensa recopilación de información acerca de las expectativas y necesidades que tienen los estudiantes de la jornada nocturna de dicha institución, esto con el fin de suplir las necesidades y generar gran interés por parte de los miembros de la institución educativa.

Finalmente (Castro & Rojas, 2023) en su trabajo presentado nombrado “Desarrollo de un aplicativo web que permite enlazar monitores académicos con estudiantes garantizando calidad, seguridad, accesibilidad y agilidad del servicio” expresan y plantean el objetivo de Desarrollar un aplicativo web que permita enlazar a monitores académicos con estudiantes garantizando la calidad, seguridad, accesibilidad y agilidad del servicio.

Se Basaron en la metodología Benchmarking que les permitió estudiar a los competidores, con el objetivo de mejorar los procesos realizados dentro de PLASIRO. La dinámica consiste en utilizar estas aplicaciones o soluciones existentes como un usuario, con base a las experiencias obtenidas analizar posibles puntos de mejora.

Al utilizar este tipo de metodología identifican oportunidades sobre cuales tomar mejoras y aprender mejores prácticas de otras empresas que en su momento realizaron o desarrollaron productos de las mismas características para solucionar problemáticas relacionadas como estas.

2.1.3 Antecedentes Locales

(Merchán & Merchán, 2014) en su proyecto de grado titulado *“Construcción e implementación de un objeto virtual de Aprendizaje en la plataforma Moodle, para facilitar el Proceso de enseñanza -aprendizaje de la asignatura de Ciencias básicas i (cálculo i), unidad formativa: Derivadas, para primer semestre de los estudiantes de Ingenierías, de la universidad popular del cesar, Seccional Aguachica.”* Nos demuestra y presenta como en este caso un Objeto Virtual de Aprendizaje (OVA), puede llegar a ser un gran apoyo pedagógico en las asignaturas de Ciencias Básicas (Calculo I) de los en el programa de ingeniería de sistemas de la Universidad Popular del Cesar, Seccional Aguachica (UPC). Este OVA obedece a la necesidad de generar también una alternativa pedagógica en temas específicos de la asignatura de Calculo I.

Bajo la metodología de desarrollo de Objetos Virtuales de aprendizajes (MEDEOVAS), llevaron a cabo este proyecto, adquiriendo los respectivos requerimientos del OVA, haciendo una propuesta Estructural, diseño y llegando al punto de un desarrollo donde con el hicieron un proceso de montaje de los recursos de este Objeto Virtual de Aprendizaje a través y con apoyo del estándar SCORM. Llevaron a cabo pruebas de calidad, manteniendo este objeto virtual dentro de la plataforma virtual de la UPC dándolo a conocer a docentes y estudiantes propiamente del área de Ciencias Básicas I.

2.2 MARCO TEÓRICO

Al contemplar los conceptos teóricos implementado en esta investigación, resaltamos las teorías que están enfocadas a través de los años en el desarrollo de aplicativos webs,

llevando las investigaciones a un punto de innovación propia para alcanzar el éxito deseado de los aplicativos webs.

Para enmarcar de manera idónea cada una de las teorías, se empieza por una teoría enfocada en el método pedagógico optado hoy en día en la Universidad Popular del Cesar, Seccional Aguachica; **La Teoría del Aprendizaje Constructivista**. Teoría que de acuerdo con (Ortiz Granja, 2015) en su artículo denominado *“El Constructivismo Como Teoría y Método De Enseñanza”* nos expresa como esta teoría mantiene un enfoque netamente educativo basándose en la idea de que el aprendizaje es un proceso activo y personal de cada individuo, en el cual, construyen sus propios conocimientos a través de la interacción con la información y su mismo entorno. Conceptos claves de la teoría del aprendizaje constructivista que podríamos resaltar dentro del entorno educativo de esta investigación serían:

- Construcción del conocimiento
- Zona de desarrollo próximo
- Aprendizaje autodirigido
- Contextualización

Si bien resaltamos que la Teoría del Aprendizaje Constructivista influye en la teoría y práctica educativa, promoviendo enfoques pedagógicos ya mencionados anteriormente que, al ser más centrados en el estudiante, como el aprendizaje basado en proyectos, el aprendizaje colaborativo y el enfoque en el desarrollo de habilidades de pensamiento

crítico y resolución de problemas. Seguidamente optamos por la que sería nuestra segunda teoría en esta investigación tal como **La Teoría del Andamiaje (Scaffolding)**. Teoría propuesta por el psicólogo y pedagogo norteamericano Jerome Bruner, basada en la fundamentación de que los tutores (docentes o alumnos) proporcionan soporte temporal a los estudiantes para que puedan alcanzar un nivel de comprensión más elevado, ya que como el psicólogo y pedagogo nos expresa, consiste en brindar guía y apoyo a los estudiantes para que puedan desarrollar y mejorar diferentes destrezas, conocimientos y actitudes.

Como nos expresa (Terán, 2020) en su artículo nombrado "*Jerome Bruner: La arquitectura del conocimiento*", uno de los términos más usados e influyente hoy en día sobre la Teoría de Andamiaje o Teoría de Bruner, es la misma palabra "*andamiaje*", cuando cada estudiante haya alcanzado ciertos conocimientos, los "*andamios*" se irán eliminando, permitiendo que nuevos conocimientos se sumen para facilitar aprendizajes más complejos. la influencia de estos guías y apoyos, ahora conocidos como tutores, docentes o estudiantes de alto rendimiento, no se limita al aula, sino que también se extiende fuera del ámbito curricular, convirtiéndose en un apoyo constante para el desarrollo académico y cognitivo de los estudiantes.

Por otra parte, este proyecto en su estructura involucra teorías netamente del ámbito de su desarrollo como en el área de sistemas informáticos dentro de las cuales relataremos las siguientes; **La Teoría de Sistemas Distribuidos**. Las aplicaciones web se construyen habitualmente como sistemas distribuidos, en los que varios elementos de hardware y software colaboran para ofrecer un servicio integrado. Esta teoría se basa en examinar

la interacción y coordinación de estos componentes para asegurar la eficacia, la resistencia a errores y la capacidad de escalar. Como afirma (Zettler), los sistemas para ser destruidos deben cumplir con específicas características tales como:

- Recursos compartidos
- Procesamiento simultáneo
- Escalabilidad
- Detección de errores
- Transparencia

Actualmente, los sistemas distribuidos son ampliamente aceptados y se emplean en gran medida en softwares modernos, como redes sociales, servicios de streaming de video y sitios de comercio electrónico. Al mismo tiempo, los sistemas centralizados están evolucionando hacia sistemas distribuidos para manejar mejor su escalabilidad. Seguidamente se tiene como segundo concepto teórico **La Teoría de la Arquitectura de Software**. Teoría que busca definir la estructura, vista y comportamiento de todo un sistema. Por lo tanto, se centra en todo el funcionamiento y ayuda a definir una infraestructura que normalmente requiere un nivel de software no tan elevado. Esta clase de arquitectura incluye elementos de software y hardware, que se emplean para diseñar un sistema compuesto.

Cabe agregar, como comparte (Huet) en su artículo *“Arquitectura de software: Qué es y qué tipos existen”*, la importancia de una buena arquitectura a la hora de desarrollar un software permite planificar de una manera óptima y mejorada para poder llevar a cabo el proyecto, es por tanto que esta teoría toma gran importancia al iniciar el desarrollo de una

proyecto que conlleve como resultado final un producto software, ya que determinará el ritmo, factores económicos y humanos durante el proceso de su desarrollo e implementación.

2.3 MARCO LEGAL

Dentro del marco legal, nuestro proyecto se centra en las siguientes leyes y normativas:

2.3.1 Ley 23 de 1982

En Colombia es conocida como la Ley de Derechos de Autor y establece un marco legal para la protección de las obras intelectuales en el país.

Aspectos clave de la Ley 23 de 1982:

- **Objeto de protección:** La ley protege las obras literarias, artísticas y científicas, así como las interpretaciones y ejecuciones artísticas, los fonogramas, las emisiones y los programas de radiodifusión.
- **Titular de derechos:** El autor de una obra es considerado el titular de los derechos de autor por defecto. Estos derechos también pueden ser transferidos o heredados.

- **Derechos de autor:** Los derechos de autor otorgan al autor el derecho exclusivo de autorizar o prohibir la reproducción, distribución, comunicación pública y transformación de su obra. Estos derechos son de carácter personal e intransferible.
- **Duración de la protección:** La protección de los derechos de autor en Colombia se extiende durante toda la vida del autor y, tras su fallecimiento, se transmite a sus herederos durante 50 años. En el caso de obras anónimas o seudónimas, la protección es de 50 años a partir de la publicación.
- **Entidades de gestión colectiva:** En Colombia, existen entidades de gestión colectiva, como la Sociedad Colombiana de Autores y Compositores (SAYCO) y la Sociedad de Gestión de Derechos Audiovisuales (EGEDA), que se encargan de recaudar y distribuir los derechos de autor en nombre de los creadores.
- **Sanciones por infracción:** La ley establece sanciones y multas para las personas o entidades que infringen los derechos de autor, como la reproducción no autorizada de obras protegidas. (Alcaldía Mayor de Bogotá D.C)

2.3.2 Ley 1581 de 2012

Es una parte importante de la regulación de la protección de datos personales en Colombia y busca garantizar la privacidad y seguridad de la información de las personas.

Las organizaciones que tratan datos personales están obligadas a cumplir con sus disposiciones y tomar medidas para proteger la información de los individuos. (Portal Único del Estado Colombiano)

Aspectos clave de la Ley 1581 de 2012:

- **Ámbito de aplicación:** La ley se aplica a todas las personas naturales o jurídicas, de carácter público o privado, que realicen el tratamiento de datos personales en Colombia.
- **Definición de datos personales:** La ley define los datos personales como cualquier información vinculada o que pueda asociarse a una persona natural. Esto incluye datos como nombres, cédulas, direcciones, números de teléfono, información financiera, y cualquier otro dato que permita identificar o hacer identificable a una persona.
- **Principios de protección de datos:** La ley establece una serie de principios que deben regir el tratamiento de datos personales, como el principio de finalidad (los datos deben ser recopilados y utilizados para propósitos legítimos), el principio de consentimiento informado (las personas deben dar su consentimiento explícito para el tratamiento de sus datos), y el principio de seguridad (se deben implementar medidas de seguridad para proteger los datos).

- **Derechos de los titulares de datos:** La ley reconoce una serie de derechos para las personas cuyos datos son tratados, como el derecho de acceso, rectificación, actualización, supresión (conocido como derecho al olvido) y oposición al tratamiento de sus datos personales.
- **Registro Nacional de Bases de Datos:** La ley establece la obligación de las organizaciones de registrar las bases de datos que contienen datos personales en el Registro Nacional de Bases de Datos, que es administrado por la Superintendencia de Industria y Comercio.
- **Transferencia internacional de datos:** La ley regula la transferencia de datos personales a países extranjeros, asegurando que se cumplan los estándares de protección de datos.
- **Sanciones por incumplimiento:** La ley establece sanciones para las organizaciones que no cumplan con sus disposiciones, que pueden incluir multas y otras medidas punitivas.

2.3.3 Ley 1915 de 2018.

En Colombia tiene como objetivo combatir esta problemática y garantizar el respeto a los derechos de propiedad intelectual y autor, así como en establecer sanciones y mecanismos de control para proteger los derechos de autor y propiedad intelectual.

Aspectos claves de la Ley 1915 de 2018:

- **Definición de piratería:** La ley define la piratería como la reproducción, distribución y comercialización no autorizada de obras protegidas por derechos de autor, incluyendo películas, música, software, libros, y otros contenidos protegidos.
- **Sanciones por piratería:** La ley establece sanciones y multas para las personas o entidades que se dedican a la piratería. Estas sanciones pueden ser significativas y se aplican tanto a nivel civil como penal.
- **Protección de derechos de autor:** La ley busca proteger los derechos de autor y la propiedad intelectual al otorgar a los titulares de derechos herramientas legales para combatir la piratería y perseguir a quienes infringen sus derechos.
- **Control en aduanas y comercio:** La ley permite a las autoridades aduaneras y de comercio llevar a cabo inspecciones y confiscar mercancías sospechosas de ser piratas, con el fin de evitar que productos piratas entren al mercado.

- **Colaboración con la industria:** La ley promueve la colaboración entre las autoridades y la industria para prevenir y combatir la piratería, incluyendo campañas de concientización y acciones conjuntas para hacer cumplir la ley.
- **Protección al consumidor:** La ley establece medidas para proteger a los consumidores, evitando que adquieran productos falsificados o piratas que puedan ser perjudiciales o de calidad deficiente.
- **Fomento de contenidos legales:** La ley busca promover la oferta de contenidos legales y autorizados en el mercado, brindando alternativas atractivas a la piratería. (Portal Único del Estado Colombiano)

2.3.4 Norma técnica colombiana 5854.

El objeto de la **Norma Técnica Colombiana (NTC) 5854** es establecer los requisitos de accesibilidad que son aplicables a las páginas web, que se presentan agrupados en tres niveles de conformidad: A, AA, y AAA. Esta norma fue desarrollada empleando como documento de referencia “Las Pautas de Accesibilidad para el Contenido web (WCAG) 2.0 del 11 de diciembre de 2008”.

Los cuatro principios fundamentales de esta norma son:

- **Perceptible:** El contenido debe ser visible y comprensible mediante alternativas (como subtítulos y descripciones de imágenes).
- **Operable:** Los usuarios deben poder navegar usando teclados u otros dispositivos de asistencia, y controlar elementos como el audio.
- **Comprensible:** El contenido debe ser claro, predecible y ofrecer ayudas ante errores o dificultades.
- **Robusto:** El sitio web debe funcionar correctamente con diferentes navegadores y tecnologías actuales y futuras.

2.3.5 Ley 1341 de 2009 (modificada por la Ley 1978 de 2019) – Ley TIC

Esta ley establece el marco regulatorio para el desarrollo y promoción de las Tecnologías de la Información y las Comunicaciones (TIC) en Colombia, considerando el acceso a estas como un derecho fundamental para el progreso del país.

Aspectos clave de la Ley 1341 de 2009 (y su modificación en 2019):

- **Objetivo de la ley:** Promover el acceso universal a las TIC, incentivar el desarrollo del sector y garantizar la prestación eficiente de estos servicios como un componente esencial del desarrollo económico y social

- **Modernización normativa (Ley 1978 de 2019):** Se unificaron regulaciones y se creó un único regulador convergente: la Comisión de Regulación de Comunicaciones (CRC). Además, se fortalecieron los incentivos para la inversión privada y se priorizó la conectividad en zonas rurales.
- **Derecho a la conectividad:** Se reconoce el acceso a Internet como un derecho habilitante, fundamental para el ejercicio de otros derechos como la educación, la participación ciudadana y el acceso a la información.
- **Fondo Único TIC:** Se consolidaron los fondos existentes en uno solo para financiar proyectos de infraestructura, alfabetización digital y contenidos educativos, con enfoque en poblaciones vulnerables y regiones apartadas.

2.4 MARCO CONCEPTUAL

APLICATIVO WEB: Una aplicación web es un software que se ejecuta en el navegador web. Las empresas tienen que intercambiar información y proporcionar servicios de forma remota. Utilizan aplicaciones web para comunicarse con los clientes cuando lo necesiten y de una forma segura. Las funciones más comunes de los sitios web, como los carros de compra, la búsqueda y el filtrado de productos, la mensajería instantánea y los canales de noticias de las redes sociales, tienen el mismo diseño que las aplicaciones web. Le permiten acceder a funcionalidades complejas sin la necesidad de instalar o configurar un software.

Las aplicaciones web son programas interactivos que pueden realizar una amplia variedad de tareas y funciones, desde procesar datos y realizar cálculos hasta proporcionar servicios en línea, como correo electrónico, redes sociales, banca en línea, compras en línea y mucho más. (Amazon, 2023).

BASE DE DATOS: Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos. (Oracle, 2020)

CSS: (Cascade Style Sheets) es un lenguaje de hojas de estilos creado para controlar la apariencia o presentación de los documentos definidos con HTML. Con este lenguaje se especifican una serie de reglas que describen el aspecto de un sitio web, como por ejemplo los colores, la tipografía o el tamaño de los elementos, entre otros. (Mundo Henry, 2011)

GESTIÓN ACADÉMICA: La gestión académica se define como el proceso orientado a mejorar los proyectos educativos institucionales y los procesos pedagógicos, con el fin de responder a las necesidades educativas locales y regionales. (Pedagogía Universitaria para la educación a distancia, 2016).

HOSTING: Un hosting web -o simplemente hosting- es un servicio que nos permite alojar nuestra página web en internet. Ahora bien, la información de nuestro sitio se guarda a su vez en un servidor web o en un conjunto de servidores web. (Jesús, 2022).

HTML: El HTML, acrónimo de HyperText Markup Language (Lenguaje de Marcado de Hipertexto en español), es un pilar fundamental de cualquier página web. Se trata de un lenguaje estándar que estructura y presenta contenido. Así mismo, es esencial en cualquier área del marketing digital y en el email marketing. (Easymailing, 2024).

HTML5: HTML5 es un estándar que sirve como referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado HTML) para la definición de contenido de una página web, como texto, imágenes, vídeos, juegos, entre otros. (Jiménez, 2019)

INTERFAZ DE USUARIO: La interfaz de usuario o UI (User Interface) es un concepto que abarca arquitectura de información, patrones y diferentes elementos visuales que nos permiten interactuar de forma eficaz con sistemas operativos y softwares de diversos dispositivos. (Corrales, 2019).

PHP: PHP es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario. (Souza, 2020).

ROLES DE USUARIO: Un rol de usuario permite vincular a los usuarios con una compañía, asignándoles responsabilidades dentro de la misma. Es posible crear múltiples roles de usuario y asociarles tareas específicas según sus funciones. De esta

manera, una misma compañía puede contar con distintos usuarios responsables de diversas actividades, facilitando la gestión y distribución del trabajo. (IBM, 2023).

SISTEMA: Se entiende por un sistema a un conjunto ordenado de componentes relacionados entre sí, ya se trate de elementos materiales o conceptuales, dotado de una estructura, una composición y un entorno particulares. Se trata de un término que aplica a diversas áreas del saber, como la física, la biología y la informática o computación. (Enciclopedia Concepto, 2021)

SISTEMAS DISTRIBUIDOS: Un sistema distribuido es un conjunto de programas informáticos que utilizan recursos computacionales en varios nodos de cálculo distintos para lograr un objetivo compartido común. Este tipo de sistemas, también denominados "computación distribuida" o "bases de datos distribuidas", usan nodos distintos para comunicarse y sincronizarse a través de una red común. Estos nodos suelen representar dispositivos de hardware físicos diferentes, pero también pueden representar procesos de software diferentes u otros sistemas encapsulados recursivos. La finalidad de los sistemas distribuidos es eliminar los cuellos de botella o los puntos de error centrales de un sistema. (Zettler, ATLISSIAN, 2024).

SOFTWARE: El término software es un vocablo inglés que fue tomado por otros idiomas y designa a todo componente intangible (y no físico) que forma parte de dispositivos como computadoras, teléfonos móviles o tabletas y que permite su funcionamiento. El software está compuesto por un conjunto de aplicaciones y programas diseñados para cumplir

diversas funciones dentro de un sistema. Además, está formado por la información del usuario y los datos procesados. (Enciclopedia Concepto, 2023).

SCRUM: La metodología Scrum es una de las llamadas metodologías ágiles, la cual se basa en la colaboración constante y consistente del equipo. Se trata de una metodología que en sus orígenes se centró en el desarrollo de software, aunque su aplicación está presente en cada vez más tipos de industrias. Scrum cuenta con enfoque flexible, siendo pensada especialmente para trabajar con proyectos en entornos donde se necesita obtener resultados rápidos. Trabaja con el ciclo de vida iterativo e incremental, donde se va liberando el producto de forma periódica aplicando las prácticas del trabajo colaborativo. (Drew, 2020).

TIC: tecnología que se utiliza para procesar, almacenar, recuperar y transmitir información y datos, así como para comunicarse, es la utilización de recursos, herramientas, equipos, programas informáticos o cualquier proceso de computación. Tiene otro significado destinado a la parte de la capacitación de los estudiantes a través del uso de herramientas que emplean procesos computacionales. (Ministerio de Tecnologías de la Información y las Comunicaciones, 2009).

TUTORÍAS ACADÉMICAS: La tutoría académica es un proceso de acompañamiento personalizado que tiene como objetivo mejorar el rendimiento escolar, solucionar problemas escolares y desarrollar hábitos de estudio y trabajo para evitar la reprobación, el rezago y el abandono escolar. De esta manera los estudiantes aprovechan mejor sus materias, se disminuyen los índices de rezago, hay mayor probabilidad de culminación

de sus estudios y la mejora en el promedio de calificaciones de los estudiantes. (Secretaria de salud, 2017).

3 DISEÑO METODOLÓGICO

En este capítulo se describe el proceso o técnica fundamental en la que se apoya el presente proyecto de investigación. En la sección 3.1 se presenta el tipo y enfoque de investigación. La sección 3.2 Describe la población que abarca el proyecto. La sección 3.3 Describe la muestra, y en La sección 3.4 explica las herramientas que se emplearán para recopilar la información. Como consecuencia de esta recolección se generarán hipótesis investigativas y estadísticas, generando con ello el análisis de las variables vinculadas en el proyectó.

3.1 TIPO Y ENFOQUE DE INVESTIGACIÓN.

Esta investigación, de carácter descriptivo con enfoque cualitativo, se orienta a comprender en profundidad los aspectos relacionados con la implementación de un aplicativo web, más allá de lo que pueden mostrar los datos cuantitativos. El enfoque cualitativo se adopta con el propósito de indagar y analizar las percepciones, experiencias y necesidades de los usuarios finales, lo cual resulta fundamental para identificar elementos subjetivos que inciden en el uso y aceptación del sistema. Asimismo, permite explorar las dinámicas, retos y particularidades que surgen en el contexto real de aplicación, proporcionando una visión más rica y detallada sobre factores como la

usabilidad, la experiencia del usuario y otros elementos contextuales que pueden influir directamente en el éxito del proyecto.

Para llevar a feliz término el trabajo monográfico se acudirá a la investigación descriptiva con enfoque cualitativo. La metodología de investigación descriptiva como nos cuenta (Niño Rojas, 2011) en su libro *“Metodología De La Investigación”*, lleva como propósito describir la realidad objeto de estudio, un aspecto de ella, sus partes, sus clases o las relaciones que se pueden establecer entre varios objetos, con el fin de esclarecer una verdad, corroborar un enunciado o comprobar una hipótesis. Se entiende como el acto de representar por medio de palabras las características de fenómenos, hechos, situaciones, cosas, personas y demás seres vivos, de tal manera que quien lea o interprete, los evoque en la mente.

Por otro lado, como nos dice (Sampieri, Fernández, & Baptista, 1997) en su libro *METODOLOGÍA DE LA INVESTIGACIÓN* la finalidad de los estudios y as investigaciones de carácter descriptivo tienen como propósito describir situaciones y eventos, es decir, *el cómo es y cómo se manifiesta este determinado fenómeno. “los estudios descriptivos buscan especificar las propiedades importantes de personas, grupos, -comunidades o cualquier otro fenómeno que sea sometido a análisis”* (Dankhe,1986). Así mismo, dentro del mismo capítulo de este mismo libro, *METODOLOGÍA DE LA INVESTIGACIÓN* nos explica que con este tipo de investigación se miden y evalúan diversos aspectos, dimensiones o componentes del mismo fenómeno o en su caso fenómenos a investigar.

3.2 POBLACIÓN Y MUESTRA.

3.2.1 POBLACIÓN

(Hernández & Baptista, 2014) la población es: “el conjunto de todos los casos que concuerdan con determinadas especificaciones” (p.174). Para (Arias, 2012) define como “...población un conjunto finito o infinito de elementos con características comunes para las cuales serán extensivas las conclusiones de la investigación...” (p.81).

La población universitaria se refiere al conjunto total de estudiantes activos en la Universidad Popular del Cesar, Seccional Aguachica, la cual actualmente ofrece siete programas académicos: Ingeniería de Sistemas, Ingeniería Agroindustrial, Ingeniería Ambiental y Sanitaria, Administración de Empresas, Contaduría, Economía y Tecnología Agropecuaria. Sin embargo, para los fines de este proyecto de grado, la población objetivo se delimita de manera específica en el programa de ingeniería de sistemas, el cual, en el periodo 2023-2, cuenta con un total de 423 estudiantes, distribuidos entre los semestres primero y décimo.

3.2.2 MUESTRA

(Tamayo & Tamayo, 2006) define la muestra como: "el conjunto de operaciones que se realizan para estudiar la distribución de determinados caracteres en totalidad de una población universo, o colectivo partiendo de la observación de una fracción de la población considerada" (p.176).

En el caso de (Palella & Martins, 2008) definen la muestra como: "...una parte o el subconjunto de la población dentro de la cual deben poseer características reproducen de la manera más exacta posible" (p.93).

La muestra está conformada por 31 estudiantes del programa de Ingeniería de Sistemas de la Universidad Popular del Cesar, Seccional Aguachica, que reprobaron las asignaturas Programación II y Cálculo II durante el periodo académico 2023-2.

Esta selección se realizó de manera intencional, considerando que estos estudiantes representan directamente el alto índice de reprobación en asignaturas críticas del plan de estudios, la cual presentan la necesidad, el interés o la demanda de acceder a tutorías académicas como se puede evidenciar en la siguiente tabla:

Ilustración 3: Tabla de asistencia de estudiantes a las tutorías en el periodo 2023-2

MATERIAS	SEMESTRES							TOTAL GENERAL
	2	3	4	5	6	7	8	
PROGRAMACION II		2	2	12	3	2	2	23
CALCULO II	1	3		5				8
TOTAL GENERAL	1	5	2	17	3	2	2	31

Fuente: Departamento de Registro y Control, UPC Seccional Aguachica.

Esta delimitación responde al enfoque del proyecto, el cual se orienta al desarrollo e implementación de un aplicativo web diseñado para facilitar la gestión de tutorías dentro de dicho programa académico.

3.3 INSTRUMENTOS DE RECOLECCIÓN DE DATOS

De acuerdo con (Sampieri, Fernández, & Baptista, 1997), dentro de su libro METODOLOGÍA DE LA INVESTIGACIÓN, todo trabajo monográfico se aplican instrumentos para medir las variables contenidas en las hipótesis. Esa medición es efectiva cuando el instrumento de recolección es el adecuado, resalta y representa las variables que se tienen en cuenta dentro del mismo trabajo investigativo. Por lo tanto, para este planteamiento y para el desarrollo mismo se utilizaron como instrumentos de recolección de datos, las entrevistas estructuradas y observación directa, acompañadas y soportadas sobre documentos y datos cuantitativos de PEBI, departamento de PERMANENCIA Y BIENESTAR INSTITUCIONAL de la Universidad Popular del Cesar Seccional Aguachica. (**Anexo 3**).

3.3.1 Instrumentos cualitativos

Las entrevistas estructuradas se definen como una herramienta de investigación que son extremadamente rígidas en sus operaciones y permiten muy poco o ningún margen para incitar a los participantes a obtener y analizar resultados. Por lo tanto, también se le conoce como una entrevista estandarizada y tiene un enfoque cuantitativo, se le aplicó a director de departamento, docentes y estudiantes.

3.4 HIPÓTESIS

Hernández, Fernández y Baptista (2014) señalan que una hipótesis es una proposición de ensayo acerca de las posibles relaciones entre dos o más variables. Esta proposición debe estar basada en el conocimiento previo y ser susceptible de comprobación mediante el método científico. Las hipótesis orientan el diseño del estudio y guían la recolección y análisis de datos.

3.4.1 Hipótesis de la investigación

La implementación de un aplicativo web para la gestión de tutorías en el programa de Ingeniería de Sistemas de la Universidad Popular del Cesar, Seccional Aguachica, contribuirá al apoyo y a optimizar la organización de las tutorías académicas.

3.5 VARIABLES

Según Hernández, Fernández y Baptista (2014), las variables son propiedades que pueden medirse y observarse en los sujetos o fenómenos de estudio, y que adoptan distintos valores. En el contexto de esta investigación, las variables se clasificaron en dos categorías principales: las independientes, que funcionaron como elementos causales o insumos dentro del sistema, y las dependientes, que evidenciaron los efectos o resultados derivados de la implementación del sistema inteligente.

3.5.1 Variable independiente

Implementación del aplicativo web

3.5.2 Variable independiente

Mejoramiento en el seguimiento y rendimiento académico de los estudiantes.

4 ESQUEMA TEMATICO

El desarrollo del aplicativo web para la gestión de tutorías académicas en el programa de Ingeniería de Sistemas de la Universidad Popular del Cesar, Seccional Aguachica, se estructuró conforme a las fases del ciclo de vida del software, garantizando su alineación con los objetivos del proyecto. Las tareas se organizaron en sprints cortos bajo la metodología ágil Scrum, lo cual permitió una planificación flexible, iterativa y centrada en las necesidades de los usuarios. A continuación, se presentan los resultados de cada fase, describiendo las actividades realizadas y los avances obtenidos en el diseño, desarrollo, implementación y validación del sistema.

4.1 Fase 1: Obtener los requisitos funcionales y no funcionales para el desarrollo del aplicativo web para la gestión de tutorías.

4.1.1 PLANIFICACIÓN INICIAL

Siguiendo la metodología scrum, el inicio del proyecto opta según su cronograma de actividades por una fase primaria denominada planeación inicial. Fase donde se denomina por medio de actividades y observaciones directas las variables como la duración total del proyecto, recursos que se necesitan y se tienen en disponibilidad, se definen roles y se planifican los sprint siguientes dentro del desarrollo del proyecto.

Por medio de esta fase se elabora un cronograma de actividades, en el cual de la misma manera incluyen fases o sprint con las cuales se desarrolla de una manera óptima y correcta el proyecto, tal y como lo designa la metodología Scrum. Se incluye en el cronograma la distribución de tiempo que le corresponde a cada una de las actividades o sprint. (**Anexo 1**).

4.1.2 Obtención De Requisitos Funcionales

Con el programa de actividades estipulado y siguiendo el orden de las actividades dentro de la metodología Scrum, se hace una obtención de requisitos funcionales y no funcionales para el desarrollo del aplicativo.

4.1.2.1 Reuniones con los representantes

Como representantes para la obtención correcta de los requisitos funcionales, se aplicó una entrevista estructurada a dos docentes que dictan tutorías dentro de la carrera de ingeniería de sistemas en la seccional, donde nos expresaban ciertos ítems para un adecuado desarrollo del aplicativo. (ver **Anexo 2**).

La entrevista estructurada consistía en preguntas netamente sobre cómo sería más accesible y optimizado el funcionamiento del aplicativo teniendo en cuenta los problemas ya identificados dentro del proceso de tutorías presenciales en la seccional. (ver **Anexo 3**).

4.1.2.2 Creación del archivo de requisitos

Los requisitos presentados en las siguientes tablas son los resultados de las respuestas de las entrevistas anteriormente llevadas a cabo a los representantes, se identifican cada uno con el rol dentro del aplicativo por parte de los funcionales.

Tabla 1: Requisitos Funcionales

Requisitos funcionales del Aplicativo Web Para Tutorías		
Ítem	Descripción	Rol
RF_01	La aplicación web requerirá un Login con usuario y contraseña para administradores, docentes y estudiantes.	Sistema
RF_02	El sistema debe permitir el registro de estudiantes y docentes para poder hacer uso de sus funciones respectivas en la aplicación.	Sistema
RF_03	El administrador podrá crear cuentas para docentes y estudiantes.	Admin
RF_04	El administrador podrá modificar los datos y el rol de los usuarios.	Admin
RF_05	El administrador podrá activar o desactivar un curso.	Admin
RF_06	El administrador podrá visualizar los usuarios (Docentes, alumnos) registrados en el sistema.	Admin
RF_07	El docente podrá agregar estudiantes a las asignaturas o cursos que esté dictando.	Docente
RF_08	El docente podrá eliminar estudiantes de las asignaturas o cursos que esté dictando.	Docente

RF_09	El docente podrá visualizar los estudiantes que estén registrados en sus asignaturas.	Docente
RF_10	El estudiante podrá visualizar los cursos de tutorías activos en ese momento.	Estudiante
RF_11	El estudiante podrá visualizar los cursos de tutorías a los que esté matriculado.	Estudiante
RF_12	El docente podrá visualizar los estudiantes que hayan asistido a sus asignaturas.	Docente

Fuente: Dirección de Programa y Docentes.

Tabla 2: Requisitos no funcionales

Ítem	Descripción
RNF01	El sistema debe calcular la cantidad total de usuarios registrados (docentes y estudiantes).
RNF02	Los usuarios deben tener privacidad en sus accesos y datos personales; solo los administradores verán la información general.
RNF03	El sistema debe validar las credenciales de usuario en cada inicio de sesión para garantizar la seguridad.
RNF04	El docente podrá ver información básica y el estado de registro de cada estudiante asignado a su clase.
RNF05	La plataforma estará disponible 24/7 para permitir acceso constante a la información de los usuarios.

RNF06	El sistema debe tener interfaces gráficas bien diseñadas y fáciles de usar para todos los usuarios.
RNF07	La aplicación web debe poseer un diseño “Responsive” a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.
RNF08	El docente podrá eliminar estudiantes de las asignaturas o cursos que esté dictando.
RNF09	El sistema llevará logos e imágenes representativas de roles o perfiles.
RNF10	El sistema llevará los colores y logos institucionales
RNF11	El estudiante podrá visualizar los cursos de tutorías a los que esté matriculado.

Fuente: Dirección de Programa y Docentes.

Historias de usuarios.

Tabla 3: RF01 LoginGeneral

RF01- LoginGeneral	La aplicación web requerirá un Login con usuario y contraseña para administradores, docentes y estudiantes.
Versión	versión 1.0
Autores	Camilo Arango; Luis Martínez
Fuentes	no cuenta por el momento
Objetivos asociados	

Descripción	Permitir el ingreso de los usuarios (Admin, docentes y estudiantes) al aplicativo.	
Precondición	Tener con anterioridad una cuenta con usuario y una contraseña.	
Secuencia Normal	Pasos	Acción
	1	Dar click en "Iniciar Sesión"
	2	Ingresar usuario.
	3	Ingresar contraseña.
	4	Dar click en Entrar.
Postcondición	El aplicativo reflejará que la sesión fue iniciada correctamente generando una vista propia según el rol logueado.	
Excepciones	Pasos	Acción
	1	Ingresar las credenciales incorrectas.
Rendimiento	Pasos	Cota de tiempo
	1	250 n segundos
Frecuencia esperada	3 registro por minuto	
Importancia	Importante.	
Urgencia	Inmediatamente.	
Comentarios	Sin comentarios.	

Fuente: Los autores.

Tabla 4: RF02 RegistroSistema.

RF02- RegistroSistema	El sistema debe permitir el registro de estudiantes y docentes para poder hacer uso de sus funciones respectivas en la aplicación.	
Versión	versión 1.0	
Autores	Camilo Arango; Luis Martínez	
Fuentes	no cuenta por el momento	
Objetivos asociados	-----	
Descripción	Registrar, usuarios docentes y estudiantes que para hacer uso de sus funciones.	
Precondición	Haber iniciado sesión como administrador en el aplicativo.	
Secuencia Normal	Paso	Acción
	1	Registrar usuario
	2	Ingresar datos
	3	Seleccionar rol
	6	Crear contraseña temporal
Postcondición	El aplicativo reflejará que el rol de la cuenta fue creado correctamente con los datos registrado.	
Excepciones	Paso	Acción
	1	Que el usuario ya esté creado
Rendimiento	Paso	Cota de tiempo
	1	250 n segundos

Frecuencia esperada	3 registro por minuto
Importancia	Importante.
Urgencia	inmediatamente
Comentarios	Sin comentarios.

Fuente: Los autores.

Tabla 5: RF03 ActivarCuentaUsuario.

RF03 - ActivarCuentaUsuario	El administrador podrá activar o desactivar las cuentas para docentes y estudiantes.	
Versión	versión 1.0	
Autores	Camilo Arango; Luis Martínez	
Fuentes	no cuenta por el momento	
Objetivos asociados	RF2	
Descripción	El administrador activará o desactivará las cuentas de usuarios.	
Precondición	Estar registrado en el aplicativo.	
Secuencia Normal	Paso	Acción
	1	Buscar usuario
	2	Identificar usuario
	3	Activar o desactivar
Postcondición	El aplicativo reflejará que la activación o desactivación fue exitosa.	

Excepciones	Paso	Acción
	1	Que el usuario no exista en el aplicativo.
Rendimiento	Paso	Cota de tiempo
	1	250 n segundos
Frecuencia esperada	3 registro por minuto	
Importancia	Importante.	
Urgencia	inmediatamente	
Comentarios	Sin comentarios.	

Fuente: Los autores.

Tabla 6: RF04 ModificarDatosUsuario.

RF04 - ModificarDatosUsuario	El administrador podrá modificar los datos y el rol de los usuarios.	
Versión	versión 1.0	
Autores	Camilo Arango; Luis Martínez	
Fuentes	no cuenta por el momento	
Objetivos asociados	RF2	
Descripción	El administrador podrá modificar los datos y el rol de los usuarios.	
Precondición	Haber perdido u olvidado la credencial de contraseña para ingreso de sesión.	
Secuencia Normal	Paso	Acción
	1	Buscar usuario

	2	Identificar usuario
	3	Activar o desactivar
Postcondición	El aplicativo reflejará que la modificación o actualización de datos fue exitosa.	
Excepciones	Paso	Acción
	1	Que el usuario no exista en el aplicativo.
Rendimiento	Paso	Cota de tiempo
	1	250 n segundos
Frecuencia esperada	3 registro por minuto	
Importancia	Importante.	
Urgencia	inmediatamente	
Comentarios	Sin comentarios.	

Fuente: Los autores.

Tabla 7: RF05 ActivarDesactivarCurso.

RF05 - ActivarDesactivarCurso	El administrador podrá activar o desactivar un curso.
Versión	versión 1.0
Autores	Camilo Arango; Luis Martínez
Fuentes	no cuenta por el momento
Objetivos asociados	

Descripción	El Administrador podrá activar o desactivar cursos para su visualización o no por parte de los usuarios.	
Precondición	Haber perdido u olvidado la credencial de contraseña para ingreso de sesión.	
Secuencia Normal	Paso	Acción
	1	Iniciar sesión
	2	Ingresar al apartado de gestión de cursos
	3	Activar o desactivar
Postcondición	El aplicativo reflejará que la modificación o actualización de datos fue exitosa.	
Excepciones	Paso	Acción
	1	Ingresar las credenciales incorrectas.
	2	El curso no tenga tutorías.
Rendimiento	Paso	Cota de tiempo
	1	250 n segundos
Frecuencia esperada	4 registro por minuto	
Importancia	Importante.	
Urgencia	inmediatamente	
Comentarios	Sin comentarios.	

Fuente: Los autores.

Tabla 8: RF06 VisualizarUsuarios.

RF06 - VisualizarUsuarios	El administrador podrá visualizar los usuarios (Docentes, alumnos) registrados en el sistema.	
Versión	versión 1.0	
Autores	Camilo Arango; Luis Martínez	
Fuentes	no cuenta por el momento	
Objetivos asociados	RF01, RF02	
Descripción	El Administrador podrá ver el total de usuarios que están registrados en el sistema.	
Precondición	Haber perdido u olvidado la credencial de contraseña para ingreso de sesión.	
Secuencia Normal	Paso	Acción
	1	Iniciar sesión
Postcondición	El aplicativo reflejará en una tabla los usuarios registrados.	
Excepciones	Paso	Acción
	1	Ingresar las credenciales incorrectas.
	2	El usuario no exista en el sistema.
	3	No ingresar con el rol de administrador.
Rendimiento	Paso	Cota de tiempo
	1	250 n segundos
Frecuencia esperada	4 registro por minuto	

Importancia	Importante.
Urgencia	inmediatamente
Comentarios	Sin comentarios.

Fuente: Los autores.

Tabla 9: RF07 AgregarEstudianteCurso.

RF07 - AgregarEstudianteCurso	El docente podrá agregar estudiantes a las asignaturas o cursos que esté dictando.	
Versión	versión 1.0	
Autores	Camilo Arango; Luis Martínez	
Fuentes	no cuenta por el momento	
Objetivos asociados	RF01, RF02, RF05	
Descripción	El docente podrá agregar estudiantes a las asignaturas que esté dictando.	
Precondición	Haber iniciado sesión como docente.	
Secuencia Normal	Paso	Acción
	1	Dar click en "Carrera"
	2	Seleccionar Asignatura
	3	Asignar Estudiante
Postcondición	El aplicativo reflejará que el estudiante fue asignado de manera correcta.	
Excepciones	Paso	Acción
	1	Que el estudiante no esté registrado.

	2	Que el estudiante no esté activo.
	3	
Rendimiento	Paso	Cota de tiempo
	1	250 n segundos
Frecuencia esperada	4 registro por minuto	
Importancia	Importante.	
Urgencia	inmediatamente	
Comentarios	Sin comentarios.	

Fuente: Los autores.

Tabla 10: RF08 EliminarEstudianteCurso.

RF08 – EliminarEstudianteCurso	El docente podrá eliminar estudiantes de las asignaturas o cursos que esté dictando.	
Versión	versión 1.0	
Autores	Camilo Arango; Luis Martínez	
Fuentes	no cuenta por el momento	
Objetivos asociados	RF09	
Descripción	El docente podrá eliminar estudiantes a las asignaturas que esté dictando.	
Precondición	Haber iniciado sesión exitosamente en rol de docente	
Secuencia	Paso	Acción
Normal	1	Dar click en “Carrera”

	2	Seleccionar Asignatura
	3	Eliminar Estudiante
Postcondición	El aplicativo reflejará que el estudiante fue asignado de manera correcta.	
Excepciones	Paso	Acción
	1	Que el estudiante no esté registrado.
	2	Que el estudiante no esté activo.
	3	Que el estudiante no esté matriculado en el curso.
Rendimiento	Paso	Cota de tiempo
	1	250 n segundos
Frecuencia esperada	3 registro por minuto	
Importancia	Importante.	
Urgencia	inmediatamente	
Comentarios	Sin comentarios.	

Fuente: Los autores

Tabla 11: RF09 VisualizarEstudiantes.

RF09 – VisualizarEstudiantes	El docente podrá visualizar los estudiantes que estén registrados en sus asignaturas.
Versión	versión 1.0
Autores	Camilo Arango; Luis Martínez

Fuentes	no cuenta por el momento	
Objetivos asociados	RF07	
Descripción	El docente podrá visualizar los estudiantes que se encuentran en las asignaturas que esté dictando.	
Precondición	Haber iniciado sesión exitosamente en rol de docente	
Secuencia Normal	Paso	Acción
	1	Dar click en "Carrera"
	2	Seleccionar Asignatura
Postcondición	El aplicativo reflejará una tabla con los estudiantes asignados en el respectivo curso.	
Excepciones	Paso	Acción
	1	Que el estudiante no esté registrado.
	2	Que el estudiante no esté activo.
	3	Que el estudiante no esté matriculado en el curso.
Rendimiento	Paso	Cota de tiempo
	1	250 n segundos
Frecuencia esperada	3 registro por minuto	
Importancia	Importante.	
Urgencia	inmediatamente	
Comentarios	Sin comentarios.	

Fuente: Los autores.

Tabla 12: RF10 VisualizarCursosActivos

RF10 – VisualizarCursosActivos	El estudiante podrá visualizar los cursos de tutorías activos en ese momento.	
Versión	versión 1.0	
Autores	Camilo Arango; Luis Martínez	
Fuentes	no cuenta por el momento	
Objetivos asociados	RF07	
Descripción	El estudiante podrá visualizar los cursos de tutorías que se encuentren activos en ese momento.	
Precondición	Haber iniciado sesión exitosamente en rol de estudiante.	
Secuencia Normal	Paso	Acción
	1	Login
	2	Dar click en “Cursos”
Postcondición	El aplicativo reflejará una vista con los cursos activos en el momento.	
Excepciones	Paso	Acción
	1	Que el estudiante no esté registrado.
	2	Que el estudiante no esté activo.
Rendimiento	Paso	Cota de tiempo
	1	250 n segundos

Frecuencia esperada	3 registro por minuto
Importancia	Importante.
Urgencia	inmediatamente
Comentarios	Sin comentarios.

Fuente: Los autores.

Tabla 13: RF11 VisualizarCursosMatriculados.

RF08 – VisualizarCursosMatriculados	El estudiante podrá visualizar los cursos de tutorías a los que esté matriculado.	
Versión	versión 1.0	
Autores	Camilo Arango; Luis Martínez	
Fuentes	no cuenta por el momento	
Objetivos asociados	RF09	
Descripción	El estudiante podrá visualizar los cursos de tutorías en los cuales se encuentre matriculado en ese momento.	
Precondición	Haber iniciado sesión exitosamente en rol de estudiante.	
Secuencia Normal	Paso	Acción
	1	Login
	2	Dar click en “Cursos”
	3	Cursos / Matriculados

Postcondición	El aplicativo reflejará una vista con los cursos y su contenido.	
Excepciones	Paso	Acción
	1	Que el estudiante no esté registrado.
	2	Que el estudiante no esté activo.
Rendimiento	Paso	Cota de tiempo
	1	250 n segundos
Frecuencia esperada	3 registro por minuto	
Importancia	Importante.	
Urgencia	inmediatamente	
Comentarios	Sin comentarios.	

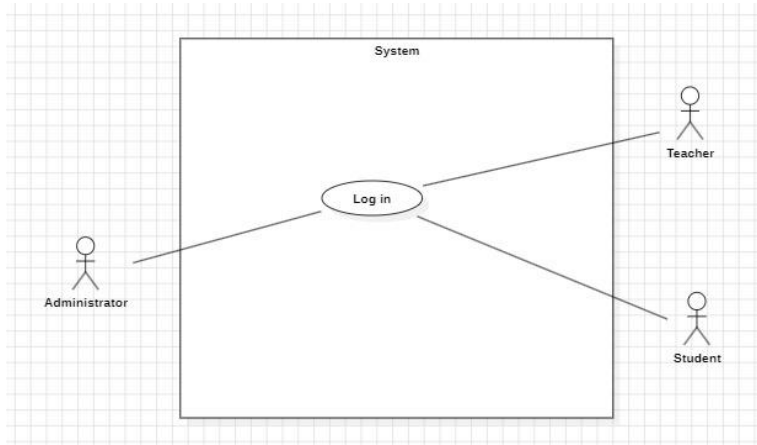
Fuente: Los autores.

Casos de uso.

En el lenguaje de modelado unificado (UML), los casos de uso son modelos que describen como diferentes tipos de usuarios interactúan con un sistema informático para resolver un determinado problema o necesidades. Por lo tanto, describen los objetivos de los usuarios, las interacciones entre los usuarios y el sistema, así como el

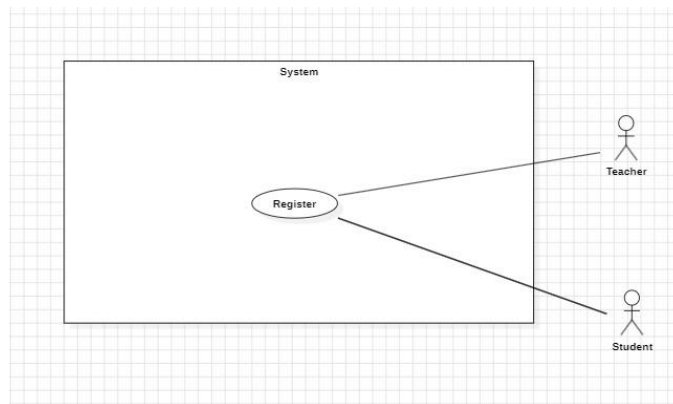
comportamiento que debe ejecutar el sistema para satisfacerlos. (PMOinformatica.com, 2021).

Figura 1: LoginGeneral



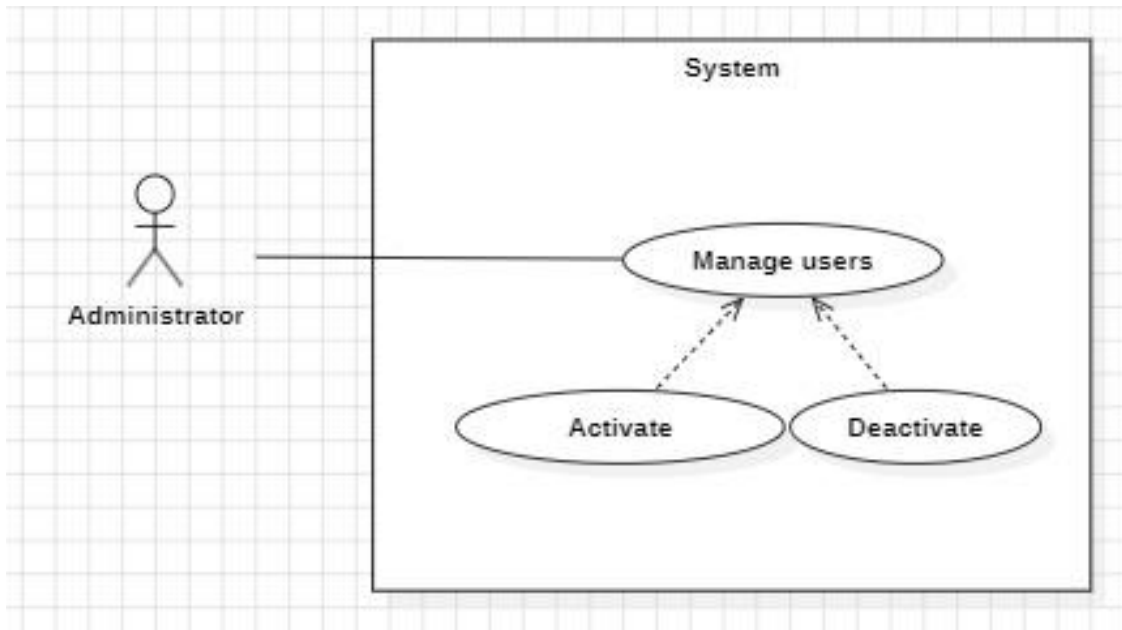
Fuente: Los autores.

Figura 2: RegistroSistema



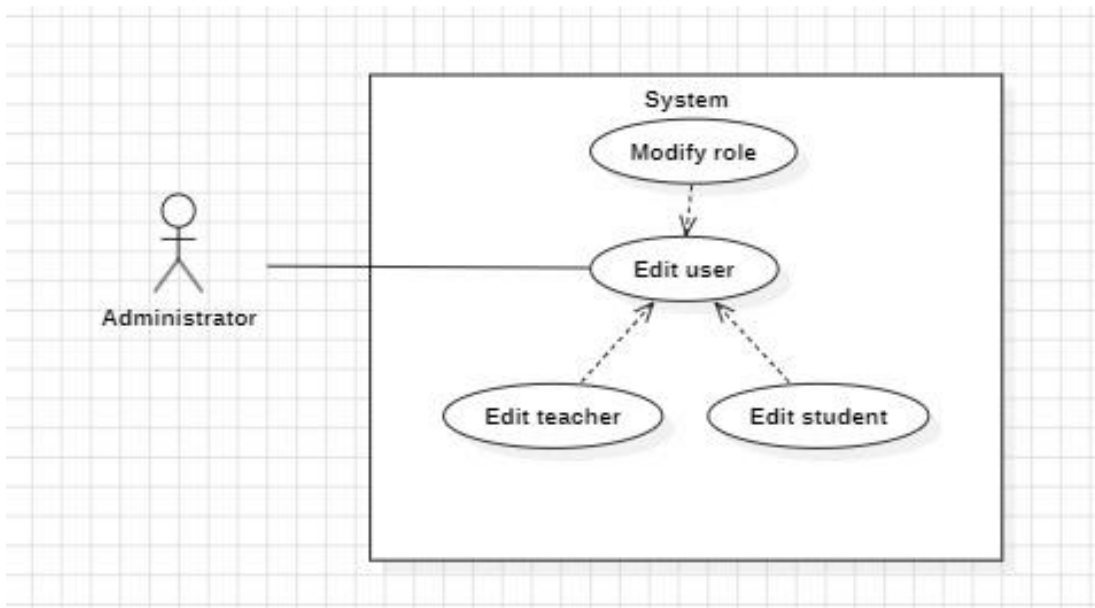
Fuente: Los autores.

Figura 3: ActivarCuentaUsuario



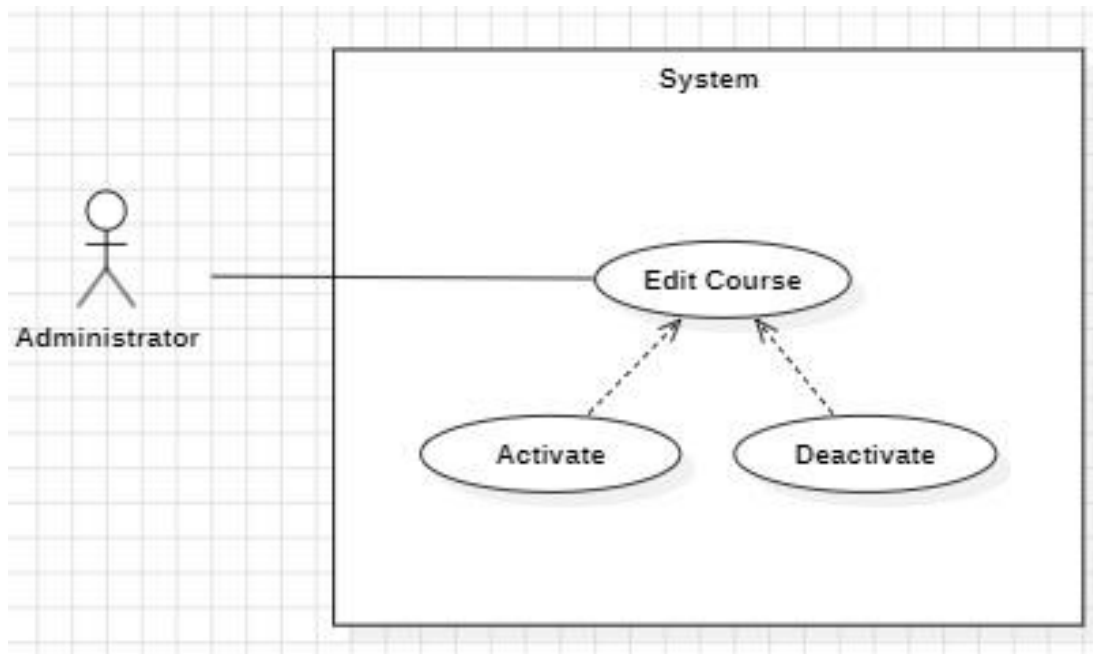
Fuente: Los autores

Figura 4: ModificarDatosUsuario



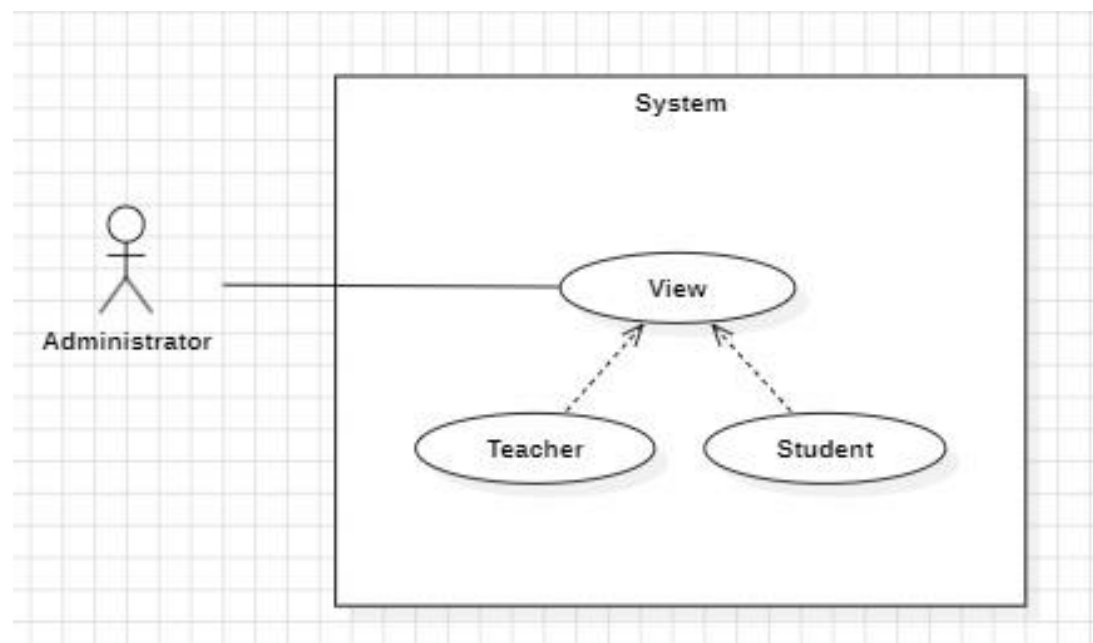
Fuente: Los autores.

Figura 5: ActivarDesactivarCurso



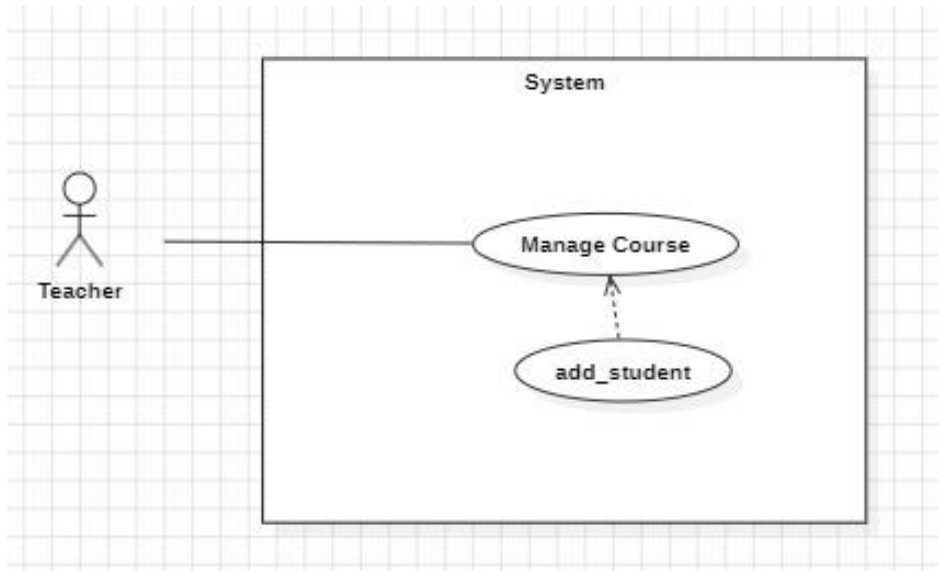
Fuente: Los autores.

Figura 6: VisualizarUsuarios.



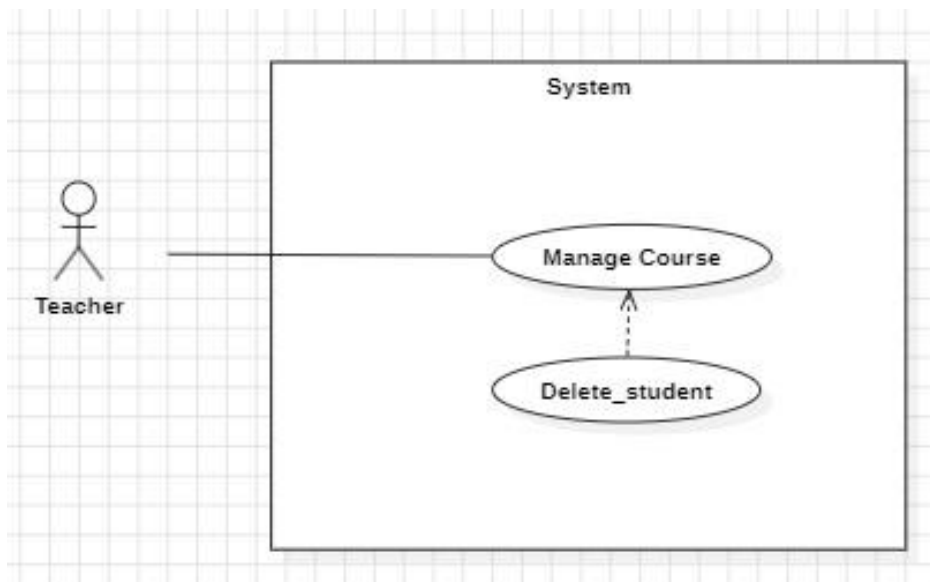
Fuente: Los autores.

Figura 7: AgregarEstudianteCurso



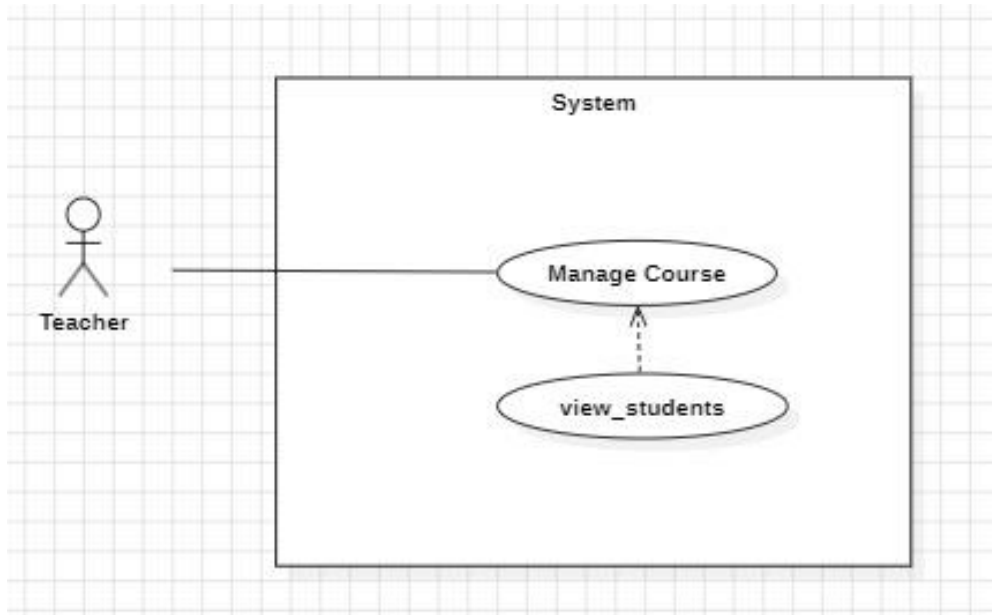
Fuente: Los autores.

Figura 8: EliminarEstudianteCurso



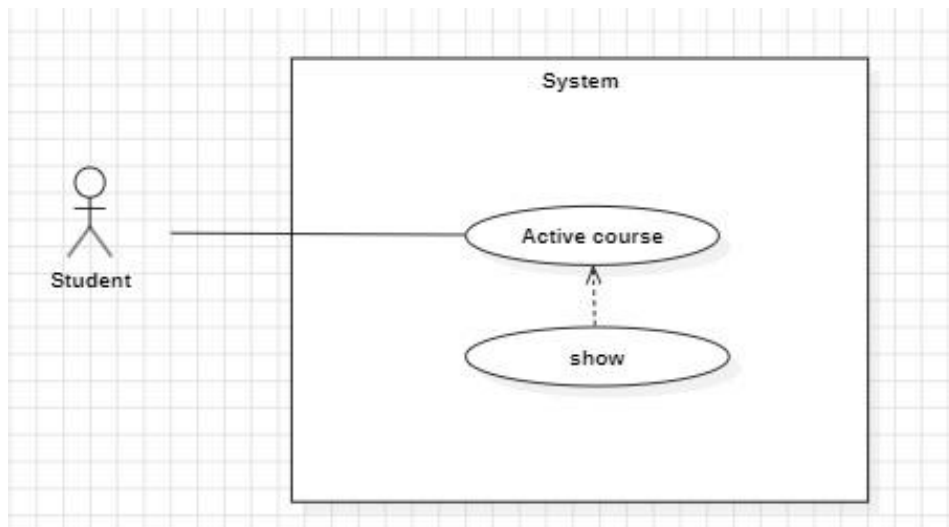
Fuente: Los autores.

Figura 9: VisualizarEstudiantes



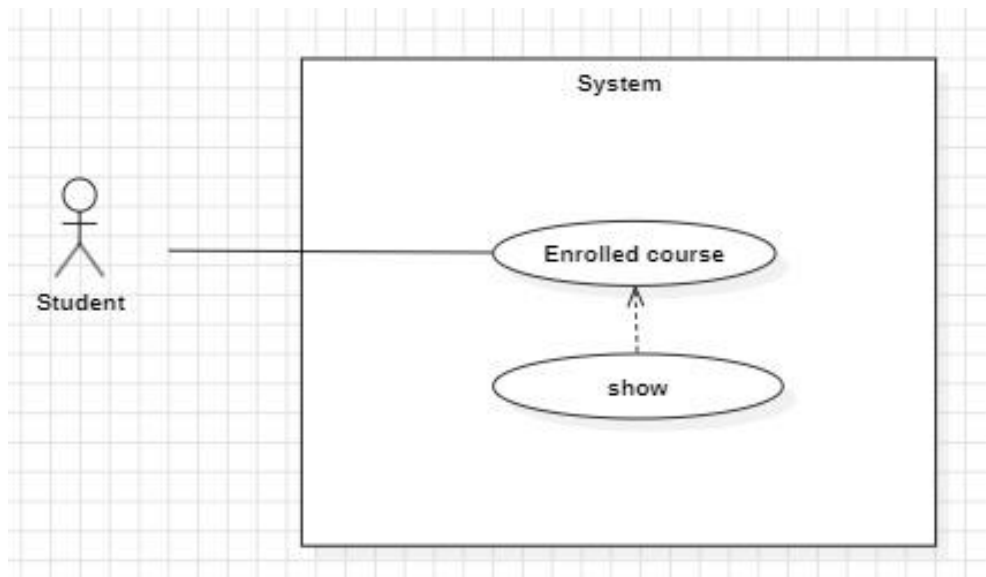
Fuente: Los autores.

Figura 10: VisualizarCursosActivos



Fuente: Los autores.

Figura 11: VisualizarCursosMatriculados



Fuente: Los autores.

4.2 Fase 2: Diseñando las vistas del aplicativo web para la gestión de tutorías, en base a los requisitos previamente obtenidos.

Diseñar las vistas del aplicativo web para la gestión de tutorías, en base a los requisitos previamente obtenidos.

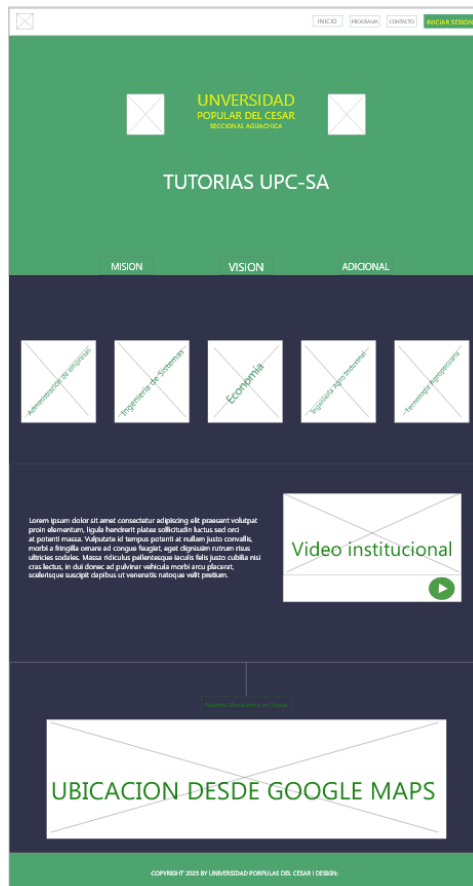
4.2.1 Diseño de vistas

El diseño de las vistas cumple un rol fundamental en esta parte del cronograma de actividades, ya que el diseño de este mismo será la primera vista para llevarla a codificación en los próximos objetivos y fase estipuladas. Aunque el software sea sofisticado y cuente con especificas funciones y procesos integrados que optimizan, el poseer un diseño atractivo es fundamental para captar la atención de tanto docentes como de estudiantes mismos, haciendo una accesibilidad plena y cómoda. Si la

aplicación tiene un diseño deficiente y una accesibilidad compleja y rustica, el interés por utilizarla se disparará rápidamente.

Los diseños se elaboraron por medio de Adobe XD, es ampliamente utilizada en el desarrollo de aplicaciones web y móviles porque simplifica la creación de prototipos interactivos y ayuda a visualizar cómo funcionará el producto antes de que se implemente.

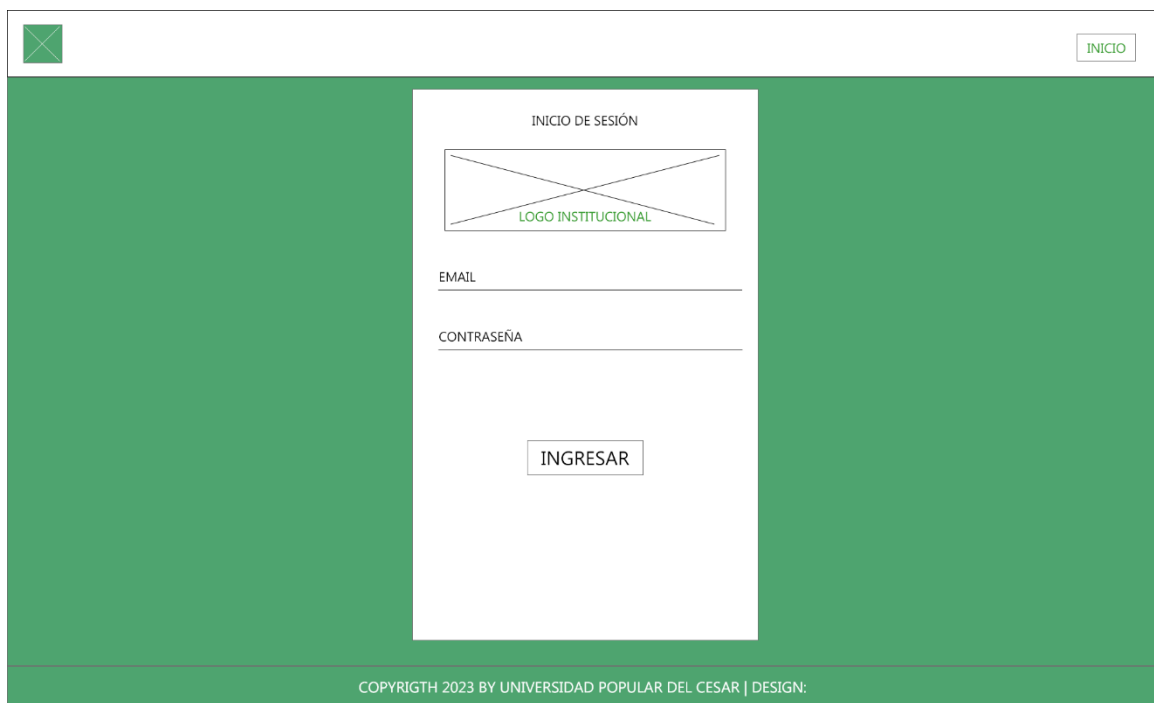
Ilustración 4: Vista Previa – ÍndexHome



Fuente: Los autores.

El anterior y primer *Mockup* es la página de inicio, la cual será por defecto la página índice (Índex) que verán todos los usuarios, se optó por un diseño con un estilo similar a todas las páginas que posee nuestra universidad, algo sencillo y cómodo para todos los usuarios, donde, además, tendrá a la vista toda la información de la seccional, como lo son el logo institucional, la misión, visión, y apartados necesarios como lo es una barra de navegación donde se encontrará: Inicio, Contacto, Programas e Iniciar sesión.

Ilustración 5: Vista Previa – Login



Fuente: Los autores.

El segundo *Mockup* es la página de Logueo o ingreso de credenciales de nuestros usuarios (Login), se opta por un diseño que mantiene los colores y los logos institucionales, una barra de navegación que provee un elemento de inicio, por si el usuario requiere un direccionamiento rápido al índice home del aplicativo.

Ilustración 6: Vista Previa – Index Admin



The screenshot shows a web interface for an administrator. On the left is a green sidebar with navigation links: ADMIN, INICIO, USUARIO, and CURSOS. The main content area is titled 'USUARIOS' and contains a table with the following data:

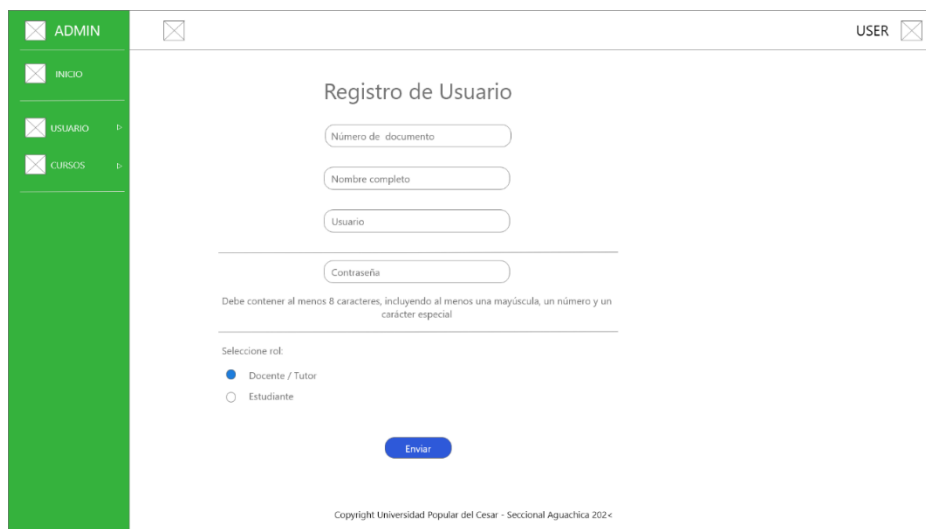
Documento	Nombre	Usuario	Rol	Estado
123456789	Camilo Andrés Arango Reyes	caarango	Estudiante	Activo

Below the table, it indicates 'Showing 1 to 10 of 57 entries' and includes pagination controls (Previous, 1, 2, 3, 4, 5, Next). The footer contains the text 'Copyright Universidad Popular del Cesar - Seccional Aguachica 202<'.

Fuente: Los autores.

El tercer *Mockup* es la página de índice para el administrador (Índice Admin), se opta por continuar el diseño que mantiene los colores y los logos institucionales, una barra de navegación que provee un elemento donde está el nombre del usuario más un logo que actúa de botón para cerrar sesión, un banner lateral con identificación del rol logueado apartados además de funciones como inicio, usuario, cursos.

Ilustración 7: Vista Previa – Add User



The screenshot shows a web interface for adding a user. On the left is a green sidebar with navigation links: ADMIN, INICIO, USUARIO, and CURSOS. The main content area is titled 'Registro de Usuario' and contains a form with the following fields:

- Número de documento
- Nombre completo
- Usuario
- Contraseña

Below the password field, there is a note: 'Debe contener al menos 8 caracteres, incluyendo al menos una mayúscula, un número y un carácter especial'. Underneath, there is a section for 'Selección rol' with two radio buttons: 'Docente / Tutor' (selected) and 'Estudiante'. At the bottom of the form is a blue 'Enviar' button. The footer contains the text 'Copyright Universidad Popular del Cesar - Seccional Aguachica 202<'.

Fuente: Los autores.

El cuarto *Mockup* es la página de registros de usuarios por parte del administrador (Add User), se opta por continuar el diseño anterior que mantiene los colores y los logos institucionales, la barra de navegación que provee el elemento donde está el nombre del usuario más el logo que actúa de botón para cerrar sesión, el banner lateral con identificación del rol logueado y los apartados además de las funciones como inicio, usuario, cursos.

Ilustración 8: Vista Previa – Edit User

ADMIN USER

INICIO

USUARIO

CURSOS

Editar Usuario

Número de documento

Nombre completo

Usuario

Contraseña

Debe contener al menos 8 caracteres, incluyendo al menos una mayúscula, un número y un carácter especial

Seleccione rol

Docente / Tutor

Estudiante

Estado del Usuario:

Activo

Inactivo

Regresar Guardar

Copyright Universidad Popular del Cesar - Seccional Aguachica 2024

Fuente: Los autores.

Quinto *Mockup*, página de modificación de datos de usuarios por parte del administrador (Edit User), continua el diseño anterior que mantiene los colores y los logos institucionales, la barra de navegación que provee el elemento donde está el nombre del usuario más el logo que actúa de botón para cerrar sesión, el banner lateral con identificación del rol logueado y los apartados además de las funciones como inicio,

usuario, cursos. Además, un formulario para modificar los datos, el rol y/o el estado del usuario, contiene al mismo tiempo dos botones adicionales dentro del formulario de modificación de los datos, para generar un retroceso o un correcto guardado de los datos.

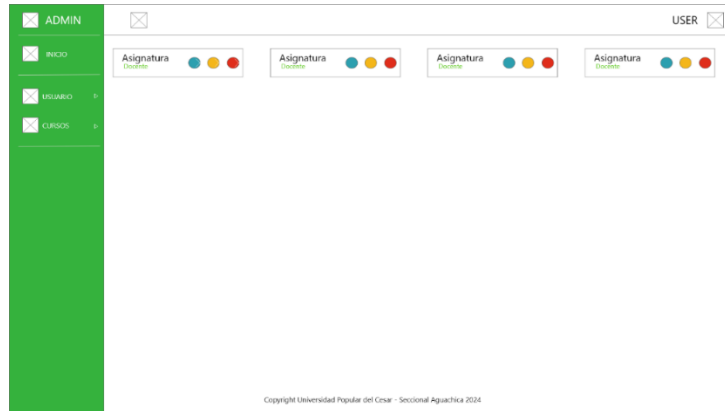
Ilustración 9: Vista Previa – Gestión User

Documento	Nombre	Usuario	Rol	Estado	Detalles
123456789	Camilo Andrés Arango Reyes	caarango	Estudiante	Activo	Activo

Fuente: Los autores.

Sexto *Mockup*, página de gestión de usuarios por parte del administrador (Gestión User), continua el diseño anterior que mantiene los colores y los logos institucionales, la barra de navegación que provee el elemento donde está el nombre del usuario más el logo que actúa de botón para cerrar sesión, el banner lateral con identificación del rol logueado y los apartados además de las funciones como inicio, usuario, cursos. Una tabla con datos reales de los usuarios que contiene: Documento, nombre, usuario, rol, estado y en la columna “detalles” un botón de ingreso para direccionar al apartado de modificación de datos y estado del usuario.

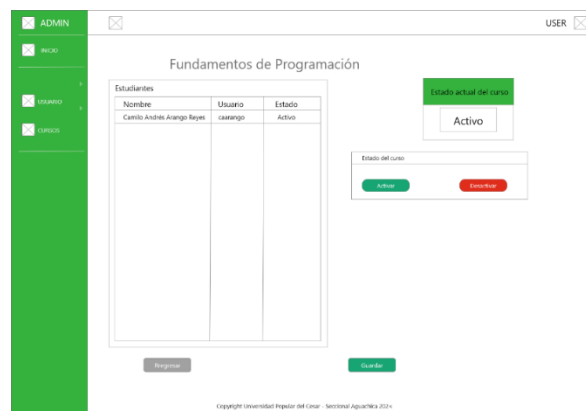
Ilustración 10: Vista Previa – Gestión Cursos



Fuente: Los autores.

Séptimo *Mockup*, página de gestión de cursos por parte del administrador (Gestión Course), en aspectos visuales continua el diseño anterior que mantiene los colores y los logos institucionales, la barra de navegación que provee el elemento donde está el nombre del usuario más el logo que actúa de botón para cerrar sesión, el banner lateral con identificación del rol logueado y los apartados además de las funciones como inicio, usuario, cursos. Cards con los nombres de las asignaturas que poseen tutorías, dando botones de funciones para modificar por individual los cursos.

Ilustración 11: Vista Previa – Gestión Cursos / Cursos individuales



Fuente: Los autores.

Octavo *Mockup*, página de gestión de cursos individuales por parte del administrador (Gestión Course Individual), en aspectos visuales continua el diseño anterior que mantiene los colores y los logos institucionales, la barra de navegación que provee el elemento donde está el nombre del usuario más el logo que actúa de botón para cerrar sesión, el banner lateral con identificación del rol logueado y los apartados además de las funciones como inicio, usuario, cursos. Tabla de estudiantes matriculados, estado del curso (Activado o desactivado) acompañado de dos botones con funciones de direccionar a una página anterior y el otro, para confirmar y hacer un guardado correcto de la modificación que se haga al curso.

Ilustración 12: Vista Previa – Índice Estudiante

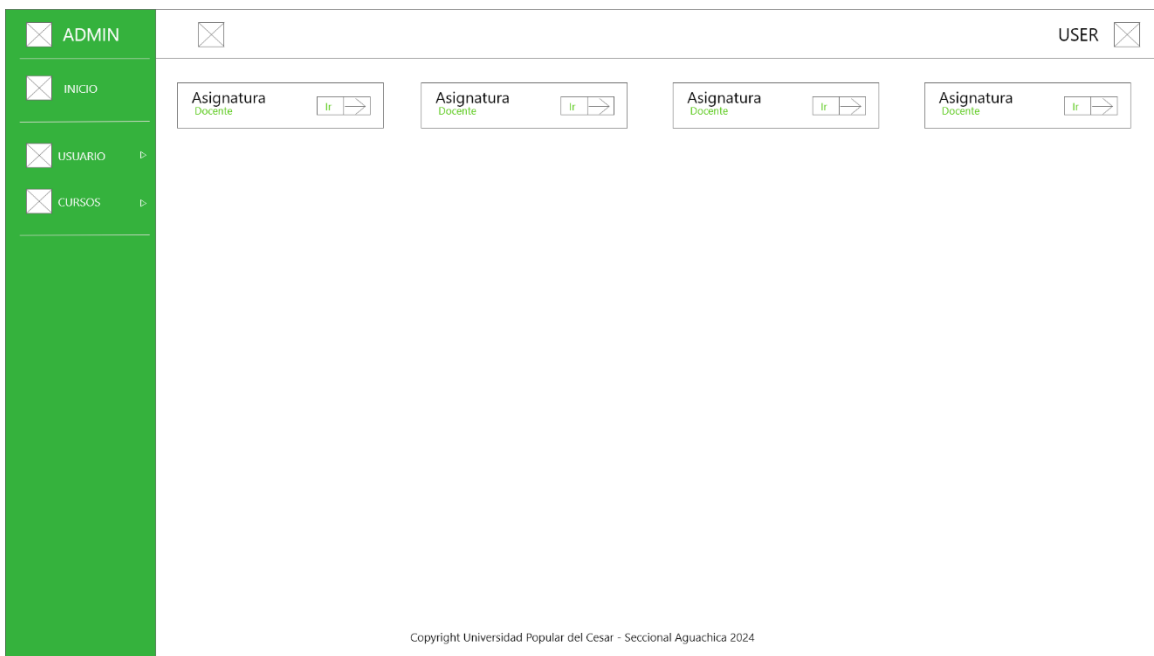


Fuente: Los autores.

Noveno *Mockup*, página de índice o primaria luego de ingreso de credenciales en rol estudiante (Índice Student), en aspectos visuales continua el diseño anterior que mantiene los colores y los logos institucionales, la barra de navegación que provee el elemento

donde está el nombre del usuario más el logo que actúa de botón para cerrar sesión, el banner lateral con identificación del rol logueado y los apartados además de las funciones como inicio, usuario, cursos. Tabla de estudiantes matriculados, estado del curso (Activado o desactivado) acompañado de dos botones con funciones de direccionar a una página anterior y el otro, para confirmar y hacer un guardado correcto de la modificación que se haga al curso.

Ilustración 13: Vista Previa – Cursos (Vista de estudiante)



Fuente: Los autores.

Décimo *Mockup*, página de cursos dentro del apartado de estudiante (Show course Student), en aspectos visuales continua el diseño anterior que mantiene los colores y los logos institucionales, la barra de navegación que provee el elemento donde está el nombre del usuario más el logo que actúa de botón para cerrar sesión, el banner lateral

con identificación del rol logueado y los apartados además de las funciones como inicio, usuario, cursos. Como anteriormente habían sido manejados los cursos, en esta vista previa encontramos Cards para cada asignatura conformada por el nombre de la asignatura, el nombre del docente y un botón de direccionamiento para la página individual de cada curso.

Ilustración 14: Vista Previa – Cursos Individuales (Vista de estudiante)

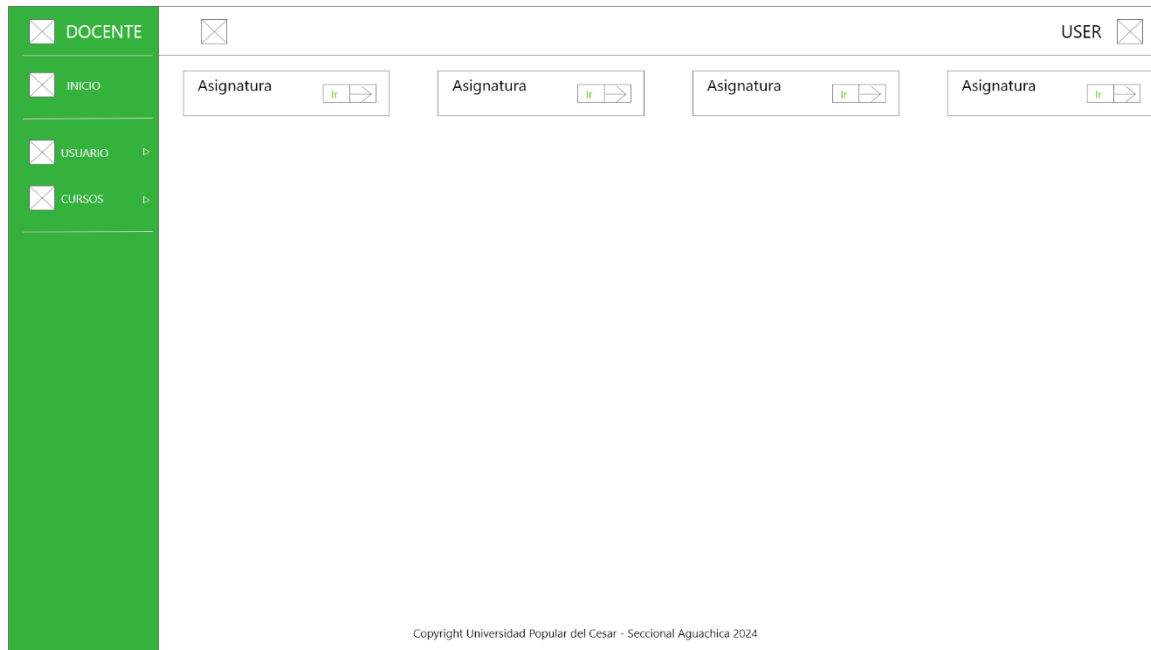


Fuente: Los autores.

Undécimo *Mockup*, página de cursos individuales dentro del apartado de estudiante (Curse Student), en aspectos visuales continua el diseño anterior que mantiene los colores y los logos institucionales, la barra de navegación que provee el elemento donde está el nombre del usuario más el logo que actúa de botón para cerrar sesión, el banner

lateral con identificación del rol logueado y los apartados además de las funciones como inicio, usuario, cursos. Conformado por el título propio de la asignatura, contenido académico del curso y un botón con direccionamiento a página anterior.

Ilustración 15: Vista Previa – Índice Docentes



Fuente: Los autores.

Duodécimo *Mockup*, página de índice o primaria luego de ingreso de credenciales en rol docente (Índice Docente), en aspectos visuales continua el diseño anterior que mantiene los colores y los logos institucionales, la barra de navegación que provee el elemento donde está el nombre del usuario más el logo que actúa de botón para cerrar sesión, el banner lateral con identificación del rol logueado y los apartados además de las funciones como inicio, usuario, cursos. Cards conformadas con nombre de las asignaturas

asignadas al docente o tutor, con botón de direccionamiento a cada curso individual de las asignaturas correspondientes.

Ilustración 16: Vista Previa – Cursos individuales de asignaturas (Docente)



Fuente: Los autores.

Decimotercer *Mockup*, página de índice o primaria luego de ingreso de credenciales en rol docente (Índex Docente), en aspectos visuales continua el diseño anterior que mantiene los colores y los logos institucionales, la barra de navegación que provee el elemento donde está el nombre del usuario más el logo que actúa de botón para cerrar sesión, el banner lateral con identificación del rol logueado y los apartados además de las funciones como inicio, usuario, cursos. Tabla con listado de estudiantes matriculados

en el curso, dos menús desplegables uno, para matricular estudiantes en las asignaturas propias y, el segundo menú desplegable para eliminar estudiante de la asignatura correspondiente, dos botones, uno direccional para página anterior y uno para confirmar modificación de asignatura.

4.2.2 Entrega de Wireframes

Continuando con la metodología de desarrollo scrum y las actividades determinadas en el cronograma de actividades del proyecto, una vez ya diseñadas las vistas en esta fase, se hace entrega de los Wireframes del proyecto (diseños anteriormente visualizados) para en el objetivo siguiente, mostrar el desarrollo inicial prototipado del frontend y el backend, haciéndose las socializaciones correspondientes sobre estos mismos y respectivos Wireframes.

4.3 Fase 3: Construyendo el aplicativo web para la gestión de tutorías de los estudiantes del programa de ingeniería de sistemas en la Universidad Popular del Cesar, Seccional Aguachica.

4.3.1 Pasar wireframes a prototipo

En esta fase, se desarrollaron las vistas del aplicativo web para la gestión de tutorías, transformando los wireframes iniciales en un prototipo funcional que refleja la estructura y diseño propuestos. Estas vistas no tienen aún funcionalidad conectada al backend, pero incluyen los elementos visuales y la interactividad básica utilizando **HTML, CSS, y JavaScript**.

Vistas Desarrolladas

- **1. Login del Usuario**

Objetivo:

Diseñar la interfaz de inicio de sesión que permitirá a los usuarios acceder al sistema.

Características:

- **HTML y CSS:**
 - Campos para correo electrónico y contraseña.
 - Botón de envío estilizado con CSS.
- **Interactividad:**
 - Uso de **JavaScript** para validaciones básicas, como asegurarse de que no se dejen campos vacíos.
 - Mensajes visuales para alertar al usuario en caso de errores.

2. Página de Inicio (index.html)

Objetivo:

Proveer una página de presentación para el sistema de tutorías "Tutorías UPC-SA", mostrando información institucional y opciones de navegación básica.

Características:

1. Encabezado:

- Incluye un logo de la Universidad Popular del Cesar.
- Menú de navegación con enlaces a secciones importantes:

- Inicio
- Programa
- Contacto
- Botón destacado para "Iniciar Sesión".

2. Sección principal:

- Fondo en video de alta calidad.
- Superposición de texto e imagen con un eslogan que resalta el sistema.

3. Ventanas flotantes (Misión, Visión y Adicional):

- Tres columnas con íconos y descripción breve de la misión, visión y datos adicionales de la universidad.
- Estilo interactivo que muestra contenido al pasar el cursor.

4. Sección "Programas":

- Carrusel con logos de los programas académicos disponibles.
- Cada logo redirige a la página de inicio de sesión.

5. Sección de video informativo:

- Video incrustado de YouTube que proporciona más detalles sobre la universidad.

6. Mapa de ubicación:

- Integración de un mapa interactivo de Google Maps mostrando la localización de la Universidad Popular del Cesar - Seccional Aguachica.

7. Pie de página:

- Derechos de autor y nombres de los desarrolladores del sistema.

Tecnologías utilizadas:

- **HTML y CSS:**
 - Diseño principal con **Bootstrap** y estilos personalizados.
- **JavaScript:**
 - Control de navegación y animaciones.
 - Scripts de Bootstrap y librerías adicionales como Owl Carousel y Lightbox.

3. Pantalla de "Agregar Nuevo Curso".

Objetivo:

Representar la vista que permitirá a los administradores registrar un curso en el sistema.

Estructura del Formulario:

- **Campos incluidos:**
 - Descripción del curso: Un campo de texto para agregar detalles del curso.
 - Semestre: Un campo de selección dinámica que permite elegir entre los semestres disponibles (1 al 10).
 - Cargar Imagen: Un campo para cargar imágenes representativas del curso.

- **Estilo Visual:**
 - Diseño centrado y responsivo usando **CSS**.
 - Botones estilizados para acciones como "Agregar Curso" y "Regresar a Gestión de Cursos".
- **JavaScript:**
 - Validaciones en tiempo real (por ejemplo, campos vacíos o tipos de archivos incorrectos).
 - Alertas visuales para notificar errores o éxito.

4. Pantalla de "Gestión de Cursos"

Objetivo:

Prototipar una vista centralizada que muestre los cursos existentes y facilite las operaciones básicas como agregar, editar o eliminar cursos.

Características:

- **Diseño Responsivo:**
 - Se utilizó un diseño basado en tarjetas (cards) para listar los cursos.
 - Cada tarjeta incluye:
 - Imagen del curso.
 - Descripción breve.
 - Opciones futuras (editar, eliminar, ver detalles).
- **Interactividad:**

- Eventos básicos como el resaltado de tarjetas al pasar el cursor con **JavaScript**.

5. Vista de Matricular Estudiantes

Objetivo:

Representar la interfaz para registrar la relación entre estudiantes y cursos.

Características:

- **Campos para:**
 - Selección de un estudiante.
 - Selección de un curso disponible.
 - Botón de envío estilizado con **CSS**.
- **Mensajes dinámicos:**
 - Notificaciones visuales para guiar al usuario en caso de errores o éxito.
- **Estilo Visual:**
 - Diseño enfocado en la simplicidad y claridad.

6. Página de Registro de Usuarios.

Objetivo:

Permitir el registro de nuevos usuarios en el sistema, diferenciando entre estudiantes y docentes.

Características:

1. Encabezado:

- Barra de navegación superior con el logo y opciones del usuario (como cerrar sesión).

2. Formulario de Registro:

- **Campos de entrada obligatorios:**

- Número de documento.
- Nombre.
- Usuario.
- Contraseña (validada para cumplir con requisitos de seguridad).

- **Rol del Usuario:**

- Selección entre **Estudiante** y **Docente** mediante botones de radio.

- **Botón de envío:**

Su objetivo será procesar el formulario a través del archivo `agregar_usuario.php` (con método POST).

3. Validación de Contraseña:

- Requisitos de la contraseña:
 - Al menos 8 caracteres.
 - Una mayúscula, un número y un carácter especial.

4. Diseño Responsivo:

- Utiliza **Bootstrap** para garantizar la adaptabilidad del diseño.
- **Metodología Aplicada**

1. **Uso de Wireframes:**

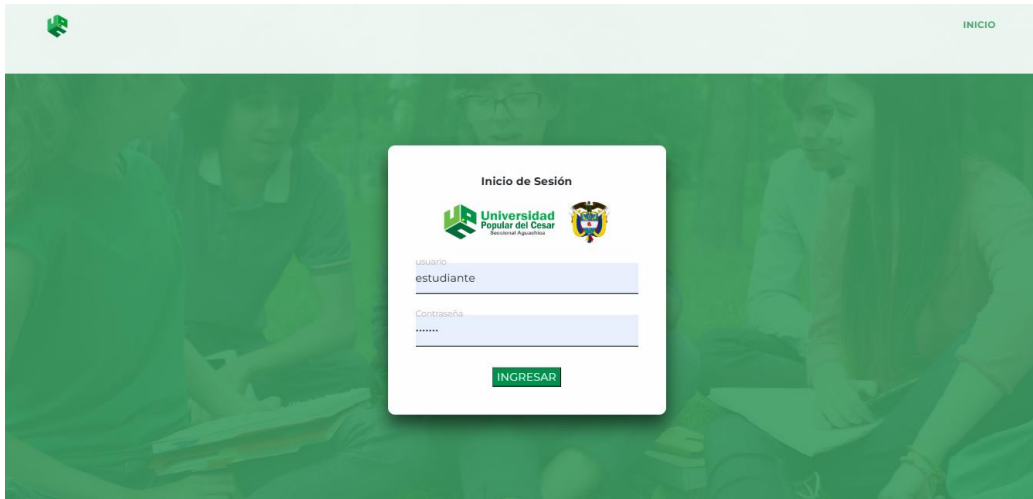
- Cada vista se desarrolló siguiendo las guías proporcionadas por los wireframes iniciales.
- Se respetaron las jerarquías de diseño y distribución de los elementos para garantizar coherencia.

2. **Prototipo Visual:**

- Las vistas no incluyen conexiones funcionales al backend, pero replican la experiencia visual e interactiva esperada.
- **Tecnologías Utilizadas**
 - a) **HTML y CSS:** Base para la estructura y el diseño.
 - b) **JavaScript:** Validaciones básicas y elementos de interacción.
 - c) **Bootstrap:** Diseño responsivo.
 - d) **SweetAlert:** Alertas visuales de validación y mensajes al usuario.

Capturas de las vistas:

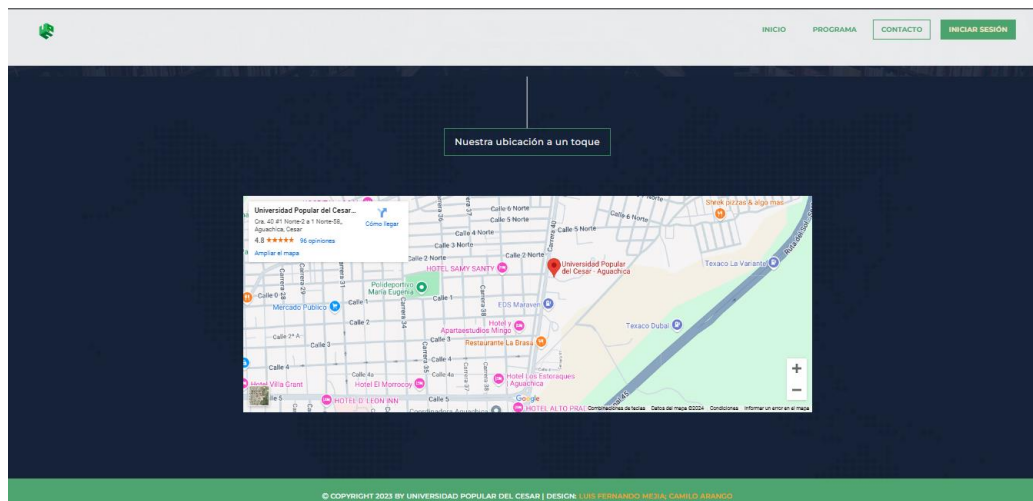
Ilustración 17: Login



Fuente: Los autores.

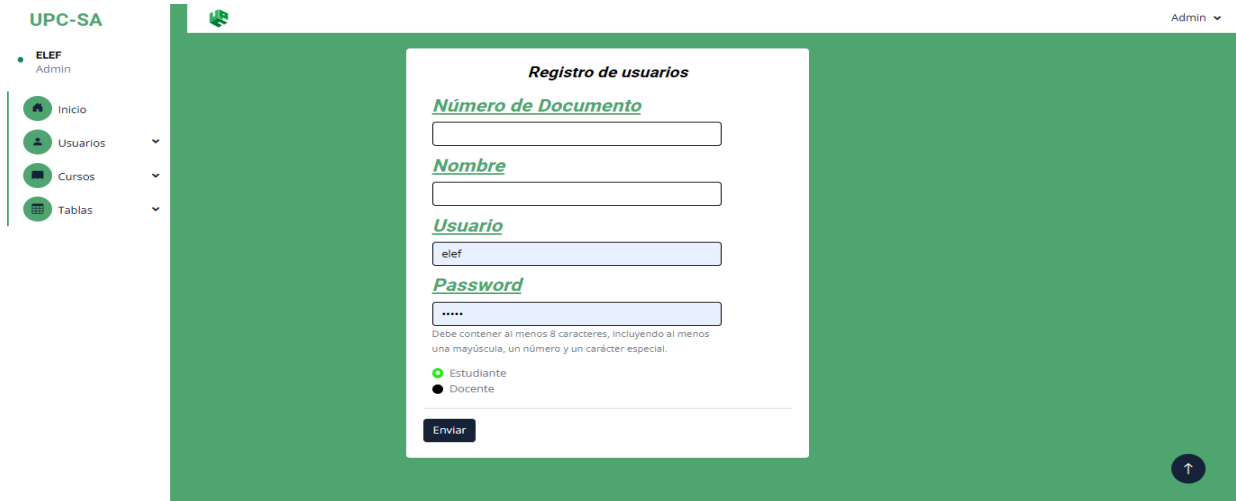
Ilustración 18: Índice





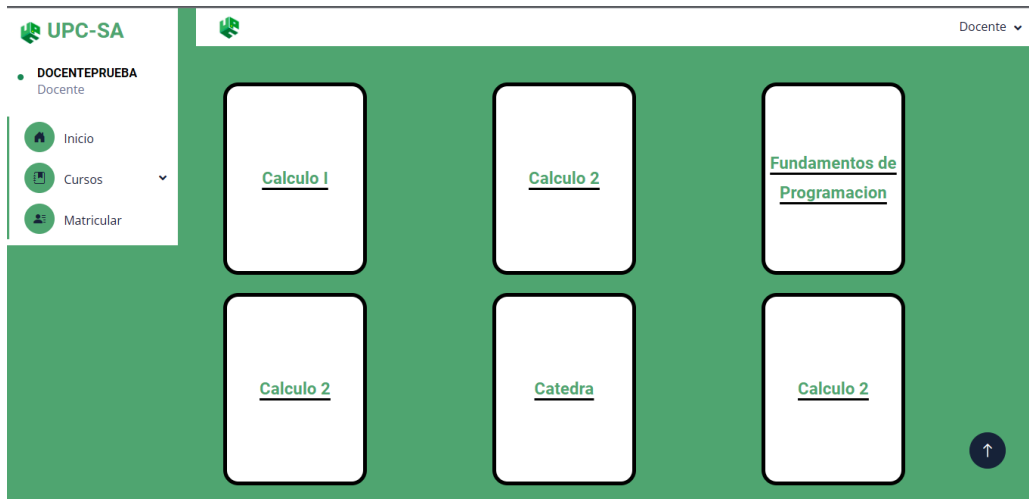
Fuente: Los autores.

Ilustración 19: Agregar usuarios



Fuente: Los autores.

Ilustración 20: Gestión de cursos



Fuente: Los autores.

Ilustración 21: Matricular usuarios



Fuente: Los autores.

4.3.2 Codificación del frontend y conexiones básicas con el Backend

Login del sistema

El login es un componente esencial del prototipo, que garantiza la autenticación de los usuarios en el sistema según sus roles y privilegios. Se implementaron tanto el frontend (interfaz gráfica) como el backend (lógica de autenticación) para la validación y redirección de usuarios.

Flujo de Funcionamiento

1. **Ingreso de datos:** El usuario introduce su nombre de usuario y contraseña en el formulario.

2. **Validación en el backend:**
 - Se busca al usuario en la base de datos.
 - Se verifica si el usuario está activo.
 - Se valida la contraseña introducida comparándola con el hash almacenado.

3. **Redirección:**
 - Si las credenciales son correctas, el usuario es redirigido a su respectiva interfaz según su rol.
 - Si las credenciales son incorrectas o el usuario está inactivo, se muestra un mensaje de error y se retorna al login.

Estructura del Prototipo

Frontend: Formulario HTML

El formulario está diseñado para captar las credenciales del usuario (nombre de usuario y contraseña). A continuación, se detalla su implementación:

Ilustración 22: Código del formulario del login en el frontend.

A screenshot of a code editor with a dark background and light-colored text. The code is HTML for a login form. It includes a form tag with action and method attributes, two input fields (text and password) with labels, and a submit button with a custom style. The code is numbered from 1 to 13.

```
1 <form action="LoginController.php" method="post">
2   <div class="user-box">
3     <input type="text" name="usuario" required="">
4     <label>usuario</label>
5   </div>
6   <div class="user-box">
7     <input type="password" name="contraseña" required="">
8     <label>Contraseña</label>
9   </div><center>
10    <button type="submit" style="background-color: rgb(0, 145, 76); color: azure;">INGRESAR</button>
11    <span></span>
12  </a></center>
13 </form>
```

Fuente: Los autores.

- **Método:** POST para enviar datos al backend de forma segura.
- **Inputs requeridos:** Nombre de usuario (usuario) y contraseña (contraseña).
- **Estilo:** Botón personalizado para mejor usabilidad.

Backend: Lógica de Login

El archivo LoginController.php gestiona la autenticación. Los pasos principales son:

1. Capturar los datos del formulario usando `$_POST`.
2. Consultar la base de datos para verificar si el usuario existe.

3. Validar el estado del usuario (activo/inactivo).
4. Comparar la contraseña ingresada con el hash almacenado mediante `password_verify`.
5. Establecer sesiones y redirigir al usuario según su rol.

Ilustración 23: Lógica de Login en el Backend

```

1  <?php
2  session_start();
3
4  // Capturar los datos del formulario
5  $usuario = $_POST['usuario'];
6  $contraseña = $_POST['contraseña'];
7
8  // Conexión a la base de datos
9  include "conexion.php";
10
11 // Consulta SQL para obtener el usuario
12 $consulta = "SELECT * FROM usuarios WHERE usuario='$usuario'";
13 $resultado = mysqli_query($conn, $consulta);
14
15 // Verificar si la consulta se ejecutó correctamente
16 if ($resultado) {
17     // Obtener la fila del usuario
18     $fila = mysqli_fetch_assoc($resultado);
19
20     // Verificar si se encontró el usuario
21     if ($fila) {
22         // Verificar si el usuario está activo
23         if ($fila['id_estado'] == 2) {
24             echo '<script>alert("No tienes permisos para acceder a esta página. Este usuario está inactivo.");
25             location.href = "login.html";</script>';
26         } else {
27             // Hash de la contraseña desde la base de datos
28             $contraseña_hash = $fila['contraseña'];
29
30             // Verificar si la contraseña ingresada coincide con el hash
31             if (password_verify($contraseña, $contraseña_hash)) {
32                 // Guardar sesión del usuario
33                 $_SESSION['usuario'] = $usuario;
34                 $_SESSION['idcargo'] = $fila['idcargo'];
35                 $_SESSION['id_estado'] = $fila['id_estado'];
36
37                 $idcargo = $_SESSION['idcargo'];
38
39                 // Redirigir según el cargo del usuario
40                 if ($idcargo == 1) {
41                     header("location: admin/home-admin.php");
42                 } elseif ($idcargo == 2) {
43                     header("location: students/index_student.php ");
44                 } elseif ($idcargo == 3) {
45                     header("location: admin/home_teach.php");
46                 }
47             }
48             echo $idcargo;
49             // else {
50             //     // Contraseña incorrecta
51             //     echo '<script>alert("Contraseña incorrecta.");
52             //     location.href = "login.html";</script>';
53             // }
54         }
55     } else {
56         // Usuario no encontrado
57         echo '<script>alert("Usuario no encontrado.");
58         location.href = "login.html";</script>';
59     }
60 } else {
61     // Error en la consulta SQL
62     echo "Error en la consulta: " . mysqli_error($conn);
63 }
64
65 // Liberar resultados y cerrar conexión
66 mysqli_free_result($resultado);
67 mysqli_close($conn);
68 ?>
69

```

Fuente: Los autores.

Tabla 14: Roles y Redirecciones

Rol	Página de destino
Administrador	admin/home-admin.php
Estudiante	students/index_student.php
Docente	admin/home_teach.php

Fuente: los autores.

- Los roles están definidos en el campo idcargo de la base de datos.
- Cada rol tiene un área específica en el sistema.

Validaciones de Seguridad

- **Contraseña Hash:** Las contraseñas están almacenadas en la base de datos como hashes generados por password_hash, lo que asegura su protección.
- **Estado del usuario:** Solo los usuarios activos pueden iniciar sesión.
- **Sesiones:** Se utilizan para mantener el estado del usuario mientras navega.

Agregar usuario:

El módulo de "Agregar Usuario" permite al administrador registrar nuevos usuarios en el sistema, asignándoles un rol y garantizando que se cumplan las validaciones necesarias para evitar duplicados y proteger la información sensible, como las contraseñas.

1. Descripción de la Funcionalidad

El administrador puede registrar usuarios proporcionando su información personal, como nombre, documento de identidad, usuario y contraseña. Adicionalmente, se puede asignar un rol (estudiante o docente) y se define su estado inicial como "activo".

2. Flujo de Funcionamiento

1. **Ingreso de datos:** El administrador completa el formulario con los datos del usuario.
2. **Validación previa:** El sistema verifica que el usuario no exista en la base de datos.
3. **Almacenamiento seguro:** La contraseña se cifra antes de almacenarse.
4. **Registro en la base de datos:** Los datos se guardan en la tabla de usuarios.
5. **Notificación:** El sistema informa al administrador si el registro fue exitoso o si ocurrió un error.

3. Estructura del Prototipo

Frontend: Formulario para Agregar Usuario

El formulario está diseñado para capturar los datos básicos del usuario, incluyendo su rol y contraseña. Además, incorpora validaciones básicas para garantizar que los campos requeridos sean completados.

Ilustración 24: Código del formulario para Agregar Usuario

```

1 <form action="agregar_usuario.php" method="post">
2   <div class="row mb-3">
3     <h3 style="color: #4fa57b;"><em>cu>Número de Documento</em></h3>
4     <div class="col-sm-10">
5       <input name="documento" required type="number" class="form-control" id="inputDoc">
6     </div>
7   </div>
8   <div class="row mb-3">
9     <h3 style="color: #4fa57b;"><em>cu>Nombre</em></h3>
10    <div class="col-sm-10">
11      <input name="nombre" required type="text" class="form-control" id="inputName">
12    </div>
13  </div>
14  <div class="row mb-3">
15    <h3 style="color: #4fa57b;"><em>cu>Usuario</em></h3>
16    <div class="col-sm-10">
17      <input name="usuario" required type="text" class="form-control" id="inputEmail3">
18    </div>
19  </div>
20  <div class="row mb-3">
21    <h3 style="color: #4fa57b;"><em>cu>Password</em></h3>
22    <div class="col-sm-10">
23      <input name="contraseña" required type="password" class="form-control" id="inputPassword3" oninput="validarContraseña(this)">
24      <small id="passwordHelp" class="form-text text-muted">Debe contener al menos 8 caracteres, incluyendo al menos una mayúscula, un número y un carácter especial.</small>
25    </div>
26  </div>
27
28  <!-- Radio Button Roll -->
29  <fieldset class="row mb-3">
30    <div class="col-sm-10">
31      <div class="form-check">
32        <input class="form-check-input" type="radio" name="id_cargo" required
33              id="gridRadios1" value="2" checked>
34        <label class="form-check-label" for="gridRadios1">
35          Estudiante
36        </label>
37      </div>
38      <div class="form-check">
39        <input class="form-check-input" type="radio" name="id_cargo"
40              id="gridRadios2" value="3">
41        <label class="form-check-label" for="gridRadios2">
42          Docente
43        </label>
44      </div>
45    </div>
46  </fieldset>
47  <hr>
48
49  <button type="submit" class="btn btn-primary">Enviar</button>
50 </form>

```

Fuente: Los autores.

Características del formulario:

- **Validaciones de Entrada:** Uso de required en los campos obligatorios y un mensaje de ayuda para la contraseña.
- **Selección de Rol:** Uso de botones de radio para definir el rol del usuario.
- **Envío Seguro:** Método POST para enviar los datos al servidor.

Backend: Controlador para Agregar Usuario

El archivo agregar_usuario.php se encarga de procesar los datos enviados desde el formulario, realizar las validaciones necesarias y registrar al usuario en la base de datos.

Ilustración 25: Código del controlador para Agregar Usuario

```
1 <?php
2 // Conexión a la base de datos
3 $servername = "localhost";
4 $username = "root";
5 $password = "";
6 $dbname = "roles";
7
8 $conn = new mysqli($servername, $username, $password, $dbname);
9 if ($conn->connect_error) {
10     die("Conexión fallida: " . $conn->connect_error);
11 }
12
13 function procesar_usuario($conn) {
14     if ($_SERVER["REQUEST_METHOD"] == "POST") {
15         $documento = $_POST["documento"];
16         $nombre = $_POST["nombre"];
17         $usuario = $_POST["usuario"];
18         $contraseña = $_POST["contraseña"];
19         $idcargo = $_POST["id_cargo"];
20         $id_estado = 1;
21
22         // Consultar si el usuario existe
23         $consulta_usuario = "SELECT * FROM usuarios WHERE documento = '$documento'";
24         $resultado_usuario = $conn->query($consulta_usuario);
25
26         if ($resultado_usuario->num_rows > 0) {
27             include("login.php");
28             ?>
29             <h1 class="bad">El usuario ya existe</h1>
30             <?php
31         }
32         else {
33             //cifrar contraseña
34             // $hash_contraseña = password_hash($contraseña, PASSWORD_DEFAULT);
35             // Insertar los datos en la tabla de la base de datos
36             $sql = "INSERT INTO usuarios (documento, nombre, usuario, contraseña, id_cargo)
37                 VALUES ('$documento', '$nombre', '$usuario', '$contraseña', '$idcargo', '$id_estado')";
38         }
39         if ($conn->query($sql) === TRUE) {
40             include("login.php");
41             ?>
42             <h1 class="good">Registro exitoso</h1>
43             <?php
44         } else {
45             include("login.php");
46             ?>
47             <h1 class="bad">Error</h1>
48             <?php
49             echo "Error: " . $sql . "<br>" . $conn->error;
50         }
51     }
52 }
53 ?>
```

Fuente: Los autores.

Pasos principales en el backend:

1. **Captura de Datos:** Se obtienen los valores enviados mediante \$_POST.
2. **Validación de Duplicados:** Se verifica si el documento del usuario ya existe en la base de datos.
3. **Cifrado de Contraseña:** Se utiliza una función de hash para proteger las contraseñas.
4. **Inserción de Datos:** Si las validaciones son exitosas, los datos se insertan en la tabla usuarios.
5. **Gestión de Errores:** Se informa al administrador en caso de duplicados o fallos en el proceso.

4. Validaciones de Seguridad

1. **Contraseñas Hasheadas:** Las contraseñas se cifran utilizando password_hash para protegerlas en la base de datos.
2. **Verificación de Duplicados:** Antes de insertar un nuevo usuario, se consulta la base de datos para evitar duplicados basados en el documento.
3. **Estado Inicial Activo:** Todos los usuarios registrados se crean con el estado activo por defecto.

5. Roles y Estado del Usuario

Tabla 15: Roles y Estado del Usuario

Rol	Estado Inicial
Estudiante	Activo
Docente	Activo

Fuente: Los autores.

El estado y el rol son asignados en el momento del registro. Estos valores se utilizan posteriormente para gestionar los permisos y accesos dentro del sistema.

6. Casos de Uso Documentados

- **Registro Exitoso:** El sistema muestra un mensaje de éxito al administrador cuando los datos son válidos y el registro se completa correctamente.
- **Usuario Existente:** Si el documento ya está registrado, se informa al administrador mediante un mensaje de error.
- **Error en la Base de Datos:** Si ocurre un fallo al insertar los datos, el sistema notifica al administrador y muestra los detalles del error.

Editar usuario:

El módulo "Editar Usuario" permite al administrador modificar la información de un usuario existente en el sistema. Esto incluye datos personales, contraseña, rol y estado. Este módulo garantiza la gestión eficiente de los usuarios registrados.

1. Descripción de la Funcionalidad

El administrador puede actualizar la información de los usuarios según sea necesario.

Esto incluye:

- Modificar datos personales, como nombre y documento.
- Cambiar la contraseña con seguridad.
- Ajustar el rol (estudiante o docente).
- Actualizar el estado del usuario (activo o inactivo).

2. Flujo de Funcionamiento

1. **Selección del Usuario:** El administrador selecciona un usuario desde la lista de usuarios.
2. **Carga de Datos Existentes:** Los datos actuales del usuario se precargan en el formulario para su edición.
3. **Edición de Datos:** El administrador modifica los campos necesarios.
4. **Almacenamiento de Cambios:** Los datos actualizados se guardan en la base de datos.

5. **Notificación:** El sistema informa si la actualización fue exitosa o si ocurrió un error.

3. Estructura del Prototipo

Frontend: Formulario para Editar Usuario

El formulario permite al administrador modificar los datos del usuario seleccionado. Todos los campos son editables, y se incluyen opciones para cambiar el rol y el estado.

***Ilustración 26:** Código del formulario para Editar Usuario*

```

1 <form action="<?php echo $_SERVER['PHP_SELF'] ?>" method="post">
2 <div class="row mb-3">
3 <h3 style="color: #4fa570;"><em><u>Número de Documento</u></em></h3>
4 <input type="text" name="documento" value="<?php echo $documento ?>">
5 </div>
6 <div class="row mb-3">
7 <h3 style="color: #4fa570;"><em><u>Nombre</u></em></h3>
8 <input type="text" name="nombre" value="<?php echo $nombre ?>">
9 </div>
10 <div class="row mb-3">
11 <h3 style="color: #4fa570;"><em><u>Usuario</u></em></h3>
12 <input type="text" name="usuario" value="<?php echo $usuario ?>">
13 </div>
14 <div class="row mb-3">
15 <h3 style="color: #4fa570;"><em><u>Contraseña actual</u></em></h3>
16 <div class="col-sm-10">
17 <input type="password" class="form-control" name="contraseña_actual">
18 </div>
19 </div>
20 <div class="row mb-3">
21 <h3 style="color: #4fa570;"><em><u> Contraseña Nueva </u></em></h3>
22 <div class="col-sm-10">
23 <input type="password" class="form-control" value="<?php echo $contraseña ?>" name="contraseña">
24 </div>
25 </div>
26
27 <!-- Radio Button Roll -->
28 <fieldset class="row mb-3">
29 <!-- Cargo -->
30 <h3 style="color: #4fa570;"><em><u> Cargo </u></em></h3>
31 <div class="col-sm-10">
32 <label>
33 <input type="radio" name="idcargo" value="2" <?php echo ($idcargo == 2) ? 'checked' : '' ; ?>
34 Estudiante
35 </label>
36 <br>
37 <label>
38 <input type="radio" name="idcargo" value="3" <?php echo ($idcargo == 3) ? 'checked' : '' ; ?>
39 Docente
40 </label>
41
42
43 <br>
44 <br>
45
46 <h3 style="color: #4fa570;"><em><u> Estado de usuario </u></em></h3>
47
48 <label>
49 <input type="radio" name="id_estado" value="1" <?php echo ($id_estado == 1) ? 'checked' : '' ; ?>
50 Activo
51 </label>
52 <br>
53 <label>
54 <input type="radio" name="id_estado" value="2" <?php echo ($id_estado == 2) ? 'checked' : '' ; ?>
55 Inactivo
56 </label>
57 </div>
58
59
60 </fieldset>
61 <hr>
62 <input type="hidden" name="id" value="<?php echo $id ; ?>">
63 <button type="submit" class="btn btn-primary" name="enviar">Guardar</button>
64 </form>

```

Fuente: Los autores.

Características del formulario:

- **Precarga de Datos:** Los campos se llenan automáticamente con la información actual del usuario.
- **Edición Segura:** Validaciones para evitar errores, como contraseñas vacías.
- **Selección de Rol y Estado:** Uso de botones de radio para facilitar la selección.

Backend: Controlador para Editar Usuario

El archivo `editar_usuario.php` gestiona la carga, validación y actualización de datos en la base de datos.

***Ilustración 27:** Código del controlador para editar usuario*

```

1  <?php
2  include "../conexion.php";
3
4  if (isset($_POST['enviar'])) {
5  $id = $_POST['id'];
6  $documento = $_POST["documento"];
7  $nombre = $_POST['nombre'];
8  $usuario = $_POST['usuario'];
9  $contraseña = $_POST['contraseña'];
10 $idcargo = $_POST['idcargo']; // Agregado
11 $id_estado = $_POST['id_estado']; // Agregado
12 $hash_contrañesa = password_hash($contraseña, PASSWORD_DEFAULT);
13
14 $sql = "UPDATE usuarios
15 SET documento = '$documento', nombre = '$nombre', usuario = '$usuario', contraseña = '$hash_contrañesa', idcargo = '$idcargo', id_estado = '$id_estado'
16 WHERE id = '$id'";
17
18 if ($conn->query($sql) === TRUE) {
19     echo '<script>
20 alert("Se actualizaron los datos");
21 location.assign("../admin/gestion_usuarios");
22 </script>';
23 } else {
24     echo '<script>
25 alert("Error al actualizar los datos: ' . $conn->error . '");
26 location.assign("../admin/gestion_usuarios");
27 </script>';
28 }
29 } else {
30 $id = $_GET['id'];
31 $sql = "SELECT * FROM usuarios WHERE id = '$id'";
32 $resultado = $conn->query($sql);
33
34 if ($resultado) {
35     $row = $resultado->fetch_array();
36     $documento = $row["documento"];
37     $nombre = $row["nombre"];
38     $usuario = $row["usuario"];
39     $contraseña = $row["contraseña"];
40     $idcargo = $row["idcargo"]; // Agregado
41     $id_estado = $row["id_estado"]; // Agregado
42 }
43
44 mysqli_close($conn);
45 ?>

```

Fuente: los autores.

Pasos principales en el backend:

1. **Precarga de Datos:** Se obtienen los datos actuales del usuario desde la base de datos.
2. **Captura de Nuevos Datos:** Los datos enviados mediante \$_POST se reciben para procesarlos.
3. **Cifrado de Contraseña:** Si la contraseña es nueva, se cifra antes de actualizarla.

4. **Actualización de la Base de Datos:** Se ejecuta una consulta SQL para guardar los cambios.
5. **Gestión de Errores:** Se muestra un mensaje en caso de fallos.

4. Validaciones de Seguridad

1. **Precarga Controlada:** Solo se cargan los datos del usuario seleccionado mediante su ID.
2. **Contraseñas Hasheadas:** Las nuevas contraseñas se cifran utilizando password_hash.
3. **Protección Contra Errores:** Validación de todos los campos antes de realizar la actualización.

5. Roles y Estado del Usuario

El administrador puede modificar tanto el rol como el estado del usuario en este módulo.

6. Casos de Uso Documentados

- **Edición Exitosa:** El sistema muestra un mensaje de éxito y redirige al administrador.
- **Error en la Actualización:** Si ocurre un error, se informa al administrador con detalles del problema.

- **Contraseña Inalterada:** Si no se introduce una nueva contraseña, se mantiene la actual.

Crear curso:

El módulo "Crear Curso" permite a los administradores agregar nuevos cursos al sistema.

Este proceso incluye la especificación de la descripción y el semestre al que pertenece.

1. Descripción de la Funcionalidad

El administrador puede registrar un nuevo curso proporcionando los siguientes datos:

- **Descripción del curso:** Breve texto descriptivo.
- **Semestre:** Número de semestre al que pertenece el curso.

2. Flujo de Funcionamiento

1. **Ingreso de Datos:** El administrador completa el formulario con la descripción y semestre.
2. **Almacenamiento en Base de Datos:** Los datos del curso se guardan en la base de datos.
3. **Notificación:** El sistema informa si el curso fue creado exitosamente o si ocurrió algún error.

3. Estructura del Prototipo

Frontend: Formulario para Crear Curso

El formulario recopila los datos necesarios para registrar un nuevo curso. Incluye validaciones básicas para asegurar que se completen todos los campos requeridos.

Ilustración 28: Código del formulario para crear curso

```
1 <form id="createCourseForm" method="POST" enctype="multipart/form-data">
2   <div class="mb-3">
3     <label for="descripcion" class="form-label">Descripción del Curso:</label>
4     <input type="text" class="form-control" id="descripcion" name="descripcion" required>
5   </div>
6
7   <div class="mb-3">
8     <label for="semestre" class="form-label">Semestre:</label>
9     <select class="form-control" id="semestre" name="semestre" required>
10      <option value="">Seleccione un semestre</option>
11      <?php for ($i = 1; $i <= 10; $i++): ?>
12        <option value="<?php echo $i; ?>"><?php echo $i; ?></option>
13      <?php endfor; ?>
14    </select>
15  </div>
16
17  <div class="mb-3">
18    <label for="imagen" class="form-label">Cargar imagen:</label>
19    <input type="file" class="form-control" id="imagen" name="imagen" accept="image/*" required>
20  </div>
21
22  <button type="submit" class="btn btn-primary">Agregar Curso</button>
23  <a href="index_course.php" class="btn btn-secondary">Regresar a Gestión de Cursos</a>
24 </form>
```

Fuente: Los autores.

Características del formulario:

- **Validación de Campos:** Todos los campos son obligatorios.
- **Selector de Semestre:** Incluye un menú desplegable para seleccionar el semestre.

Backend: Controlador para Crear Curso

El archivo crear_curso.php gestiona la lógica para validar y almacenar los datos del curso.

Ilustración 29: Código del controlador para crear curso

```
1  <?php
2  if (session_status() == PHP_SESSION_NONE) {
3      session_start();
4  }
5  include "../conexion.php";
6
7  $response = ['status' => '', 'message' => ''];
8
9  if ($_SERVER['REQUEST_METHOD'] == 'POST') {
10     $descripcion = $_POST['descripcion'];
11     $semestre = $_POST['semestre'];
12
13     $imagen = $_FILES['imagen'];
14     $ruta_imagen = 'img_courses/' . basename($imagen['name']);
15
16     if (move_uploaded_file($imagen['tmp_name'], $ruta_imagen)) {
17         $consulta = "INSERT INTO courses (descripcion, imagen, semestre) VALUES (?, ?, ?)";
18         $stmt = $conn->prepare($consulta);
19         $stmt->bind_param("sss", $descripcion, $ruta_imagen, $semestre);
20
21         if ($stmt->execute()) {
22             $response['status'] = 'success';
23             $response['message'] = 'Curso guardado correctamente.';
24         } else {
25             $response['status'] = 'error';
26             $response['message'] = "Error: " . $stmt->error;
27         }
28     } else {
29         $response['status'] = 'error';
30         $response['message'] = "Error al cargar la imagen.";
31     }
32
33     echo json_encode($response);
34     exit();
35 }
36 ?>
```

Fuente: Los autores.

Pasos principales en el backend:

1. **Validación de Datos:** Verifica que todos los campos estén completos.
2. **Inserción en la Base de Datos:** Usa una consulta preparada para guardar la descripción y semestre.
3. **Gestión de Errores:** Proporciona mensajes detallados si ocurre algún problema.

4. Validaciones de Seguridad

1. **Consultas Preparadas:** Previenen ataques de inyección SQL al almacenar datos.
2. **Validación del Semestre:** Asegura que el semestre sea un valor dentro del rango permitido.

5. Campos del Curso

Tabla 16: Campos del Curso

Campo	Descripción
Descripción	Nombre o descripción del curso.
Semestre	Número del semestre del curso.

Fuente: Los autores.

6. Casos de Uso Documentados

- **Curso Creado Exitosamente:** El sistema guarda el curso en la base de datos y muestra un mensaje de éxito.
- **Error en Base de Datos:** Se muestra un mensaje de error detallado si falla la inserción en la base de datos.

Gestión de cursos:

1. Descripción de la Gestión de Cursos

La gestión de cursos permite visualizar, editar y eliminar cursos existentes en el sistema.

Se implementaron tanto el frontend como el backend, asegurando la funcionalidad de listar los cursos registrados y ofrecer opciones para su administración.

2. Flujo de Funcionamiento

- **Visualización de cursos:**

Los cursos registrados en la base de datos se muestran en un formato estructurado, indicando el semestre, descripción, y con opciones para administrar cada curso.

- **Edición de cursos:**

Redirige al formulario correspondiente para modificar los datos de un curso específico.

- **Eliminación de cursos:**

Permite eliminar un curso seleccionado tras la confirmación del usuario.

3. Estructura del Prototipo

Frontend: Interfaz de Gestión de Cursos

El diseño de la interfaz presenta una cuadrícula de tarjetas que muestran los cursos registrados. Cada tarjeta incluye las opciones de visualizar, editar y eliminar.

Ilustración 30: Código de la pantalla “Gestión de cursos”

```
1 <div class="row">
2 <?php if (!empty($courses)): ?>
3 <?php foreach ($courses as $course): ?>
4 <div class="col-xxl-4 col-xl-4 col-lg-4 col-md-6">
5 <div class="course_item white-bg mb-30 fix">
6 <div class="course_thumb w-img p-relative fix">
7 <a href=".."calculo_course.php">
8 <?php if (empty($course["imagen"])): ?>
9 <p>No tiene imagen</p>
10 <?php else: ?>
11 " alt="">
12 <?php endif; ?>
13 </a>
14 <div class="course_tag">
15 <a href=".."calculo_course.php"><?php echo htmlspecialchars($course["semestre"]); ?> Semestre</a>
16 </div>
17 </div>
18 <div class="course_content">
19 <h3 class="course_title">
20 <a href=".."calculo_course.php"><?php echo htmlspecialchars($course["descripcion"]); ?></a>
21 </h3>
22 <div class="course_teacher d-flex align-items-center">
23 <div class="course_teacher-thumb mr-15">
24 
25 </div>
26 <h6><a href="#"></a>Aquí va el nombre del docente que proxiamamente se asignara</h6>
27 </div>
28 </div>
29 </div>
30 <div class="course_more d-flex justify-content-between align-items-center">
31 <div class="course_btn">
32 <a href="show_course.php?code=<?php echo $course['code']; ?>" class="btn btn-sm btn-primary">
33 <i class="fas fa-eye"></i>
34 </a>
35 <a href="edit_course.php?code=<?php echo $course['code']; ?>" class="btn btn-sm btn-warning">
36 <i class="fas fa-edit"></i>
37 </a>
38 <button class="btn btn-sm btn-danger delete-course" data-code="<?php echo $course['code']; ?>">
39 <i class="fas fa-trash-alt"></i>
40 </button>
41 </div>
42 </div>
43 </div>
44 </div>
45 <?php endforeach; ?>
46 <?php else: ?>
47 <style>
48 body{
49 background-color: green;
50 }
51 </style>
52 <p style="color: black;" class="h5">No se encontraron cursos.</p>
53 <?php endif; ?>
54 </div>
```

Fuente: Los autores.

Backend: Consulta de Cursos

La lógica backend se encarga de obtener los cursos almacenados en la base de datos y pasarlos al frontend para su representación.

Ilustración 31: Código de la lógica de la consulta de cursos.

```

1  <?php
2  session_start();
3  //mensajes
4
5  $usuario = $_SESSION['usuario'] ?? null;
6  if (!isset($usuario)) {
7      header("Location: login.html");
8      exit();
9  }
10
11 // Incluir el archivo de conexión
12 require_once('../conexion.php');
13
14 // Verificar la conexión
15 if ($conn->connect_error) {
16     die("Connection failed: " . $conn->connect_error);
17 }
18
19 // Consulta para obtener los cursos
20 $query = "SELECT * FROM courses";
21 $result = mysqli_query($conn, $query);
22
23 $courses = [];
24 if ($result && mysqli_num_rows($result) > 0) {
25     while ($row = mysqli_fetch_assoc($result)) {
26         $courses[] = $row;
27     }
28 }
29 ?>

```

Fuente: Los autores.

4. Funcionalidades Principales

Visualización de Cursos

Los cursos son mostrados con la siguiente estructura:

- Descripción.
- Semestre correspondiente.
- Opciones para administrar el curso:
 - **Visualizar:** Redirige a la vista detallada del curso.
 - **Editar:** Redirige al formulario para modificar el curso.
 - **Eliminar:** Inicia la acción de eliminación.

Listado de Cursos Vacío

Si no hay cursos registrados, se muestra un mensaje indicando que no hay cursos disponibles.

5. Validaciones de Seguridad

- **Control de Acceso:**

Los usuarios deben estar autenticados para acceder a la gestión de cursos. Si no tienen sesión activa, se les redirige al login.

- **Consulta Segura:**

Las consultas a la base de datos están diseñadas para prevenir errores y manejar la falta de resultados.

6. Casos de Uso Documentados

- **Lista de cursos vacía:** Se muestra un mensaje: *"No se encontraron cursos"*.
- **Edición de curso:** El curso seleccionado redirige al formulario de edición.
- **Eliminación de curso:** El curso seleccionado es eliminado de la base de datos tras confirmación.

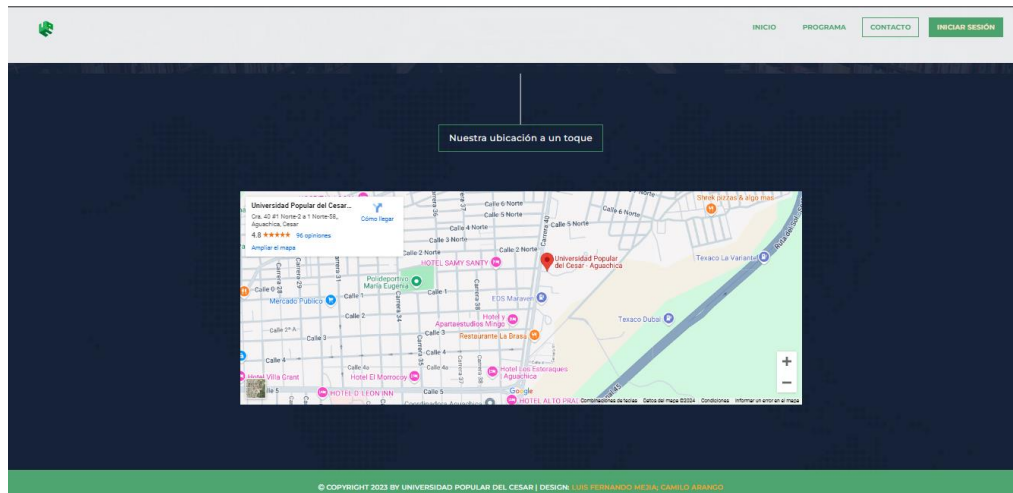
4.3.3 sacar versión inicial del frontend con funcionalidad básica

Índex:

Descripción: Esta es la pantalla de bienvenida o página principal del sistema. Aquí los usuarios pueden acceder a distintas secciones según su rol, como "Login", "Registro", o "Información de la plataforma". Sirve como punto de entrada para estudiantes, docentes y administradores.

Ilustración 32: Captura de la pantalla "índex"



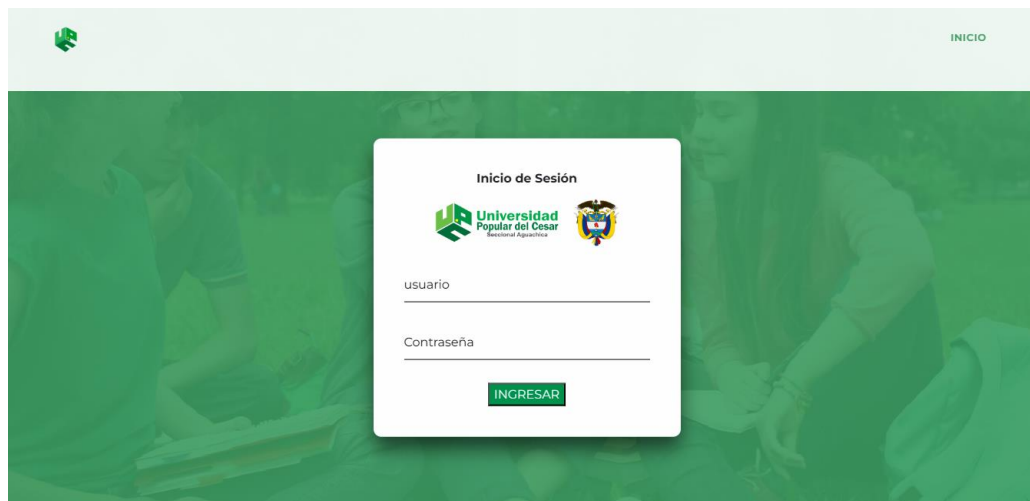


Fuente: Los autores.

Login

Descripción: La pantalla de inicio de sesión permite a los usuarios autenticarse en el sistema utilizando sus credenciales. Incluye campos para el nombre de usuario y la contraseña, y un botón de inicio de sesión. La interfaz fue diseñada para ser sencilla e intuitiva, facilitando el acceso al sistema.

Ilustración 33: Captura de la pantalla “login”.



Fuente: Los autores.

Vista principal admin

Descripción: Una vez autenticado, el administrador accede a esta vista principal o tablero de control, donde puede gestionar usuarios, cursos, y configuraciones. Esta vista muestra una estructura clara de opciones y funcionalidades específicas para el rol de administrador, incluyendo accesos directos a tareas críticas.

Ilustración 34: Captura de la pantalla “vista principal admin”.

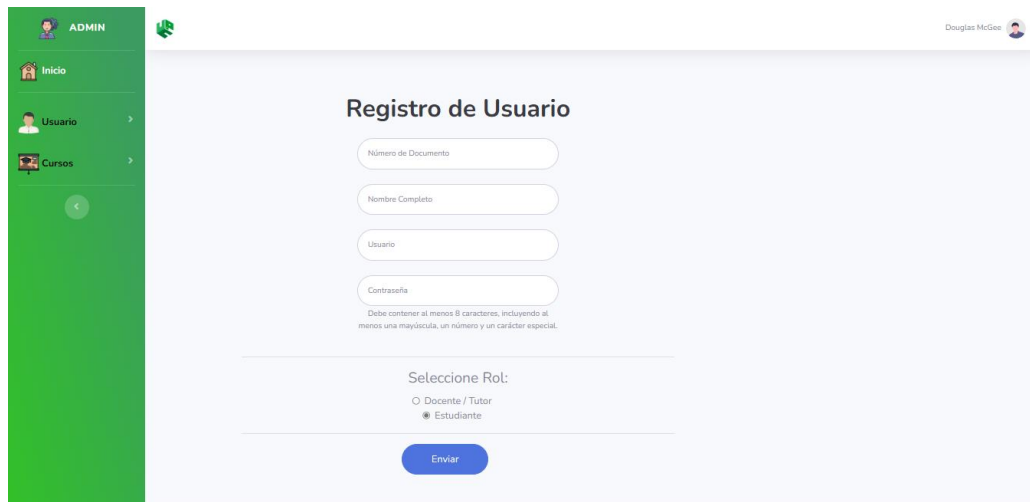
Documento	Nombre	Usuario	Rol	Estado
123123	Luis Fernando	LFernando	Docente	Activo
145265	docenteprueba	docenteprueba	Docente	Activo
232425	pruebaestudiante	pruebaestudiante	Estudiante	Activo
100237649	Laura Gomez	LAura	Estudiante	Activo
100246789	Johan Caamano	Johannn	Estudiante	Inactivo
109289309	Natalia Campo	NCampo	Estudiante	Activo
257892093	David del Valle	DDelvalle	Estudiante	Activo
1002465117	docente	docente	Docente	Activo

Fuente: Los autores.

Admin/ Agregar usuario

Descripción: En esta pantalla, el administrador puede agregar nuevos usuarios al sistema. Incluye campos como nombre, rol, y botones para guardar o cancelar. La interfaz está diseñada para guiar al administrador en la entrada de datos y asegurar la integridad de la información.

Ilustración 35: Título: Captura de la pantalla “agregar usuario”.



The screenshot shows a web application interface for user registration. On the left is a green sidebar with navigation options: 'ADMIN' (with a user icon), 'Inicio' (with a house icon), 'Usuario' (with a person icon), and 'Cursos' (with a book icon). The main content area is titled 'Registro de Usuario' and contains the following elements:

- Four input fields: 'Número de Documento', 'Nombre Completo', 'Usuario', and 'Contraseña'.
- A password requirement note: 'Debe contener al menos 8 caracteres, incluyendo al menos una mayúscula, un número y un carácter especial.'
- A section titled 'Seleccione Rol:' with two radio button options: 'Docente / Tutor' and 'Estudiante' (which is selected).
- An 'Enviar' button at the bottom.

The top right corner of the page shows the user 'Douglas McGee' with a profile icon.

Fuente: Los autores.

Admin/Modificar usuario

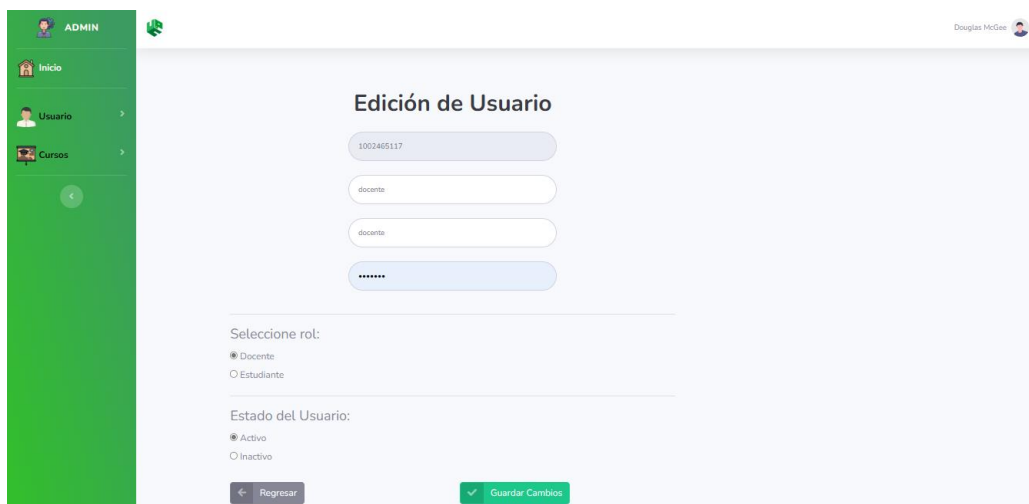
Descripción: Esta pantalla permite al administrador editar y actualizar los datos de un usuario registrado en el sistema. La funcionalidad incluye la modificación de datos clave como el número de documento, nombre, nombre de usuario, y contraseña. Además, se pueden actualizar atributos específicos como el cargo del usuario (Estudiante o Docente) y su estado (Activo o Inactivo).

La pantalla presenta los datos actuales del usuario en campos de entrada, que pueden ser editados directamente. En el caso de la contraseña, el sistema permite introducir una nueva contraseña, la cual se encripta antes de ser almacenada para mayor seguridad. La interfaz también incluye opciones de selección de cargo y estado en formato de botones de radio, lo cual simplifica la selección de estas categorías.

Una vez realizadas las modificaciones, el administrador puede guardar los cambios con el botón Guardar, y si la actualización es exitosa, se muestra un mensaje de confirmación. En caso de error, se muestra una alerta detallando el problema.

Al finalizar, el administrador tiene la opción de regresar a la página principal de gestión de usuarios.

Ilustración 36: Captura de la pantalla “Modificar usuario”



Fuente: Los autores.

Admin/ Gestión de usuario

Descripción: Esta pantalla permite al administrador visualizar, gestionar y editar los datos de los usuarios registrados en el sistema, la vista está diseñada para mostrar información clave de cada usuario en una tabla, facilitando una administración más eficiente.

La tabla incluye columnas para:

Documento: el número de documento de identificación del usuario.

Nombre: el nombre completo del usuario.

Usuario: el nombre de usuario asignado.

Rol: indica el tipo de usuario, ya sea "Estudiante" o "Docente".

Estado: muestra si el usuario está "Activo" o "Inactivo" en el sistema.

Detalles: un botón de acción que redirige a una página de edición de usuario, donde se pueden actualizar los datos específicos del usuario seleccionado.

La tabla permite una fácil navegación y revisión de usuarios gracias a su diseño responsivo, y está integrada con el sistema de base de datos para cargar los datos en tiempo real. Si no hay usuarios disponibles, se muestra un mensaje en la tabla informando al administrador de esta situación.

Componentes adicionales:

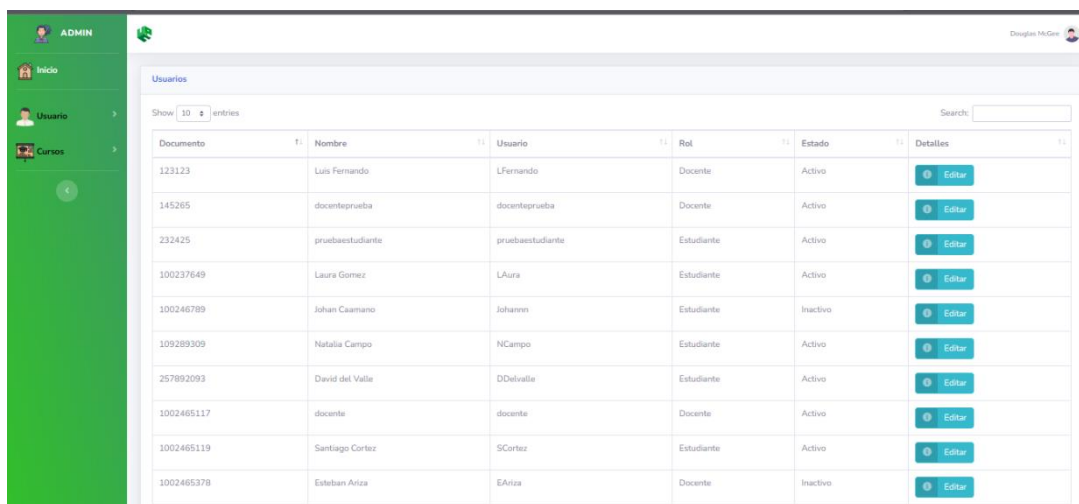
Sidebar: incluye el menú de navegación lateral para acceder rápidamente a otras áreas de administración.

Topbar: la barra superior que muestra el nombre del administrador y permite cerrar sesión.

Footer: incluye los derechos de autor y una referencia institucional a la universidad.

Esta vista es fundamental para que el administrador mantenga un control preciso de los usuarios activos en el sistema y realice ajustes en los datos cuando sea necesario.

Ilustración 37: Captura de la pantalla “Gestión de usuarios”



Documento	Nombre	Usuario	Rol	Estado	Detalles
123123	Luis Fernando	LFernando	Docente	Activo	Editar
145285	docenteprueba	docenteprueba	Docente	Activo	Editar
232425	pruebaestudiante	pruebaestudiante	Estudiante	Activo	Editar
100237649	Laura Gomez	LAura	Estudiante	Activo	Editar
100246789	Johan Caamano	Johann	Estudiante	Inactivo	Editar
109289309	Natalia Campo	NCampo	Estudiante	Activo	Editar
257892093	David del Valle	DDelvalle	Estudiante	Activo	Editar
1002465117	docente	docente	Docente	Activo	Editar
1002465119	Santiago Cortez	SCortez	Estudiante	Activo	Editar
1002465378	Esteban Ariza	EAriza	Docente	Inactivo	Editar

Fuente: Los autores

Admin/Gestionar cursos

Descripción: La pantalla de gestión de cursos permite al administrador visualizar todos los cursos registrados, así como realizar acciones específicas sobre cada curso, como editar, ver información detallada o eliminar el curso. Cada curso está representado en una tarjeta individual que proporciona información básica y opciones de administración rápidas.

Detalles de cada tarjeta de curso:

- **Nombre del curso:** el nombre o descripción del curso, que se muestra en la parte superior de cada tarjeta.
- **Docente asignado:** el nombre del docente que se ha asignado al curso. Si no hay un docente asignado, se muestra una entrada predeterminada.
- **Estado:** indica si el curso está "Activo" o "Inactivo" en el sistema. Este estado se muestra con texto en color verde si el curso está activo y en rojo si está inactivo, proporcionando una identificación visual rápida.

Acciones disponibles:

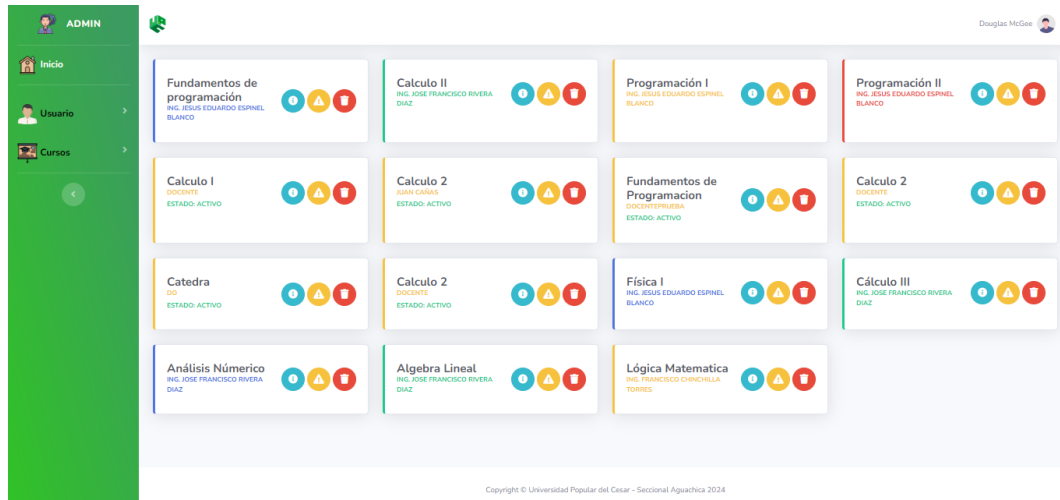
- **Información:** el botón con el ícono de información permite ver detalles adicionales del curso.
- **Editar:** el botón con el ícono de advertencia permite acceder a la vista de edición, donde se pueden modificar datos específicos del curso.
- **Eliminar:** el botón con el ícono de papelera permite eliminar el curso de la base de datos. Este botón está resaltado en rojo para indicar su función de eliminación.

Componentes adicionales:

- **Sidebar:** incluye el menú lateral de navegación, permitiendo al administrador acceder a otras secciones.
- **Topbar:** la barra superior, que muestra la sesión del usuario y la opción para cerrar sesión.
- **Footer:** incluye los derechos de autor y hace referencia a la universidad.

Esta vista centraliza el proceso de gestión de cursos, facilitando la administración y permitiendo acciones rápidas sobre cada curso registrado.

Ilustración 38: Captura de la pantalla “gestión de cursos”.



Fuente: Los autores.

Gestión cursos individuales

Descripción

1. Obtención del Código del Curso:

- El código del curso se extrae a través de los parámetros GET. Si no se proporciona, se muestra un mensaje de error.

2. Cambio de Estado del Curso:

- Si se recibe un parámetro estado, el sistema actualiza el estado del curso en la base de datos (1 para activo, 2 para inactivo).

- Tras la actualización, se despliega una notificación de éxito o error usando la biblioteca SweetAlert2.

3. Visualización del Estado Actual del Curso:

- Mediante una consulta, se obtiene el estado actual del curso y se muestra en una tarjeta.

4. Lista de Estudiantes Matriculados:

- Se consulta la base de datos para obtener los estudiantes matriculados en el curso.
- Si hay estudiantes inscritos, se presenta una tabla con sus nombres, nombres de usuario y estado. Si no hay estudiantes matriculados, se indica esta situación.

5. Botón para Matricular Estudiantes:

- Un botón dirige al usuario a una página para matricular estudiantes en el curso.

6. Visualización y Asignación del Docente:

- Una sección muestra el docente asignado al curso; si no hay docente asignado, se indica también.

- Incluye un botón para asignar o cambiar el docente, que abre un modal con la lista de docentes disponibles, indicando los que ya están asignados a otros cursos.

7. Modal para Asignar/Cambiar Docente:

- Al abrir el modal, se muestra un formulario con los docentes disponibles como opciones de selección.
- El docente actual está preseleccionado, y el administrador puede asignar otro docente si lo desea.

8. Cambio de Estado del Curso:

- Una tarjeta permite al administrador cambiar el estado del curso entre activo e inactivo, según su estado actual.

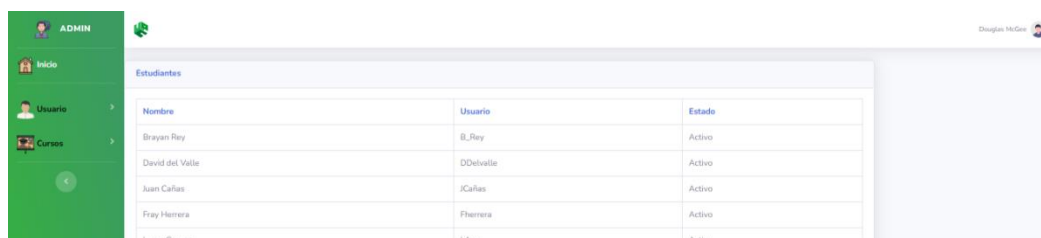
9. Botón para Regresar:

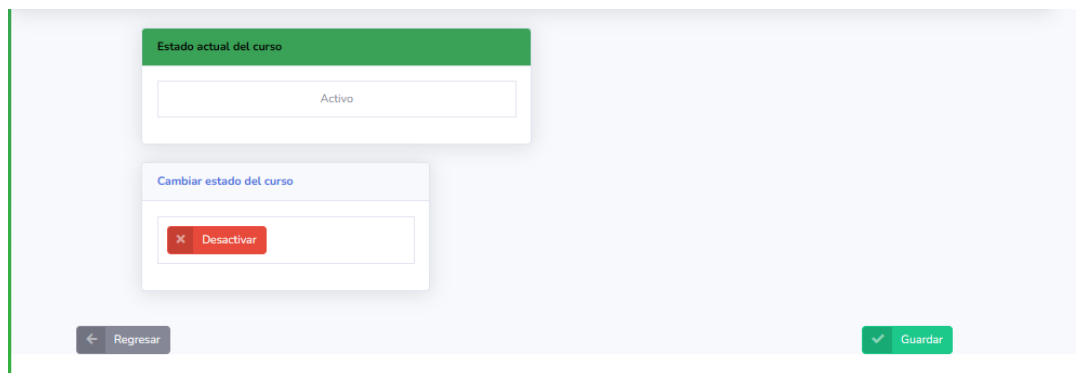
- Incluye un botón de regreso a la vista general de gestión de cursos.

10. Estilos Adicionales:

- Se define un estilo específico para el modal, optimizando el contraste de colores para mejorar la legibilidad.

Ilustración 39: Captura de la pantalla “gestión de un curso en específico”





Fuente: Los autores.

Index Estudiante:

El índice del estudiante es una página de bienvenida diseñada para proporcionar un espacio introductorio en el portal académico. Al ingresar, los estudiantes son recibidos con un mensaje de bienvenida destacando su rol dentro de la plataforma. La página está diseñada para ser visualmente amigable, con un encabezado claro que da la bienvenida al estudiante y un pequeño texto de apoyo resaltando el propósito de la página: ofrecer recursos académicos.

Además, la interfaz es sencilla, con un formulario de usuario y un pie de página que muestra los derechos de autor de la universidad. Es una página que sirve como punto de inicio para acceder a otras secciones del portal, facilitando el acceso a información relevante sobre su vida académica. En resumen, el índice del estudiante es una puerta de entrada con un diseño limpio y accesible que invita a explorar más sobre el contenido y los servicios disponibles en la plataforma.

Ilustración 40: Captura de la pantalla “índice estudiante”.

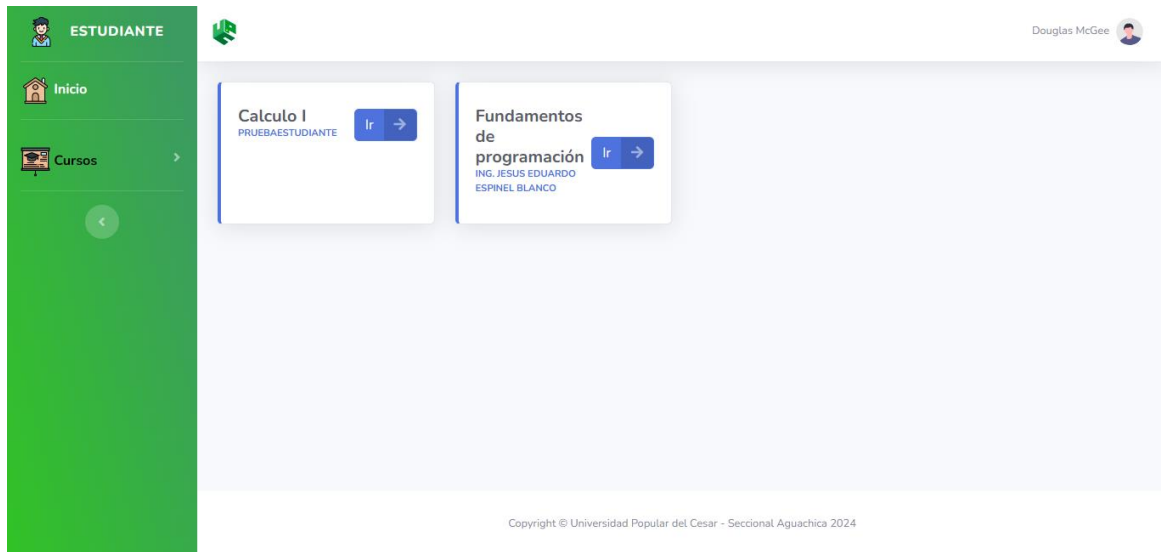


Fuente: Los autores.

Estudiante/cursos:

Esta página muestra los cursos en los que el estudiante está matriculado, permitiendo una visualización detallada y un acceso rápido a la programación de cada uno. La interfaz está estructurada con tarjetas para cada curso, donde se muestra la descripción del curso, el semestre y el nombre del docente asignado.

Ilustración 41: Captura de la pantalla “cursos de un estudiante”



Fuente: Los autores.

Detalle/curso

Descripción: Al entrar a esta página, el estudiante es recibido con un mensaje de bienvenida y se le presenta un breve resumen sobre el contenido del curso. Cada corte, representado en tarjetas colapsables, puede expandirse para revelar los temas, actividades y materiales correspondientes. Cada tarjeta incluye un encabezado interactivo que muestra y oculta el contenido con un clic, facilitando la navegación.

La interfaz ofrece una organización clara por cortes (Primer, Segundo, y Tercer Corte) para que el estudiante pueda encontrar rápidamente el material de estudio adecuado. Finalmente, hay un botón que permite regresar a la lista de cursos, manteniendo la navegación intuitiva y accesible para el estudiante.

Ilustración 42: Captura de la pantalla “Detalle del curso”.



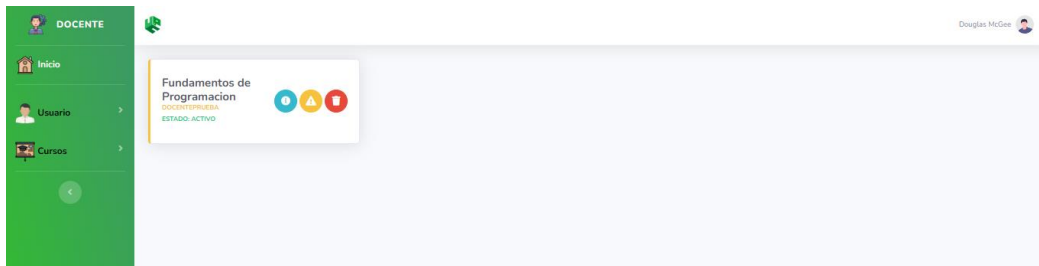
Fuente: Los autores.

Docente/índex

Descripción: Esta página permite a los docentes visualizar y gestionar de forma rápida los cursos que tienen asignados. Cada curso se presenta en una tarjeta que incluye detalles importantes como el nombre, descripción, semestre, estado (activo o inactivo) y

el nombre del docente. Además, cada tarjeta ofrece botones de acción rápida para ver información detallada, editar o eliminar el curso. La información es obtenida de la base de datos y se filtra para mostrar solo los cursos del docente que ha iniciado sesión. En caso de no tener cursos asignados, la página muestra un mensaje informativo al usuario.

Ilustración 43: Captura de la pantalla “índice docente”.



Fuente: Los autores.

Docente/ gestión curso en específico

Descripción:

En esta página, el docente puede ver la lista de estudiantes matriculados en el curso seleccionado, con información detallada de cada estudiante, como nombre, usuario y estado (activo o inactivo). Al final de la sección, se muestra un botón para matricular nuevos estudiantes, que permite al docente agregar más alumnos al curso directamente desde esta interfaz.

Ilustración 44: Captura de la pantalla “gestión de un curso del docente”.

Nombre	Usuario	Estado
Brayan Rey	B_Rey	Activo
David del Valle	DDeValle	Activo
Fray Herrera	Fherrera	Activo
Jose Mejia	Jmeja	Activo
Juan Cañas	JCañas	Activo

Matricula

Matricular estudiante

Fuente: Los autores.

4.3.4 Codificación del backend

Lógica del login

El LoginController gestiona el proceso de autenticación de usuarios en el sistema. Valida los datos ingresados, verifica que el usuario esté activo y comprueba la contraseña utilizando una función de encriptación segura. Dependiendo del rol del usuario (administrador, estudiante o docente), lo redirige a su sección correspondiente dentro del sistema. Además, el controlador utiliza mensajes de error visuales para notificar al usuario en caso de credenciales incorrectas, usuarios inactivos o problemas en la consulta, proporcionando una experiencia de inicio de sesión segura y orientada al usuario.

Ilustración 45: Código de la lógica del login

```

1  <?php
2  session_start();
3
4  // Conexión a la base de datos
5  require_once "conexion.php";
6
7  // Capturar y sanitizar datos
8  $usuario = isset($_POST['usuario']) ? mysqli_real_escape_string($conn, $_POST['usuario']) : '';
9  $contraseña = isset($_POST['contraseña']) ? $_POST['contraseña'] : '';
10
11 if (empty($usuario) || empty($contraseña)) {
12     mostrarError("Todos los campos son requeridos");
13     exit();
14 }
15
16 // Usar consultas preparadas para prevenir SQL Injection
17 $consulta = "SELECT * FROM usuarios WHERE usuario = ?";
18 $stmt = mysqli_prepare($conn, $consulta);
19 mysqli_stmt_bind_param($stmt, "s", $usuario);
20 mysqli_stmt_execute($stmt);
21 $resultado = mysqli_stmt_get_result($stmt);
22
23 if ($resultado) {
24     $fila = mysqli_fetch_assoc($resultado);
25
26     if ($fila) {
27         if ($fila['id_estado'] == 2) {
28             mostrarError("Usuario Inactivo", "No tienes permisos para acceder. Este usuario está inactivo.");
29             exit();
30         }
31
32         if (password_verify($contraseña, $fila['contraseña'])) {
33             // Guardar el id del usuario en la sesión
34             $_SESSION['usuario_id'] = $fila['id'];
35             $_SESSION['usuario'] = $usuario;
36             $_SESSION['idcargo'] = $fila['idcargo'];
37             $_SESSION['id_estado'] = $fila['id_estado'];
38
39             // Redirigir según el cargo
40             switch ($fila['idcargo']) {
41                 case 1:
42                     header("Location: under/index_admin.php");
43                     break;
44                 case 2:
45                     header("Location: under/index_student.php");
46                     break;
47                 case 3:
48                     header("Location: under/gestion_cursos_teacher.php");
49                     break;
50                 default:
51                     mostrarError("Error", "Tipo de usuario no válido");
52                     exit();
53             }
54         } else {
55             mostrarError("Error", "Contraseña incorrecta");
56         }
57     } else {
58         mostrarError("Error", "Usuario no encontrado");
59     }
60 } else {
61     mostrarError("Error", "Error en la consulta: " . mysqli_error($conn));
62 }
63
64 mysqli_stmt_close($stmt);
65 mysqli_close($conn);
66
67 function mostrarError($titulo, $mensaje = "") {
68     echo '<!DOCTYPE html>';
69     <html>
70     <head>
71         <meta charset="UTF-8">
72         <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
73     </head>
74     <body>
75         <script>
76             Swal.fire({
77                 icon: "error",
78                 title: "" . $titulo . "",
79                 text: "" . $mensaje . "",
80                 allowOutsideClick: false
81             }).then((result) => {
82                 window.location.href = "login.html";
83             });
84         </script>
85     </body>
86     </html>';
87 }
88 ?>
89

```

Fuente: Los autores.

Admin/ Listar usuarios

Este backend genera una lista de estudiantes y docentes mostrando información clave en una tabla HTML. Primero, se conecta a la base de datos y consulta todos los usuarios que tienen roles de estudiante o docente, filtrando con idcargo (2 para estudiantes y 3 para docentes). Luego, para cada registro, se determina y muestra el rol (Estudiante o Docente) y el estado (Activo o Inactivo) de cada usuario. Los datos se despliegan en una tabla dentro de una interfaz visualmente estructurada, usando el componente DataTable para la presentación y funcionalidad de la tabla. Al finalizar, el código cierra la conexión a la base de datos para liberar recursos.

Ilustración 46: Código de la lógica de listar usuarios.

```

1  <?php
2  include "../conexion.php"; // Asegurate de que tu conexión esté configurada correctamente
3
4  // Realiza la consulta para obtener todos los usuarios
5  $sql = "SELECT documento, nombre, usuario, idcargo, id_estado FROM usuarios";
6  $resultado = $conn->query($sql);
7  ?>
8
9  <div class="container-fluid">
10 <div class="card shadow mb-4">
11 <div class="card-header py-3">
12 <h6 class="m-0 font-weight-bold text-primary">Estudiantes</h6>
13 </div>
14 <div class="card-body">
15 <div class="table-responsive">
16 <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
17 <thead>
18 <tr>
19 <th>Documento</th>
20 <th>Nombre</th>
21 <th>Usuario</th>
22 <th>Rol</th>
23 <th>Estado</th>
24 </tr>
25 </thead>
26 <tbody>
27 <?php
28 // Realizar la consulta para obtener solo estudiantes y docentes
29 $consulta = "SELECT documento, nombre, usuario, idcargo, id_estado FROM usuarios WHERE idcargo IN (2, 3)";
30 $resultado = $conn->query($consulta);
31
32 if ($resultado->num_rows > 0) {
33     while ($fila = $resultado->fetch_assoc()) {
34         // Determinar el rol basado en idcargo
35         $rol = ($fila["idcargo"] == 2) ? "Estudiante" : "Docente";
36         // Determinar el estado basado en id_estado
37         $estado = ($fila["id_estado"] == 1) ? "Activo" : "Inactivo";
38         echo "<tr>
39             <td>{$fila['documento']}</td>
40             <td>{$fila['nombre']}</td>
41             <td>{$fila['usuario']}</td>
42             <td>{$rol}</td>
43             <td>{$estado}</td>
44         </tr>";
45     }
46 } else {
47     echo "<tr><td colspan='5'>No hay datos disponibles</td></tr>";
48 }
49 ?>
50 </tbody>
51 </table>
52 </div>
53 </div>
54 </div>
55 </div>
56 </div>
57
58 <?php
59 $conn->close(); // Cierra la conexión a la base de datos
60 ?>

```

Fuente: Los autores

Admin/ Agregar usuario

Este backend gestiona la funcionalidad para agregar un nuevo estudiante (o usuario) a la base de datos, validando la información proporcionada y mostrando mensajes de éxito o error. Primero, captura los datos del formulario (documento, nombre, usuario, contraseña, id_cargo) y los limpia para evitar inyecciones de SQL. A continuación, verifica

si el documento ya está registrado en la base de datos para prevenir duplicados. Si el documento existe, muestra una advertencia y redirige al formulario de registro.

Si el documento es único, el código encripta la contraseña usando `password_hash` para mejorar la seguridad y procede a insertar los datos en la base de datos. Al finalizar, muestra un mensaje de éxito si la inserción fue exitosa, o un mensaje de error si ocurrió algún problema durante la inserción. Para los mensajes de alerta y redirección, utiliza `SweetAlert` para mejorar la experiencia del usuario. Finalmente, el código cierra las conexiones a la base de datos para liberar recursos.

Ilustración 47: Código de la lógica de “agregar usuario”.

```

1 <?php
2 include "../conexion.php";
3 if ($_SERVER["REQUEST_METHOD"] == "POST") {
4     // Capturar los datos del formulario y limpiarlos para evitar inyecciones
5     $documento = mysql_real_escape_string($conn, $_POST["documento"]);
6     $nombre = mysql_real_escape_string($conn, $_POST["nombre"]);
7     $usuario = mysql_real_escape_string($conn, $_POST["usuario"]);
8     $contraseña = $_POST["contraseña"];
9     $id_cargo = mysql_real_escape_string($conn, $_POST["id_cargo"]);
10    $id_estado = 1;
11
12    // Consultar si existe el documento
13    $consulta_documento = "SELECT * FROM usuarios WHERE documento = ?";
14    $stmt = $conn->prepare($consulta_documento);
15    $stmt->bind_param("s", $documento);
16    $stmt->execute();
17    $resultado = $stmt->get_result();
18
19    if ($resultado->num_rows > 0) {
20        // El documento ya existe
21        ?>
22        <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
23        <script>
24            document.addEventListener('DOMContentLoaded', function() {
25                Swal.fire({
26                    title: 'Documento Duplicado!',
27                    text: 'Ya existe un usuario registrado con el documento <?php echo $documento; ?>',
28                    icon: 'warning',
29                    confirmButtonText: 'Entendido',
30                    confirmButtonColor: '#3085d6'
31                }).then((result) => {
32                    window.location.href = 'add_user.php';
33                });
34            });
35        </script>
36        <?php
37        exit();
38    } else {
39        // Cifrar la contraseña
40        $hash_contraseña = password_hash($contraseña, PASSWORD_DEFAULT);
41
42        // Insertar los datos en la tabla de usuarios
43        $sql = "INSERT INTO usuarios (documento, nombre, usuario, contraseña, idcargo, id_estado)
44        VALUES (?, ?, ?, ?, ?)";
45        $stmt = $conn->prepare($sql);
46        $stmt->bind_param("ssssii", $documento, $nombre, $usuario, $hash_contraseña, $id_cargo, $id_estado);
47
48        if ($stmt->execute()) {
49            // Usuario agregado exitosamente
50            ?>
51            <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
52            <script>
53                document.addEventListener('DOMContentLoaded', function() {
54                    Swal.fire({
55                        title: '¡Éxito!',
56                        text: 'Usuario agregado correctamente',
57                        icon: 'success',
58                        confirmButtonText: 'Continuar',
59                        confirmButtonColor: '#28a745'
60                    }).then((result) => {
61                        window.location.href = 'add_user.php';
62                    });
63                });
64            </script>
65            <?php
66        } else {
67            // Error al agregar usuario
68            ?>
69            <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
70            <script>
71                document.addEventListener('DOMContentLoaded', function() {
72                    Swal.fire({
73                        title: '¡Error!',
74                        text: 'Error al agregar usuario: <?php echo $conn->error; ?>',
75                        icon: 'error',
76                        confirmButtonText: 'Intentar nuevamente',
77                        confirmButtonColor: '#dc3545'
78                    }).then((result) => {
79                        window.location.href = 'add_user.php';
80                    });
81                });
82            </script>
83            <?php
84        }
85    }
86
87    // Cerrar las conexiones
88    $stmt->close();
89    $conn->close();
90 }
91 ?>

```

Fuente: Los autores

Admin/ Editar usuario

Este fragmento de código permite editar los datos de un usuario en una base de datos a través de un formulario. Cuando se envía el formulario, el script recibe los datos ingresados, como el `documento`, `nombre`, `usuario`, `contraseña`, `idcargo` (cargo o rol del usuario) y `id_estado` (estado de actividad del usuario). La contraseña se cifra con `password_hash` para asegurarla antes de actualizarla en la base de datos. Luego, se ejecuta una consulta SQL para actualizar la información del usuario correspondiente al `id` proporcionado. Si la actualización es exitosa, se muestra un mensaje de confirmación

Fuente: Los autores

Admin/ Agregar Curso

Este código permite agregar un curso a la base de datos y, opcionalmente, asignarle un docente si se proporciona uno en el formulario. Primero, verifica si el formulario fue enviado mediante el método POST y, en caso afirmativo, captura los datos ingresados, incluyendo el código del curso, el nombre, el semestre y, opcionalmente, el ID del docente asignado.

Luego, inserta el curso en la tabla `courses` con los valores capturados y establece un estado activo (`id_estado = 1`). Si la inserción es exitosa, el código obtiene el ID del curso recién creado para relacionarlo con el docente en la tabla `docente_curso`, siempre que se haya seleccionado uno. Si el docente fue asignado correctamente al curso, muestra un mensaje de éxito; de lo contrario, informa de un error al asignar el docente. Finalmente, si el curso se guarda correctamente, pero sin docente asignado, se muestra un mensaje de éxito independiente.

Ilustración 49: Código de la lógica de “agregar usuario”.

```
1 <?php
2 include "../conexion.php";
3
4 if ($_SERVER["REQUEST_METHOD"] == "POST") {
5     $code = mysqli_real_escape_string($conn, $_POST["code"]);
6     $nombre_curso = mysqli_real_escape_string($conn, $_POST["nombre_curso"]);
7     $semestre = mysqli_real_escape_string($conn, $_POST["semestre"]);
8     $docente_id = !empty($_POST["docente_id"]) ? mysqli_real_escape_string($conn, $_POST["docente_id"]) : null;
9
10    // 1. Inserta el curso en la tabla courses
11    $sql_curso = "INSERT INTO courses (code, descripcion, semestre, id_estado) VALUES ('$code', '$nombre_curso', '$semestre', 1)";
12
13    if ($conn->query($sql_curso) === TRUE) {
14        $curso_id = $conn->insert_id; // Obtiene el ID del curso recién insertado
15
16        // 2. Si se seleccionó un docente, inserta la relación en docente_curso
17        if ($docente_id) {
18            $sql_docente_curso = "INSERT INTO docente_curso (docente_id, course_code) VALUES ('$docente_id', '$curso_id')";
19            if ($conn->query($sql_docente_curso) === TRUE) {
20                echo "<script>
21                    function showAlert() {
22                        var message = 'Curso guardado exitosamente';
23                        message += ' El docente también ha sido asignado al curso.';
24                        alert(message);
25                        window.location.href = 'index_docente.php';
26                    }
27                    showAlert();
28                </script>";
29            } else {
30                echo "<script>
31                    function showAlert() {
32                        alert('Error al asignar docente: " . $conn->error . "');
33                    }
34                    showAlert();
35                </script>";
36            }
37        } else {
38            echo "<script>
39                function showAlert() {
40                    alert('Curso guardado exitosamente');
41                    window.location.href = 'index_docente.php';
42                }
43                showAlert();
44            </script>";
45        }
46    } else {
47        echo "<script>
48            function showAlert() {
49                alert('Error al guardar el curso: " . $conn->error . "');
50            }
51            showAlert();
52        </script>";
53    }
54 }
55
56 $conn->close();
57 ?>
```

Fuente: Los autores

Admin/ Listar cursos

Este código genera una lista de cursos en formato de tarjetas, mostrando la información de cada curso y permitiendo realizar acciones como ver detalles, editar o eliminar. Primero, el script se conecta a la base de datos y ejecuta una consulta para obtener todos los cursos junto con su código, descripción, semestre y estado. A continuación, recorre cada curso y verifica si tiene un docente asignado, consultando la tabla docente_curso para obtener el nombre del docente correspondiente; si no hay uno asignado, muestra el texto "DO". Para cada curso, se genera una tarjeta HTML que presenta la descripción del

curso, el nombre del docente (o "DO" si no hay asignación), el estado (mostrado en verde como "Activo" si id_estado es 1, o en rojo como "Inactivo" si es 2), y botones de acción para ver la información del curso, editarlo o eliminarlo. Si no se encuentran cursos, muestra un mensaje alternativo que dice "No hay cursos disponibles". Finalmente, el código cierra la conexión a la base de datos para optimizar el uso de recursos. Este diseño permite una visualización organizada y clara de los cursos, facilitando la gestión y actualización de la información en un entorno web.

Ilustración 50: Código de la lógica de “listar cursos”.

```

1 <?php
2 // Suponiendo que tienes una conexión a la base de datos
3 include "../conexion.php";
4
5 // Consulta para obtener los cursos con su información
6 $sql = "SELECT code, descripcion, semestre, id_estado FROM courses";
7 $result = $conn->query($sql);
8
9 if ($result->num_rows > 0) {
10 while ($row = $result->fetch_assoc()) {
11     $code = $row['code'];
12     $descripcion = $row['descripcion'];
13     $semestre = $row['semestre'];
14     $estado = $row['id_estado'];
15
16     $nombre_docente = "DO";
17     $consulta_docente = "SELECT u.nombre
18 FROM docente_curso dc
19 JOIN usuarios u ON dc.docente_id = u.id
20 WHERE dc.course_code = ?";
21
22 $stmt = $conn->prepare($consulta_docente);
23 $stmt->bind_param("i", $code);
24 $stmt->execute();
25 $stmt->bind_result($nombre);
26 if ($stmt->fetch()) {
27     $nombre_docente = $nombre; // Cambia a nombre del docente si está asignado
28 }
29 $stmt->close();
30 // Generar la tarjeta para cada curso
31 echo
32 <div class="col-xl-3 col-md-6 mb-4">
33 <div class="card border-left-warning shadow h-100 py-2">
34 <div class="card-body">
35 <div class="row no-gutters align-items-center">
36 <div class="col mr-2">
37 <div class="h5 mb-0 font-weight-bold text-gray-800"> . htmlspecialchars($descripcion) . </div>
38 <div class="text-xs font-weight-bold text-warning text-uppercase mb-1">
39 . htmlspecialchars($nombre_docente) .
40 </div>
41 <div class="text-xs font-weight-bold text- . ($estado == 1 ? 'success' : 'danger') . ' text-uppercase mb-1">
42 Estado: . ($estado == 1 ? 'Activo' : 'Inactivo') .
43 </div>
44 </div>
45 <div class="col-auto">
46 <!-- Información -->
47 <a href="edit_calculol.php?code=" . urlencode($code) . '" class="btn btn-info btn-circle">
48 <i class="fas fa-info-circle"></i>
49 </a>
50 <!-- Editar -->
51 <a href="edit_calculol.php?code=" . urlencode($code) . '" class="btn btn-warning btn-circle">
52 <i class="fas fa-exclamation-triangle"></i>
53 </a>
54 <!-- Eliminar -->
55 <a href="delete_course.php?code=" . urlencode($code) . '" class="btn btn-danger btn-circle">
56 <i class="fas fa-trash"></i>
57 </a>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 }
64 } else {
65 echo "opNo hay cursos disponibles.</p>";
66 }
67 $conn->close();
68 >
69 >
70

```

Fuente: Los autores

Admin / Listar estudiantes matriculados a un curso

Este código permite al administrador ver la lista de estudiantes matriculados en un curso específico y cambiar el estado del curso entre activo e inactivo. Primero, obtiene el código del curso desde la URL y, si está presente el parámetro estado, actualiza el estado del curso en la base de datos y muestra una alerta con SweetAlert confirmando la actualización. Luego, realiza una consulta para obtener la lista de estudiantes matriculados, filtrando solo aquellos con el rol de estudiante, y muestra sus nombres, usuarios y estado en una tabla. Si no hay estudiantes matriculados, muestra un mensaje informativo. La interfaz proporciona una visualización clara y opciones de gestión rápida sobre el curso seleccionado.

Ilustración 51: Código de la lógica de "listar estudiantes matriculados a un curso".

```

1  </php
2  // Conexión a la base de datos
3  include "../conexion.php";
4
5  // Obtener el código del curso desde los parámetros
6  if (isset($_GET['code'])) {
7      $code = $_GET['code'];
8  } else {
9      // Si no se proporciona el código, puedes mostrar un mensaje de error o redirigir a otra página
10     echo "No se proporcionó el código del curso.";
11     exit;
12 }
13
14 // Verificar si se han recibido parámetros para cambiar el estado
15 if (isset($_GET['estado'])) {
16     $estado = $_GET['estado']; // 1 para activo, 2 para inactivo
17
18     // Actualizar el estado del curso en la base de datos
19     $sql = "UPDATE courses SET id_estado = ? WHERE code = ?";
20     $stmt = $conn->prepare($sql);
21     $stmt->bind_param("iI", $estado, $code);
22
23     if ($stmt->execute()) {
24         echo "<script src='https://cdn.jsdelivr.net/npm/sweetalert2@11.7.0/dist/sweetalert2.all.min.js'></script>";
25         echo "<script>";
26             Swal.fire({
27                 icon: 'success',
28                 title: 'Estado actualizado',
29                 text: 'El estado del curso ha sido actualizado correctamente.',
30                 showConfirmButton: false,
31                 timer: 1500
32             }).then(() => {
33                 window.location.href = window.location.pathname + '?code=' . $code . '';
34             });
35         </script>";
36     } else {
37         echo "<script src='https://cdn.jsdelivr.net/npm/sweetalert2@11.7.0/dist/sweetalert2.all.min.js'></script>";
38         echo "<script>";
39             Swal.fire({
40                 icon: 'error',
41                 title: 'Error',
42                 text: 'Error al actualizar el estado del curso.'
43             });
44         </script>";
45     }
46 }
47
48 // Obtener el estado actual del curso
49 $sql = "SELECT id_estado FROM courses WHERE code = ?";
50 $stmt = $conn->prepare($sql);
51 $stmt->bind_param("s", $code);
52 $stmt->execute();
53 $result = $stmt->get_result();
54 $row = $result->fetch_assoc();
55 $estado_actual = $row['id_estado'];
56
57 // Consulta para obtener los estudiantes matriculados en el curso
58 $sql = "SELECT u.nombre, u.usuario, u.id_estado
59 FROM estudiante_curso ec
60 JOIN usuarios u ON ec.usuario_id = u.id
61 WHERE ec.course_code = ? AND u.idcargo = 2";
62 $stmt = $conn->prepare($sql);
63 $stmt->bind_param("s", $code);
64 $stmt->execute();
65 $result = $stmt->get_result();
66
67 if ($result->num_rows > 0) {
68     // Mostrar la tabla de estudiantes
69     echo '
70 <div class="card shadow mb-4" style="width: 80%;>
71 <div class="card-header py-3">
72 <h3 class="m-0 font-weight-bold text-primary">Estudiantes</h3>
73 </div>
74 <div class="card-body">
75 <div class="table-responsive">
76 <table class="table table-bordered" id="dataTable" width="100%" cellpadding="0">
77 <thead>
78 <tr style="color: rgb(78, 115, 223);>
79 <th>Nombre</th>
80 <th>Usuario</th>
81 <th>Estado</th>
82 </tr>
83 </thead>
84 <tbody>';
85
86     while ($row = $result->fetch_assoc()) {
87         echo '
88 <tr>
89 <td> . $row['nombre'] . '</td>
90 <td> . $row['usuario'] . '</td>
91 <td> . ($row['id_estado'] == 1 ? 'Activo' : 'Inactivo') . '</td>
92 </tr>';
93     }
94
95     echo '
96 </tbody>
97 </table>
98 </div>
99 </div>
100 </div>';
101 } else {
102     echo "<p>No hay estudiantes matriculados en este curso.</p>";
103 }

```

Fuente: Los autores

Admin / Matricular estudiante a un curso

Este código permite matricular a un estudiante en un curso específico y cuenta con una interfaz de usuario que muestra la "Ficha del Estudiante". Primero, se genera una tarjeta en la que se ofrece la opción de matricular estudiantes. Al hacer clic en "Matricular estudiante", el sistema verifica si el estudiante ya está matriculado en el curso seleccionado para evitar duplicados. Si el estudiante ya está matriculado, muestra un

Fuente: Los autores

Admin / asignar - eliminar docente de un curso

El siguiente fragmento de código tiene como objetivo consultar y mostrar el docente asignado a un curso específico. Para ello, se utiliza una consulta SQL que verifica en la tabla `docente_curso` si existe una relación entre un curso y un docente registrado en la tabla `usuarios`. La consulta selecciona los campos `nombre`, `usuario` e `id_estado` del docente, asegurándose de que su cargo (`idcargo`) sea 3, lo que identifica a los usuarios con rol de docente.

El código emplea una declaración preparada (`$stmt`) para evitar inyecciones SQL, vinculando el parámetro del código del curso (`$code`). Si el resultado de la consulta devuelve al menos un registro, se muestran en una fila los datos del docente: su nombre, usuario y estado (activo o inactivo). En caso contrario, se muestra un mensaje en una fila indicando que no hay docente asignado al curso. Además, se aplican funciones como `htmlspecialchars` para sanitizar los datos y prevenir vulnerabilidades como el cross-site scripting (XSS).

Ilustración 53: Código de la lógica de “Mostrar docente (sí hay) asignado a un curso”.

```

1
2 // Consultar el docente asignado al curso
3 $sql_docente = "SELECT u.nombre, u.usuario, u.id_estado
4                 FROM docente_curso dc
5                 JOIN usuarios u ON dc.docente_id = u.id
6                 WHERE dc.course_code = ? AND u.idcargo = 3";
7 $stmt = $conn->prepare($sql_docente);
8 $stmt->bind_param("s", $code);
9 $stmt->execute();
10 $result = $stmt->get_result();
11
12 if ($result->num_rows > 0) {
13     $row = $result->fetch_assoc();
14     echo "<tr>
15         <td>" . htmlspecialchars($row['nombre']) . "</td>
16         <td>" . htmlspecialchars($row['usuario']) . "</td>
17         <td>" . ($row['id_estado'] == 1 ? 'Activo' : 'Inactivo') . "</td>
18     </tr>";
19 } else {
20     echo "<tr><td colspan='3' class='text-center'>No hay docente asignado</td></tr>";
21 }

```

Fuente: Los autores

El siguiente fragmento de código tiene como propósito manejar la posibilidad de eliminar la asignación de un docente a un curso cuando ya hay un docente registrado para ese curso. Si la consulta previa (\$result) devuelve un número de filas mayor a 0, significa que hay un docente asignado. En ese caso, se genera un formulario (form) que envía una solicitud POST al archivo eliminar_docente.php. Este formulario incluye un campo oculto (input type="hidden") para enviar el código del curso (\$code) como parámetro.

El formulario no se envía directamente; en lugar de eso, se utiliza un botón con la clase btn btn-danger btn-icon-split, que llama a la función confirmarEliminacion () mediante el

atributo onclick. Esto permite solicitar confirmación al usuario antes de proceder con la eliminación, garantizando que no se realicen acciones accidentales.

El botón también incluye un ícono (`<i class="fas fa-user-minus"></i>`) y un texto ("No Asignar Docente") que indican claramente la acción a realizar. Si no hay docentes asignados (es decir, `$result->num_rows` es igual a 0), no se muestra ningún formulario ni botón. Finalmente, se cierra la declaración preparada `$stmt` y los bloques de código HTML correspondientes.

Ilustración 54: Código de la lógica de "eliminar docente de un curso"



```
1
2 if ($result->num_rows > 0) {
3     echo ' <form id="formEliminarDocente" action="eliminar_docente.php" method="POST" style="display: inline;">
4         <input type="hidden" name="course_code" value="' . htmlspecialchars($code) . '">
5         <button type="button" class="btn btn-danger btn-icon-split" onclick="confirmarEliminacion()">
6             <span class="icon text-white-50">
7                 <i class="fas fa-user-minus"></i>
8             </span>
9             <span class="text">No Asignar Docente</span>
10        </button>
11    </form>;
12
13 } else {
14     echo "";
15 }
16 $stmt->close();
17
18
19
20 echo ' </div>
21 </div>;
22
```

Fuente: Los autores.

El siguiente código está diseñado para asignar un docente a un curso mediante un modal interactivo que utiliza Bootstrap y PHP para mostrar una lista de docentes disponibles

como opciones seleccionables. La funcionalidad del código se divide en varias partes que detallan el proceso de consulta y presentación de datos.

Primero, se establece la estructura del modal, que incluye un formulario HTML que se enviará al archivo `asignar_docente.php` utilizando el método POST. Dentro del formulario, se incluye un campo oculto (`<input type="hidden">`) que contiene el código del curso, asegurando que el sistema pueda identificar a qué curso se está asignando el docente.

El siguiente paso es realizar una consulta a la base de datos para obtener una lista de todos los docentes activos. Esto se logra mediante una sentencia SQL que selecciona a los usuarios cuyo rol corresponde al de docente (`idcargo = 3`) y que tienen un estado activo (`id_estado = 1`). Los resultados de esta consulta se ordenan alfabéticamente por el nombre del docente, lo que facilita la navegación y selección.

Después, se realiza otra consulta para determinar si ya existe un docente asignado al curso actual. Si se encuentra un docente asignado, su ID se almacena en una variable (`$docente_actual_id`) para que pueda ser identificado y marcado como seleccionado en la lista de opciones.

Posteriormente, se genera una lista de docentes disponibles utilizando botones de radio. Para cada docente, el sistema verifica si está asignado a otro curso mediante una consulta adicional. Si un docente está asignado a otro curso y no es el actual, se muestra un indicador junto a su nombre que dice "(Asignado a otro curso)". Esto proporciona claridad sobre la disponibilidad de los docentes. Además, si el docente es el actualmente

asignado al curso, su opción se marca como seleccionada automáticamente (checked), garantizando que la asignación actual sea fácilmente visible.

Por último, la lista de docentes se presenta dentro de una sección desplazable para mantener la interfaz compacta y manejable, incluso si hay una gran cantidad de docentes. El atributo required en los botones de radio asegura que el usuario debe seleccionar obligatoriamente un docente antes de enviar el formulario.

Ilustración 55: Código de la lógica de "asignar docente a un curso"

```

1
2 // Modal para asignar docente
3 echo '
4 <div class="modal fade" id="asignarDocenteModal" tabindex="-1" role="dialog" aria-labelledby="asignarDocenteModallabel" aria-hidden="true">
5   <div class="modal-dialog" role="document">
6     <div class="modal-content">
7       <div class="modal-header">
8         <h5 class="modal-title" id="asignarDocenteModallabel">Asignar Docente al Curso</h5>
9         <button type="button" class="close" data-dismiss="modal" aria-label="Close">
10          <span aria-hidden="true">&times;</span>
11        </button>
12      </div>
13      <form id="formAsignarDocente" method="POST" action="asignar_docente.php">
14        <div class="modal-body">
15          <input type="hidden" name="course_code" value="'. htmlspecialchars($code) . "'>
16          <div class="form-group">
17            <label class="d-block mb-3">Seleccione un Docente:</label>
18            <div class="border p-3" style="max-height: 300px; overflow-y: auto;">
19              <div class="custom-control custom-radio">
20
21 // Consultar todos los docentes activos
22 $sql_lista = "SELECT DISTINCT u.id, u.nombre
23             FROM usuarios u
24             WHERE u.idcargo = 3
25             AND u.id_estado = 1
26             ORDER BY u.nombre";
27 $result_lista = $conn->query($sql_lista);
28
29 // Obtener el docente actualmente asignado
30 $sql_actual = "SELECT docente_id FROM docente_curso WHERE course_code = ?";
31 $stmt_actual = $conn->prepare($sql_actual);
32 $stmt_actual->bind_param("s", $code);
33 $stmt_actual->execute();
34 $result_actual = $stmt_actual->get_result();
35 $docente_actual = $result_actual->fetch_assoc();
36 $docente_actual_id = $docente_actual ? $docente_actual['docente_id'] : null;
37 $stmt_actual->close();
38
39 // Ahora listar todos los docentes como radio buttons
40 $counter = 0;
41 while($row = $result_lista->fetch_assoc()) {
42   // Verificar si el docente está asignado a algún curso
43   $sql_asignado = "SELECT course_code FROM docente_curso WHERE docente_id = ?";
44   $stmt_asignado = $conn->prepare($sql_asignado);
45   $stmt_asignado->bind_param("i", $row['id']);
46   $stmt_asignado->execute();
47   $result_asignado = $stmt_asignado->get_result();
48
49   $esta_asignado = $result_asignado->num_rows > 0;
50   $es_docente_actual = ($docente_actual_id == $row['id']);
51
52   // Si el docente está asignado pero no es el docente actual del curso, agregar indicador
53   $texto_asignado = ($esta_asignado && !$es_docente_actual) ? ' (Asignado a otro curso)' : '';
54
55   // Seleccionar el docente actual del curso
56   $checked = $es_docente_actual ? 'checked' : '';
57
58   echo '<div class="custom-control custom-radio mb-2">
59     <input type="radio" id="docente_' . $counter . '" name="docente_id"
60       class="custom-control-input" value="'. htmlspecialchars($row['id']) . "'
61       ' . $checked . ' required>
62     <label class="custom-control-label" for="docente_' . $counter . '">
63       ' . htmlspecialchars($row['nombre']) . '
64       <small class="text-muted">' . htmlspecialchars($texto_asignado) . '</small>
65     </label>
66   </div>';
67
68   $counter++;
69   $stmt_asignado->close();
70 }

```

Fuente: Los autores.

Estudiante / Cursos matriculados

Este código está diseñado para que los estudiantes puedan visualizar los cursos en los que están matriculados. Utiliza PHP y una conexión a la base de datos para consultar y presentar esta información en tarjetas visuales mediante un diseño responsivo.

Primero, se incluye el archivo de conexión a la base de datos (`conexion.php`) y se verifica si el usuario está autenticado. Esto se logra comprobando si existe la variable de sesión `usuario_id`. Si el usuario no está autenticado, se muestra un mensaje indicando que no tiene acceso, y se detiene la ejecución.

A continuación, se obtiene el ID del estudiante desde la sesión y se utiliza para realizar una consulta en la base de datos. La consulta selecciona información relevante de los cursos en los que está matriculado el estudiante, como el código del curso, su descripción, el semestre, el estado del curso y el nombre del docente asociado. Esto se hace utilizando una combinación de tablas: `courses`, `estudiante_curso` (que relaciona a los estudiantes con los cursos) y `usuarios` (para obtener el nombre del docente).

Si el estudiante está matriculado en uno o más cursos, se utiliza un bucle para recorrer los resultados y generar una tarjeta por cada curso. Cada tarjeta muestra el nombre del curso (`descripcion`), el docente asignado (`nombre_docente`), y un botón que redirige a una página específica (`prog_1.php`) para más detalles del curso. El diseño de las tarjetas incluye estilos atractivos como sombras, íconos y colores, utilizando clases de Bootstrap.

Por otro lado, si el estudiante no tiene cursos matriculados, se muestra un mensaje simple indicando que no hay cursos registrados.

Finalmente, las conexiones y sentencias de la base de datos se cierran para liberar recursos. Este enfoque asegura que solo los estudiantes autenticados puedan acceder a la información de sus cursos, presentándola de manera organizada y visualmente agradable.

Ilustración 56: Código de la lógica "visualizar los cursos en los que está matriculado un estudiante"

```
1 <?php
2
3 include "../conexion.php";
4
5 // Verificar que el usuario está autenticado
6 if (!isset($_SESSION['usuario_id'])) {
7     echo 'No autenticado';
8     exit();
9 }
10
11 // Obtener el ID del estudiante desde la sesión
12 $estudiante_id = $_SESSION['usuario_id'];
13
14 // Consulta para obtener los cursos en los que está matriculado el estudiante
15 // Obtener los cursos a los que está matriculado el estudiante
16 $sql = "SELECT c.code, c.descripcion, c.semestre, c.id_estado, u.nombre
17        FROM courses c
18        INNER JOIN estudiante_curso ec ON c.code = ec.course_code
19        INNER JOIN usuarios u ON ec.usuario_id = u.id
20        WHERE ec.usuario_id = ?";
21 $stmt = $conn->prepare($sql);
22 $stmt->bind_param("i", $estudiante_id);
23 $stmt->execute();
24 $result = $stmt->get_result();
25
26 // Verificar si el estudiante está matriculado en algún curso
27 if ($result->num_rows > 0) {
28     while ($row = $result->fetch_assoc()) {
29         $code = $row['code'];
30         $descripcion = $row['descripcion'];
31         $semestre = $row['semestre'];
32         $estado = $row['id_estado'];
33         $nombre_docente = $row['nombre']; // Nombre del docente
34
35         // Generar la tarjeta para cada curso
36         echo '
37 <div class="col-xl-3 col-md-6 mb-4">
38 <!-- Primer curso -->
39 <div class="card border-left-primary shadow h-100 py-2">
40 <div class="card-body">
41 <div class="row no-gutters align-items-center">
42 <div class="col mr-2">
43 <div class="h5 mb-0 font-weight-bold text-gray-800"> . htmlspecialchars($descripcion) . '</div>
44 <div class="text-xs font-weight-bold text-primary text-uppercase mb-1">
45 <div class="text-xs font-weight-bold text-primary text-uppercase mb-1">
46 <div class="text-xs font-weight-bold text-primary text-uppercase mb-1">
47 <div class="text-xs font-weight-bold text-primary text-uppercase mb-1">
48 <div class="col-auto">
49 <a href="prog_1.php" class="btn btn-primary btn-icon-split">
50 <span class="text">Ir</span>
51 <span class="icon text-white-50">
52 <i class="fas fa-arrow-right"></i>
53 </span>
54 </a>
55 </div>
56 </div>
57 </div>
58 </div>
59 <!-- end curso -->
60 </div>';
61     }
62 } else {
63     echo "<p>No tienes cursos matriculados.</p>";
64 }
65
66 // Cerrar conexiones
67 $stmt->close();
68 $conn->close();
69 ?>
70
```

Fuente: Los autores.

4.4 Fase 4: Realizando pruebas de usabilidad y funcionalidad del aplicativo web para la gestión de tutorías, involucrando activamente a estudiantes, tutores y administradores del sistema.

4.4.1 Pruebas Unitarias

4.4.1.1 Prueba Automatizada: Creación de Cuentas Docente

La prueba automatizada realizada en **Selenium IDE** se denominó "*AdminCrea cuentas docente*", y su objetivo fue validar que la funcionalidad para crear cuentas de docentes dentro de la aplicación opera correctamente. La prueba fue exitosa, ya que no se reportaron errores (el contador de fallos está en 0) y se mostró el mensaje "**completed successfully**" al finalizar.

1. ¿Qué hizo la prueba?

2. Acceso al formulario de creación:

- Se accedió a la página de inicio de sesión.
- Se navegó hacia el apartado de administración de usuarios y posteriormente al formulario para agregar un nuevo docente.

3. Ingreso de datos en el formulario:

- Se escribió el nombre de usuario en el campo correspondiente (inputUsuario) como "**Eramirezz**".

- Se ingresó la contraseña en el campo de contraseña (inputPassword) como **"Eramirezz"**.
- Se completaron los datos adicionales: número de documento **"1002394556"** y nombre completo del docente **"Erney Ramirez"**.

4. Selección de roles:

- Se seleccionó una opción de rol o atributo (mediante un elemento de tipo casilla de verificación).

5. Envío del formulario:

- Se presionó el botón para confirmar y enviar los datos ingresados.
- Posteriormente, se confirmó una ventana emergente (con el botón de **SweetAlert**) que validó el éxito del proceso.

6. ¿Qué significa esto?

La prueba se ejecutó correctamente, cumpliendo con los pasos necesarios para verificar la funcionalidad de creación de cuentas docentes. Esto demuestra que:

- Los campos interactivos del formulario (usuario, contraseña, número de documento, nombre, roles, y botón de confirmación) responden adecuadamente.
- No hubo errores o fallos durante la ejecución, lo que confirma que esta funcionalidad está bien implementada.

- La ventana emergente de confirmación se mostró y respondió como se esperaba, garantizando un flujo de trabajo completo.

7. Conclusión

La prueba de creación de cuentas docentes fue completada con éxito. Los resultados indican que la funcionalidad para registrar nuevos docentes en el sistema opera correctamente, asegurando que este proceso crítico esté listo para su uso por parte de los administradores.

Ilustración 57: captura del test “Creación de Cuentas Docente”.

```
Running 'AdminCrea cuentas docente'  
1. open on http://localhost/TutoriasUPC/login.html OK  
2. setWindowSize on 874x708 OK  
3. click on name=usuario OK  
4. click on css=.login-btn OK  
5. click on linkText=Usuario OK  
6. click on linkText=Agregar Nuevo OK  
7. type on id=inputUsuario with value elef OK  
8. type on id=inputPassword with value admin OK  
9. click on id=inputDoc OK  
10. type on id=inputDoc with value 100239576 OK  
11. click on id=inputName OK  
12. type on id=inputName with value Erney Ramirez OK  
13. type on id=inputUsuario with value ERamirezz OK  
14. type on id=inputPassword with value ERamirezz OK  
15. click on css=div:nth-child(2) > .form-check-label OK  
16. click on css=.btn-user OK  
'AdminCrea cuentas docente' completed successfully
```

Fuente: Los autores

4.4.1.2 Prueba Automatizada: Login del Docente

La prueba automatizada llamada **"Login del Docente"** tuvo como objetivo validar que un docente pueda iniciar sesión correctamente en la aplicación. El resultado fue exitoso, ya que no se reportaron errores y al finalizar se mostró el mensaje **"completed successfully"**.

1. ¿Qué hizo la prueba?

2. Acceso a la página de inicio de sesión:

- Se abrió la URL principal de la página de inicio de sesión (login.html).

3. Ingreso de credenciales:

- En el campo de usuario, se escribió el nombre **"docente"**.
- En el campo de contraseña, se escribió la clave correspondiente **"docente"**.

4. Acción de inicio de sesión:

- Se hizo clic en el botón para enviar el formulario e iniciar sesión.

5. ¿Qué significa esto?

La prueba se ejecutó correctamente, confirmando que:

- Los campos de usuario y contraseña son interactivos y aceptan datos como se espera.

- El formulario de inicio de sesión funciona adecuadamente para usuarios con rol de docente.
- No hubo errores en el flujo del inicio de sesión, lo que indica que esta funcionalidad está correctamente implementada.

6. Conclusión

La prueba fue completada sin problemas y demuestra que el sistema permite a los docentes iniciar sesión sin inconvenientes.

Ilustración 58: captura del test "login del docente".

```
Running 'login del Docente'  
1. open on http://localhost/TutoriasUPC/login.html OK  
2. setWindowSize on 674x708 OK  
3. click on name=usuario OK  
4. type on name=usuario with value docente OK  
5. type on name=contraseña with value docente OK  
6. click on css=button OK  
'login del Docente' completed successfully
```

Fuente: Los autores

4.4.1.3 Prueba Automatizada: Login del Estudiante

La prueba "**Login del Estudiante**" tuvo como objetivo verificar que un estudiante pueda acceder a la aplicación mediante sus credenciales. La prueba fue exitosa, con un mensaje final de "**completed successfully**".

¿Qué hizo la prueba?

1. Acceso a la página de inicio de sesión:

- Se abrió la URL principal de la página de inicio de sesión (login.html).

2. Ingreso de credenciales:

- En el campo de usuario, se ingresó "**pruebaestudiante**".
- En el campo de contraseña, se escribió la clave "**pruebaestudiante1**".

3. Acción de inicio de sesión:

- Se presionó el botón para enviar el formulario.

¿Qué significa esto?

El resultado demuestra que:

- Los campos de ingreso de usuario y contraseña son funcionales para estudiantes.
- El sistema reconoce las credenciales correctas de estudiantes y permite el acceso.
- No se encontraron errores, indicando que esta funcionalidad está debidamente implementada.

Conclusión

La prueba confirma que el sistema permite a los estudiantes iniciar sesión correctamente.

Ilustración 59: captura del test "login del estudiante".

```
Running 'Login del estudiante'  
1. open on http://localhost/TutoriasUPC/login.html OK  
2. setWindowSize on 674x708 OK  
3. click on name=usuario with value pruebaestudiante OK  
4. click on name=contraseña with value pruebaestudiante1 OK  
5. click on css=button OK  
'Login del estudiante' completed successfully
```

Fuente: Los autores

4.4.1.4 Prueba Automatizada: Login del Administrador

La prueba llamada "**Login del Administrador**" tuvo como objetivo garantizar que un administrador pueda iniciar sesión exitosamente. La prueba concluyó con el mensaje "**completed successfully**", indicando su éxito.

1. ¿Qué hizo la prueba?

2. Acceso a la página de inicio de sesión:

- Se accedió a la página principal de inicio de sesión (login.html).

3. Ingreso de credenciales:

- En el campo de usuario, se ingresó el valor "**elef**".
- En el campo de contraseña, se escribió la clave "**admin**".

4. Acción de inicio de sesión:

- Se hizo clic en el botón para enviar el formulario.

5. ¿Qué significa esto?

El resultado indica que:

- Los campos interactivos de usuario y contraseña funcionan correctamente para administradores.
- El formulario procesa las credenciales de administrador y permite el acceso sin errores.

6. Conclusión

La prueba fue completada exitosamente y demuestra que los administradores pueden iniciar sesión sin inconvenientes.

Ilustración 60: captura del test “login del administrador”.

```
Running 'Login del administrador'  
1. open on http://localhost/TutoriasUPC/login.html OK  
2. setWindowSize on 674x708 OK  
3. click on name=usuario OK  
4. type on name=usuario with value elef OK  
5. type on name=contraseña with value admin OK  
6. click on css=button OK  
'Login del administrador' completed successfully
```

Fuente: Los autores

4.4.1.5 Prueba Automatizada: Modificar datos y roles del usuario

La prueba automatizada "**Modificar datos y roles del usuario**" tiene como objetivo validar que un administrador pueda actualizar tanto los datos personales como los roles asignados a un usuario dentro de la plataforma. La prueba concluyó exitosamente con el mensaje "**completed successfully**".

1. ¿Qué hizo la prueba?

2. Acceso al sistema:

- Se abrió la página de inicio de sesión (login.html).
- El administrador inició sesión utilizando las credenciales:
 - Usuario: **elef**
 - Contraseña: **Admin**

3. Navegación hacia la gestión de usuarios:

- Desde el menú principal, se seleccionó la opción "**Usuario**" y luego "**Gestión de Usuario**".

4. Selección de un usuario:

- Se eligió un usuario específico desde la lista en la tabla de usuarios.

5. Modificación de datos y roles:

- Se seleccionaron nuevos roles para el usuario utilizando las opciones de la interfaz.
- Se cambió la contraseña del usuario, escribiendo el nuevo valor **"abscdkl21"**.

6. Confirmación de los cambios:

- Se hizo clic en el botón para guardar los cambios.
- Se confirmó la operación en una ventana emergente.

7. ¿Qué significa esto?

La prueba demuestra que:

- El sistema permite navegar correctamente hacia la sección de gestión de usuarios.
- Los datos del usuario, como roles y contraseña, son editables y pueden actualizarse.
- Los botones de acción y confirmación funcionan según lo esperado.
- No hubo errores durante el proceso, lo que indica que la funcionalidad está implementada correctamente.

8. Conclusión

La prueba fue completada sin problemas y confirma que el sistema permite a los administradores modificar datos personales y roles de los usuarios de forma

exitosa. Esto garantiza que la plataforma soporte la gestión de usuarios con precisión y sin interrupciones.

Ilustración 61: Captura del test "Modificar datos y roles del usuario".

```
Running 'Modificar datos y roles del usuario'  
1. open on http://localhost/TutoriasUPC/login.html OK  
2. setWindowSize on 1382x744 OK  
3. click on name=usuario OK  
4. type on name=usuario with value elef OK  
5. type on name=contraseña with value admin OK  
6. click on css=button OK  
7. click on linkText=Usuario OK  
8. click on linkText=Gestión de Usuario OK  
9. click on css=.odd:nth-child(1) .text OK  
10. click on css=fieldset:nth-child(10) > div:nth-child(2) > label OK  
11. click on css=fieldset:nth-child(12) > div:nth-child(2) > label OK  
12. click on name=contraseña OK  
13. type on name=contraseña with value absodki21 OK  
14. click on css=.btn-success > .text OK  
15. click on css=.swal2-confirm OK  
'Modificar datos y roles del usuario' completed successfully
```

Fuente: Los autores

4.4.1.6 Prueba Automatizada: Matricular estudiante a curso.

La prueba automatizada "**Matricular estudiante a curso**" tiene como objetivo verificar que un docente pueda inscribir correctamente a un estudiante en un curso desde la plataforma. La prueba finalizó exitosamente con el mensaje "**completed successfully**".

1. ¿Qué hizo la prueba?

Acceso al sistema:

Se abrió la página de gestión de cursos del docente (gestion_cursos_teacher.php).

Selección del curso:

Se seleccionó el primer curso disponible en la lista haciendo clic en el botón correspondiente.

Acceso al formulario de matrícula:

Se hizo clic en un botón identificado como "**Matricular estudiante**", lo que probablemente abrió un formulario para realizar la matrícula.

Selección del estudiante:

Se seleccionó un estudiante desde una lista (representado por el selector **#form_17**) para matricularlo en el curso.

Confirmación de la matrícula:

Se confirmó la acción inicial de matrícula y luego una segunda confirmación en una ventana emergente.

¿Qué significa esto?

La prueba asegura que:

- La funcionalidad para abrir la interfaz de gestión de cursos y acceder al formulario de matrícula funciona correctamente.

- Los botones y formularios de matrícula permiten seleccionar estudiantes sin errores.
- Las ventanas emergentes de confirmación operan según lo esperado.

Conclusión

La prueba se completó sin problemas, lo que indica que la funcionalidad de matrícula de estudiantes está implementada correctamente. Esto garantiza que los docentes puedan gestionar las inscripciones en los cursos de manera eficiente y sin interrupciones en la plataforma.

Ilustración 62: Captura del test “Matricular estudiante a curso”.

```
Running 'Matricular estudiante a curso'  
1. open on http://localhost/TutoriasUPC/under/gestion_cursos_teacher.php OK  
2. setWindowSize on 874x708 OK  
3. click on css=.col-xl-3:nth-child(1) .btn:nth-child(1) > .fas OK  
4. click on css=.btn-primary:nth-child(1) > .text OK  
5. click on css=#form_17 .text OK  
6. click on css=.swal2-confirm OK  
7. click on css=.swal2-confirm OK  
'Matricular estudiante a curso' completed successfully
```

Fuente: Los autores

4.4.1.7 Prueba Automatizada: Eliminación de Estudiante del Curso por un Docente

La prueba automatizada realizada en **Selenium IDE** se denominó “*Docente elimina estudiante del curso*”. Su objetivo fue validar que un docente pueda eliminar correctamente a un estudiante de un curso específico. La prueba fue exitosa, ya que no

se reportaron errores (el contador de fallos está en 0) y al finalizar se mostró el mensaje **"completed successfully"**.

1. ¿Qué hizo la prueba?

2. Acceso a la gestión de cursos:

- Se abrió la página principal de gestión de cursos asignados al docente (gestion_cursos_teacher.php).

3. Selección de curso:

- Se seleccionó la primera tarjeta de un curso listado en la página.

4. Eliminación de un estudiante:

- Dentro del curso seleccionado, se hizo clic en el enlace o botón correspondiente para **eliminar a un estudiante**.
- Se confirmó la acción en un cuadro de diálogo emergente utilizando el botón de **SweetAlert**.

5. ¿Qué significa esto?

La prueba se ejecutó correctamente, validando el flujo completo para la eliminación de un estudiante por parte de un docente. Esto confirma que:

- La interfaz de gestión de cursos responde adecuadamente a las interacciones del usuario.

- La funcionalidad para identificar y seleccionar estudiantes en un curso específico está bien implementada.
- El cuadro de diálogo emergente para la confirmación de la eliminación responde como se espera, completando el proceso sin errores.

6. Conclusión

La prueba fue completada con éxito. Los resultados demuestran que la funcionalidad para que un docente elimine a un estudiante de un curso está operativa y cumple con los requisitos establecidos, garantizando un flujo de trabajo eficiente y sin fallos.

Ilustración 63: Captura del test “Eliminación de Estudiante del Curso por un Docente”

```
Running 'Docente elimina estudiante del curso'
1. open on http://localhost/TutoriasUPC/under/gestion_cursos_teacher.php OK
2. setWindowSize on 674x708 OK
3. click on css=.col-xl-3:nth-child(1) .btn:nth-child(1) > .fas OK
4. click on linkText=Eliminar OK
5. click on css=.swal2-confirm OK
'Docente elimina estudiante del curso' completed successfully
```

Fuente: Los autores

4.4.1.8 Prueba Automatizada: Estudiante visualiza tutorías activas

La prueba automatizada "**Estudiante visualiza tutorías activas**" tiene como objetivo verificar que los estudiantes puedan acceder y visualizar las tutorías activas disponibles

en la plataforma. La prueba finalizó exitosamente con el mensaje "**completed successfully**".

1. ¿Qué hizo la prueba?

2. Acceso a la plataforma:

- Se abrió la página principal de los estudiantes (index_student.php).

3. Configuración inicial:

- Se ajustó el tamaño de la ventana del navegador para garantizar una visualización adecuada de la interfaz.

4. Interacción con el menú:

- Se hizo clic en el botón del menú correspondiente para desplegar las opciones relacionadas con los cursos.

5. Visualización de cursos:

- Se seleccionó la opción "**Todos los cursos**", probablemente para mostrar todas las tutorías activas disponibles para el estudiante.

6. ¿Qué significa esto?

La prueba asegura que:

- Los estudiantes pueden acceder correctamente a la sección de tutorías desde su panel principal.

- La navegación por el menú y la selección de opciones funcionan como se espera.
- Las tutorías activas se visualizan sin errores en la plataforma.

7. Conclusión

La prueba se completó con éxito, confirmando que los estudiantes pueden navegar sin problemas hasta la sección de tutorías activas. Esto asegura que la funcionalidad de visualización de tutorías está correctamente implementada y accesible para los usuarios.

Ilustración 64: Captura del test “Título: captura del test”.

```
Running 'Estudiante visualiza tutorias activas '  
1. open on http://localhost/TutoriasUPC/under/index_student.php OK  
2. setWindowSize on 674x708 OK  
3. click on css=.collapsed > img OK  
4. click on linkText=Todos los cursos OK  
'Estudiante visualiza tutorias activas ' completed successfully
```

Fuente: Los autores

4.4.1.9 Prueba Automatizada: El administrador activa o desactiva un curso

La prueba automatizada **"El administrador activa o desactiva un curso"** valida la funcionalidad que permite al administrador cambiar el estado (activo/inactivo) de un curso desde la plataforma. La ejecución concluyó exitosamente con el mensaje **"completed successfully"**.

¿Qué hizo la prueba?

1. Inicio de sesión del administrador:

- Se accedió a la página de inicio de sesión (login.html), y se ingresaron las credenciales del administrador (usuario: *elf*, contraseña: *admin*).
- Se confirmó la autenticación al presionar el botón de inicio de sesión.

2. Navegación a la sección de gestión de cursos:

- Desde el panel de administración, se seleccionó la opción "**Cursos**" y posteriormente "**Gestión de Cursos**".

3. Acción de activación/desactivación:

- Se buscó un curso específico en la lista de cursos disponibles.
- Se seleccionó el botón correspondiente para cambiar su estado (activarlo o desactivarlo).

4. ¿Qué significa esto?

Esta prueba válida que:

- El administrador puede iniciar sesión y navegar por el panel de administración sin problemas.
- La funcionalidad para gestionar el estado de los cursos está operativa y responde como se espera.

- No se reportaron fallos en la interacción con los botones y enlaces para realizar esta acción.

5. Conclusión

La prueba confirmó que el administrador puede activar o desactivar cursos con éxito, lo que garantiza que la funcionalidad de gestión de cursos está correctamente implementada. La plataforma permite modificar el estado de los cursos de forma eficiente y sin errores.

Ilustración 65: Captura del test “El administrador activa o desactiva un curso”.

```
Running 'El administrador activa o desactiva un curso'  
1. open on http://localhost/TutoriasUPC/login.html OK  
2. setWindowSize on 674x708 OK  
3. click on name=usuario OK  
4. type on name=usuario with value elef OK  
5. type on name=contraseña with value admin OK  
6. click on css=.login-btn with value ${KEY_ENTER} OK  
7. click on linkText=Cursos OK  
8. click on linkText=Gestión de Cursos OK  
9. Trying to find linkText=Detalles... OK  
10. click on css=.btn-success > .text OK  
'El administrador activa o desactiva un curso' completed successfully
```

Fuente: Los autores

4.4.2.10 Prueba Automatizada: Estudiante visualiza tutorías activas matriculadas

La prueba automatizada **"Estudiante visualiza tutorías activas matriculadas"** valida que el estudiante pueda acceder correctamente a la lista de tutorías activas en las que está matriculado. La prueba fue completada exitosamente, como lo indica el mensaje **"completed successfully"**.

1. ¿Qué hizo la prueba?

2. Inicio de sesión del estudiante:

- Se accedió a la página de inicio de sesión (login.html), y se ingresaron las credenciales del estudiante (usuario: *pruebaestudiante*, contraseña: *pruebaestudiante1*).
- Se confirmó la autenticación al presionar el botón de inicio de sesión.

3. Acción de visualización de tutorías:

- Se verificó que el estudiante pudiera acceder a las tutorías activas en las que está matriculado dentro de la plataforma, sin errores ni interrupciones.

4. ¿Qué significa esto?

Esta prueba demuestra que:

- El estudiante puede iniciar sesión correctamente en el sistema.
- El sistema muestra correctamente las tutorías activas en las que el estudiante está matriculado.
- No se presentaron errores en la visualización de las tutorías activas.

5. Conclusión

La prueba validó que el estudiante puede visualizar sin problemas las tutorías activas a las cuales está matriculado, lo que asegura que esta funcionalidad está funcionando correctamente en la plataforma.

Ilustración 66: *Captura del test “Estudiante visualiza tutorías activas matriculadas”.*

```
Running 'Estudiante visualiza tutorias activas matriculadas '  
1. open on http://localhost/TutoriasUPC/login.html OK  
2. setWindowSize on 674x708 OK  
3. click on name=usuario with value pruebaestudiante OK  
4. click on name=contraseña with value pruebaestudiante1 OK  
5. click on css=button OK  
'Estudiante visualiza tutorias activas matriculadas ' completed successfully
```

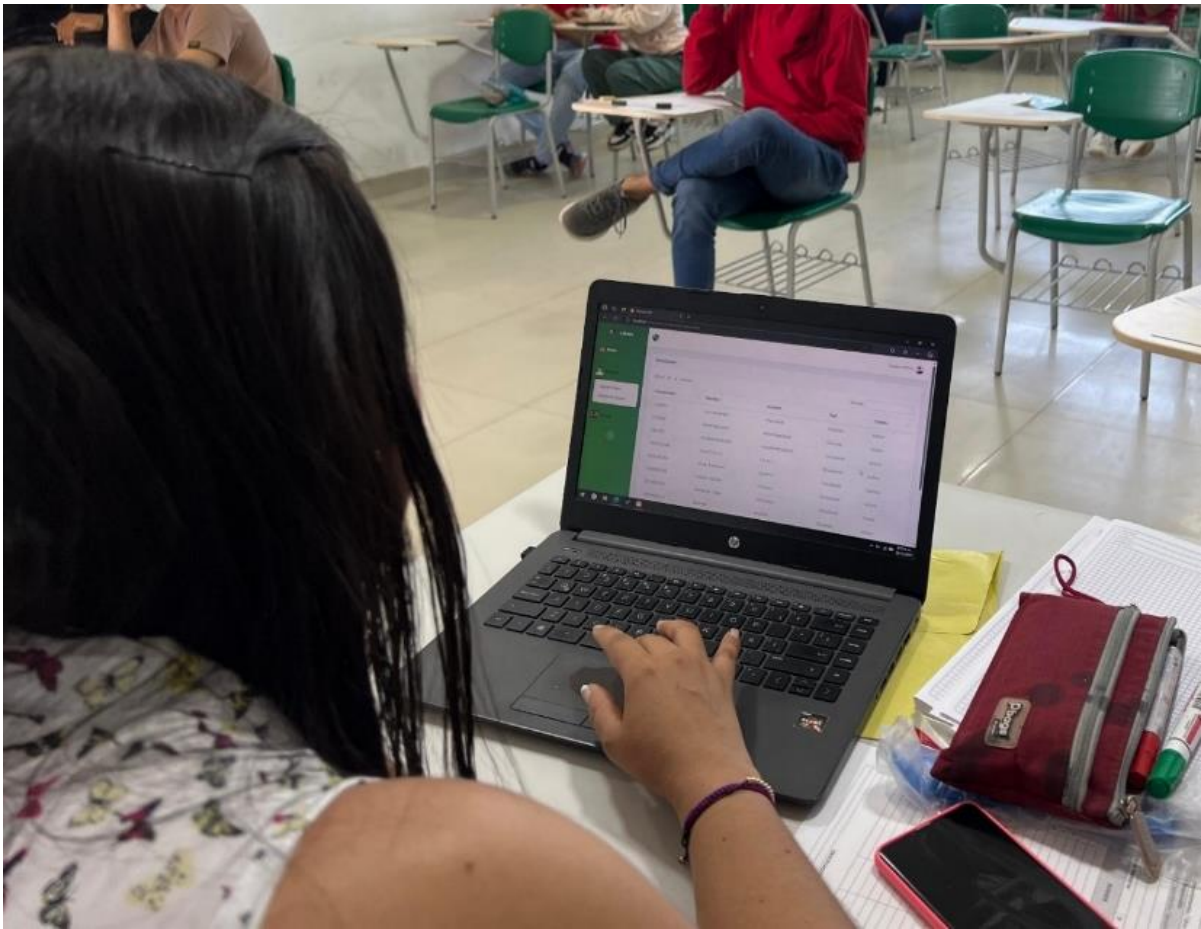
Fuente: Los autores

4.4.2 Realizar pruebas de usabilidad.

Una vez elaborado en su totalidad el aplicativo web y con su funcionamiento, se continua con las pruebas de usabilidad en los diferentes roles (Docente, estudiante). Para ellas, se realizaron pruebas de usabilidad en tutorías de cálculo II, lideradas y ejecutadas por la docente *Leidy Ximena Cortés Velásquez*.

Los estudiantes interactuaron de manera directa con el software, ingresando a su contenido verificando así que fuese un entorno accesible, cómodo y apto para su finalidad académica, mientras que, por parte de la docente, quien realizó monitoreo de quienes estaban en su curso, pudo evidenciar como el aplicativo cumplía con sus expectativas y con lo antes mencionado por ella en las entrevistas realizadas en la fase 1.

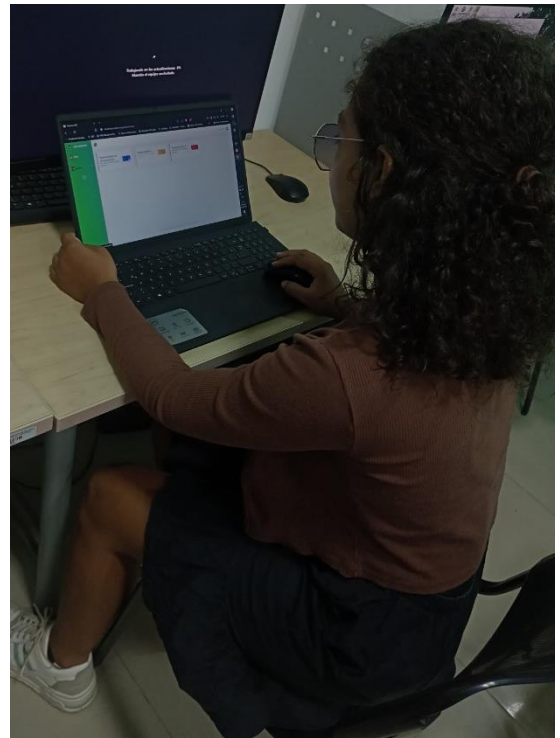
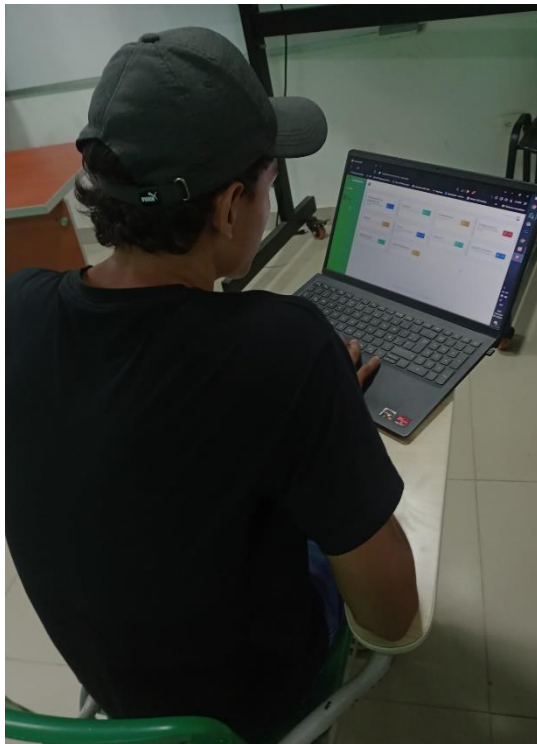
Ilustración 67: Prueba de usabilidad docente.



Fuente: Los autores

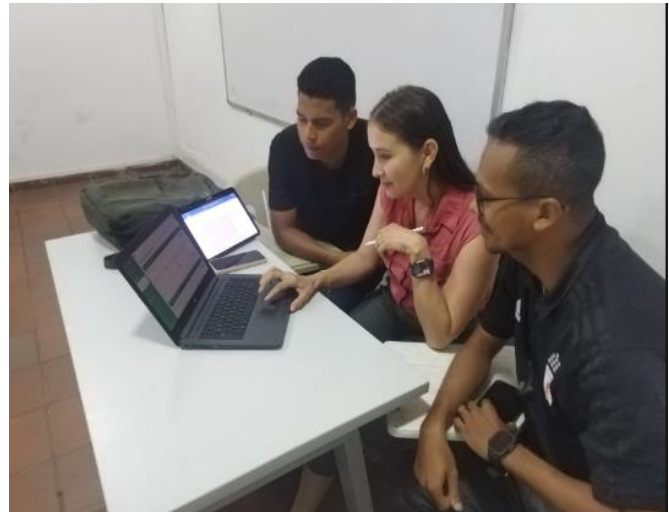
Ilustración 68: Prueba de usabilidad estudiantes.





Fuente: Los autores.

Ilustración 69: Dirección de Programa



Fuente: Los autores.

4.4.3 Análisis de resultados.

Mediante las pruebas de usabilidad que se realizaron entorno al funcionamiento del aplicativo, encontramos que los resultados son positivos en ambos roles. Por parte del estudiante, nos comentó que el aplicativo funcionó de una manera óptima, transcurrió las pruebas con completa normalidad desde su login hasta su finalización de sesión con su rol. Fue una interacción amigable y cómoda ya que por medio de los banner e iconos dentro del aplicativo todo estaba de manera clara y accesible.

Por los lados del rol docente, la tutora Leidy Ximena Cortés Velásquez nos comentó que es un aplicativo que facilitaría más el hecho de que los estudiantes estuviesen desde sus casas más concentrados en la temática vista recientemente y en la temática siguiente, ya que como ella nos comenta *“es un aplicativo que, al estar siempre, ellos pueden ingresar y mirar también lo que se va a dar la próxima clase, así no vienen como tan perdidos”*.

4.4.1 Realizar ajustes finales.

4.4.1.1 El colapsarlo de todos los cursos se pasó a una opción y página independiente

1. Descripción de los cambios realizados

Eliminación de enlaces directos a cursos individuales dentro del menú:

- Anteriormente, los cursos como "Cálculo", y "Programación" se mostraban directamente como enlaces en el colapsable del menú. Estos enlaces fueron eliminados.

2. Adición de un enlace a una página independiente para listar todos los cursos:

- Ahora se incluye un enlace titulado "Todos los cursos" que redirige a una página independiente (all_courses.php) donde se muestran todos los cursos disponibles.

3. Beneficios del cambio

a) Organización más limpia y escalable:

- Al eliminar los enlaces directos a cursos individuales, el menú se vuelve más ordenado y menos saturado, especialmente si en el futuro se agregan más cursos. Esto mejora la navegabilidad del sistema.

b) Centralización del contenido:

- En lugar de tener los cursos listados en un menú desplegable, todos los cursos se gestionan y muestran en una página específica. Esto

permite mantener un diseño más consistente y centralizado, facilitando la implementación de filtros, búsquedas o información detallada de los cursos.

c) Mejora de la experiencia del usuario:

- Los estudiantes ya no tienen que buscar cursos específicos en un menú desplegable potencialmente largo. En su lugar, pueden acceder a una página organizada con todos los cursos y opciones relacionadas, mejorando la experiencia de navegación.

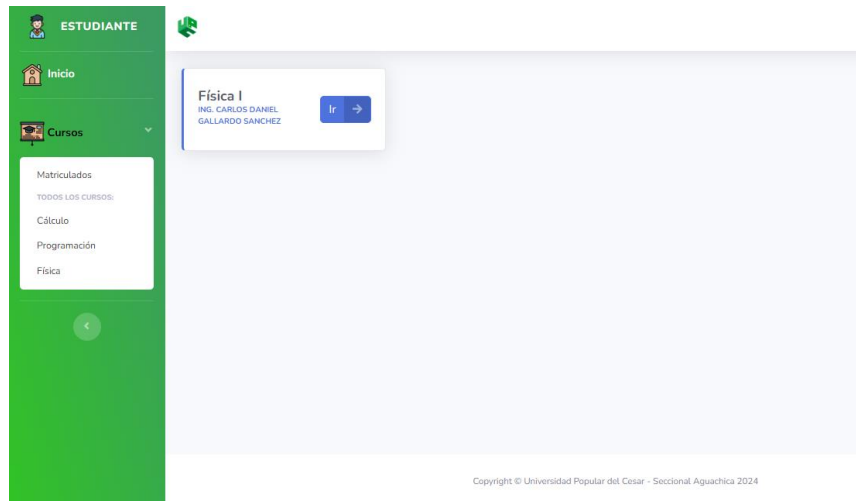
d) Facilidad de mantenimiento:

- Si se necesita modificar el listado de cursos o su funcionalidad, los cambios se realizan en la página `all_courses.php`, evitando tocar el código del menú cada vez que se agregue o elimine un curso.

e) Escalabilidad:

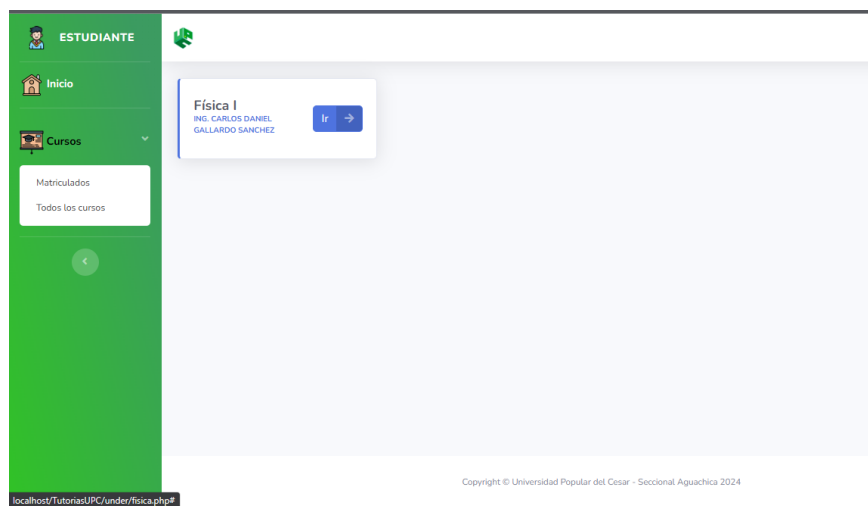
- Este enfoque facilita la gestión de un número creciente de cursos sin afectar la usabilidad del menú. Además, la página `all_courses.php` puede incluir opciones avanzadas como filtros, ordenamientos y descripciones detalladas, lo que no sería posible en un menú desplegable.

Ilustración 70: Captura de la pantalla a modificar en el sidebar estudiante.



Fuente: Los autores.

Ilustración 71: Captura de la pantalla del sidebar de estudiante modificado



Fuente: Los autores.

4.4.1.2 Se mejoraron los estilos del login

Descripción de los cambios realizados

Se añadieron bordes redondeados (`border-radius: 4px`) para un diseño más moderno. El botón ahora tiene un efecto de transición al cambiar de color al pasar el mouse (`transition: background-color 0.3s ease`).

El color del fondo y el texto son más consistentes con un diseño profesional.

Beneficios del cambio:

1. Mejor mantenimiento del código:

- Al utilizar una clase CSS, los estilos del botón pueden ser reutilizados en otros botones del proyecto. Si es necesario modificar el diseño, solo se cambia en la hoja de estilos, afectando a todos los elementos que usan la clase.

2. Separación de responsabilidades:

- El HTML ahora solo define la estructura del botón, mientras que el CSS controla el diseño. Esto sigue el principio de separación de responsabilidades, haciendo el código más limpio y legible.

3. Diseño mejorado:

- Se añadieron bordes redondeados (`border-radius: 4px`) para un diseño más moderno.
- El botón ahora tiene un efecto de transición al cambiar de color al pasar el mouse (`transition: background-color 0.3s ease`).

- El color del fondo y el texto son más consistentes con un diseño profesional.

4. Accesibilidad:

- Al usar: focus en CSS, el botón incluye un efecto visual para usuarios que navegan con el teclado, mejorando la accesibilidad.

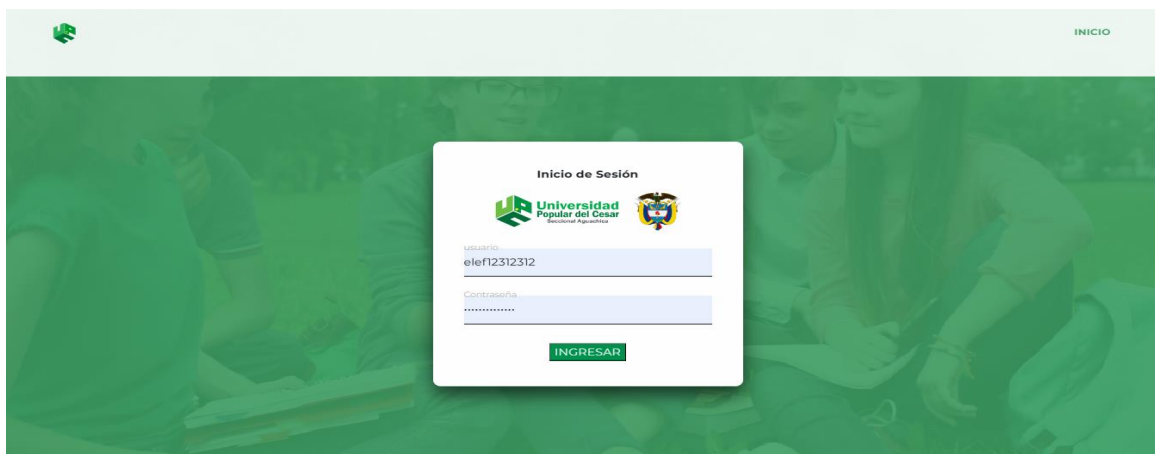
5. Experiencia del usuario (UX):

- La animación al pasar el mouse mejora la percepción de interactividad, haciendo la experiencia más atractiva para el usuario.

6. Uniformidad en el diseño:

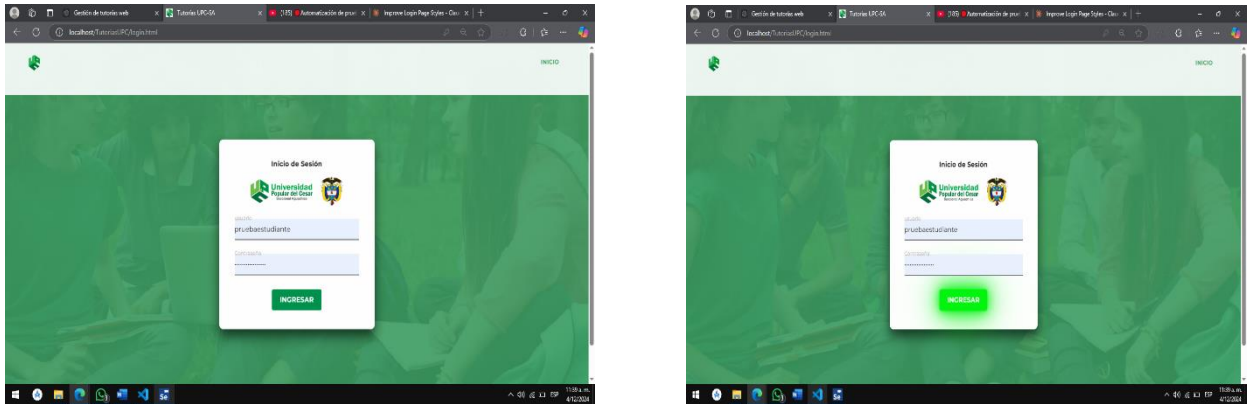
- Con el uso de la clase, todos los botones que utilicen login-btn tendrán el mismo diseño, garantizando coherencia visual en toda la interfaz.

Ilustración 72: Captura de la pantalla del “login” antes de modificar



Fuente: Los autores

Ilustración 73: Capturas de la pantalla del “login” modificado y mejorado en estilos



Fuente: Los autores.

4.4.1.3 Mejoras en los estilos de las tarjetas de los cursos (Administrador)

Se realizaron los siguientes cambios en el diseño de las tarjetas de los cursos que se muestran al administrador:

1. Adición de efectos visuales interactivos:

- Las tarjetas ahora tienen una animación de desplazamiento suave hacia arriba al pasar el cursor por encima (hover), proporcionando una retroalimentación visual atractiva.
- Se agrega un efecto de sombra más prominente durante el hover, destacando la tarjeta seleccionada.

2. Clasificación visual de estados:

- Las tarjetas cuentan con un borde y un color distintivo según el estado del curso:
 - **Verde (Activo):** Los cursos activos tienen un borde verde y una insignia verde.
 - **Rojo (Inactivo):** Los cursos inactivos tienen un borde rojo y una insignia roja.

3. Información organizada y clara:

- El título del curso se muestra en un estilo de encabezado grande y audaz.
- Se incluye el estado del curso como una **insignia (badge)** con colores relevantes.
- Se presenta el nombre del docente asignado con un estilo más discreto, manteniendo el foco en el curso.

4. Acciones dinámicas:

- Se han incorporado botones para **detalles**, **editar** y **eliminar**:
 - Los botones de editar y eliminar están organizados en un grupo de botones para optimizar el espacio y mantener una estética limpia.
 - Los íconos acompañan cada acción para facilitar la identificación de las funciones.

5. Experiencia del usuario mejorada:

- Los botones para las acciones aparecen con mayor visibilidad al pasar el cursor sobre la tarjeta, reduciendo la saturación visual inicial y enfocándose en la interacción del usuario.

6. Beneficios del nuevo diseño

a) Mayor interacción y atractivo visual:

- Los efectos dinámicos hacen que las tarjetas sean más atractivas visualmente y brindan una experiencia moderna y profesional.

b) Claridad y jerarquización de la información:

- El diseño actualizado organiza la información de manera que lo más relevante (estado y nombre del curso) sea claramente visible, mientras que los detalles secundarios (nombre del docente) se presentan de manera elegante.

c) Identificación rápida del estado de los cursos:

- Los colores de los bordes e insignias permiten identificar rápidamente si un curso está activo o inactivo, mejorando la eficiencia del administrador.

d) Facilidad de uso:

- Las acciones asociadas a cada curso (editar, eliminar, ver detalles) están claramente definidas y accesibles, agilizando las tareas administrativas.

e) **Estética profesional y consistente:**

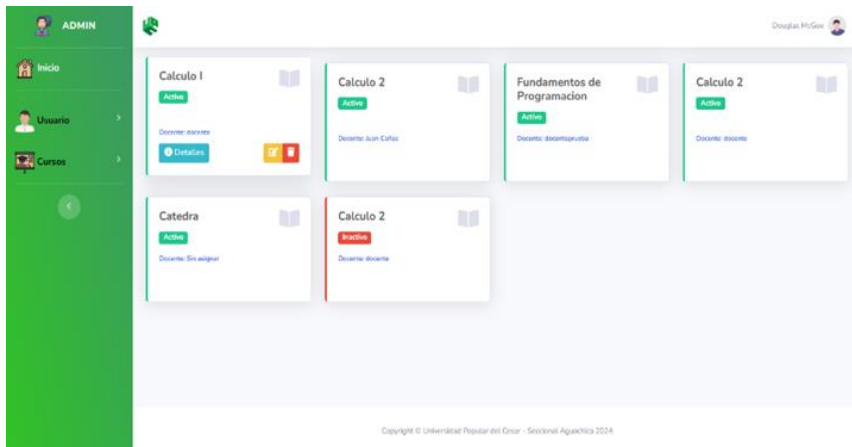
- El diseño actualizado está alineado con una estética moderna y profesional, lo que mejora la percepción general del sistema y la satisfacción del usuario.

Ilustración 74: Captura de las tarjetas de los cursos (Administrador) antes de ajustes.



Fuente: los autores

Ilustración 75: Captura de las tarjetas de los cursos (Administrador) después de ajustes.



Fuente: Los autores.

5 CONCLUSIONES

El desarrollo de este aplicativo web para tutorías responde a la problemática de la falta de herramientas adecuadas para coordinar y gestionar las tutorías académicas sin interferir con los horarios de las asignaturas matriculadas por los estudiantes. En muchos casos, los estudiantes enfrentan dificultades para encontrar momentos en los que puedan recibir apoyo académico debido a los cruces con sus clases, lo que impacta negativamente en su rendimiento y aprovechamiento de las tutorías. A partir de este análisis, se identificó la necesidad de una solución flexible que permitiera a los usuarios acceder a las tutorías en cualquier momento, sin depender de un horario fijo, y que a su vez optimizara el proceso de programación y gestión de tutorías.

El sistema desarrollado permite a los estudiantes y tutores coordinar sesiones de tutoría en cualquier momento del día, sin la necesidad de ajustarse a horarios predeterminados. Utilizando una plataforma disponible las 24 horas del día, los 7 días de la semana, el aplicativo facilita que los estudiantes puedan solicitar tutorías de acuerdo con su disponibilidad y las necesidades de apoyo académico. La plataforma emplea un sistema inteligente que asegura que no existan solapamientos entre las tutorías y las asignaturas académicas de los estudiantes, evitando conflictos y garantizando que las tutorías sean accesibles sin afectar el desarrollo de las actividades académicas principales.

El resultado obtenido demuestra que el aplicativo logra satisfacer la necesidad planteada, eliminando el problema de la gestión de horarios conflictivos al ofrecer un sistema flexible y accesible para la programación de tutorías. Gracias a su disponibilidad continua, los

estudiantes pueden acceder a sesiones de apoyo en el momento que mejor se ajuste a sus horarios y compromisos académicos, lo que mejora la eficiencia y el aprovechamiento de los recursos disponibles. En resumen, el sistema no solo resuelve la problemática de los cruces de horarios, sino que también optimiza la organización y gestión de las tutorías, contribuyendo a una experiencia académica más eficiente y personalizada.

6 RECOMENDACIONES

Para el uso óptimo del software desarrollado, se recomienda designar un administrador capacitado que se encargue de gestionar las operaciones principales del sistema, como registrar usuarios (docentes y estudiantes), crear y configurar cursos, asignar docentes responsables y cambiar el estado de los cursos según sea necesario. Además, es fundamental capacitar a todos los usuarios del sistema para garantizar un manejo adecuado, abarcando desde el ingreso al sistema hasta la utilización de cada funcionalidad. Es aconsejable realizar mantenimientos periódicos, preferiblemente cada 3 o 6 meses, para actualizar el sistema, corregir errores y garantizar su compatibilidad con nuevas tecnologías, así como respaldar periódicamente la información almacenada para evitar pérdidas de datos. Se sugiere también implementar el sistema de manera gradual, iniciando con pruebas controladas antes de su despliegue total, y monitorear su uso para identificar posibles mejoras. Por último, se debe garantizar la escalabilidad del sistema para adaptarlo al crecimiento de la universidad y planificar futuras actualizaciones que permitan optimizar su funcionamiento y satisfacer las necesidades de los usuarios. Estas medidas asegurarán la sostenibilidad y eficiencia del software a largo plazo.

7 BIBLIOGRAFIA

- (UNIR), L. U. (2024). La Universidad en Internet (UNIR). Obtenido de <https://colombia.unir.net/actualidad-unir/arquitectura-software-y-sistemas/>
- Alcaldía de Aguachica Cesar. (28 de 09 de 2017). Obtenido de Alcaldía de Aguachica Cesar: <https://www.aguachica-cesar.gov.co/municipio/nuestro-municipio>
- Alcaldia Mayor de Bogotá D.C. (s.f.). Obtenido de <https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=3431>
- Amazon. (2023). Aws. Obtenido de <https://aws.amazon.com/es/what-is/web-application/>
- Analuisa Medina, L. A. (2019). Desarrollo de una aplicación web para la reservación de tutorías, aplicado a los estudiantes de la Pontificia Universidad Católica del Ecuador Sede Ambato. . Ambato: Pontificia Universitaria Católica del Ecuador.
- Arias. (2012).
- Barragán, S., & Barrera, O. (2020). Monitorias y tutorias, Benchmarking interinstitucional para el mejoramiento de la calidad académica y la permanencia estudiantil. Bogotá: Universida Piloto de Colombia.
- Capítulo II Marco Teórico. (s.f.).
- Castro, A., & Rojas, J. (2023). Desarrollo de un aplicativo web que permite enlazar monitores académicos con estudiantes garantizando calidad, seguridad, accesibilidad y agilidad del servicio, Santa marta. Santa Marta: Universidad del Magdalena.
- Colcha Toscano, D. (2023). Desarrollo de una plataforma web para el registro de tutorías de la Carrera de Ciencias de la Computación de la Universidad Politécnica Salesiana. Quito: Universidad Politécnica Salesiana.
- Conopoima, Y. (15 de 07 de 2020). Espíritu Emprendedores. Obtenido de <http://espirituemprendedores.com/index.php/revista/article/view/200>
- Conopoima, Y. (15 de 07 de 2020). Espíritu Emprendedores. Obtenido de <http://espirituemprendedores.com/index.php/revista/article/view/200>
- Corrales, J. A. (2 de Agosto de 2019). rockcontent. Obtenido de <https://rockcontent.com/es/blog/interfaz-de-usuario/>
- Coulorius. (2002). Capítulo II Marco Teórico.

Dick, F., & Bryan, V. (2023). Sistema web de gestión de tutorías pedagógicas para el control y seguimiento de actividades de los estudiantes de la Carrera de Software de la Universidad de Guayaquil. GUAYAQUIL: UNIVERSIDAD DE GUAYAQUIL.

Distancia, U. E. (2016). Pedagogía Universitaria para la educación a distancia. Obtenido de https://multimedia.uned.ac.cr/pem/pedagogia_universitaria/paginas_unidad3/concepto_gestion.html

Drew. (2020). Drew. Obtenido de <https://blog.wearedrew.co/productividad/-ventajas-y-desventajas-de-la-metodologia-scrum>

Drumond, C. (s.f.). ATlassian . Obtenido de <https://www.atlassian.com/es/agile/scrum>

Easymailing. (2024). Obtenido de <https://easymailing.com/que-es/html>

Enciclopedia Concepto. (22 de Octubre de 2021). Enciclopedia Concepto. Obtenido de <https://concepto.de/sistema/>

Enciclopedia Concepto. (19 de Noviembre de 2023). Enciclopedia Concepto. Obtenido de <https://concepto.de/software/>

Escobar, L., Isasi Olmedo, B., & Ayala, K. (2022). Aplicación web para gestión de solicitudes de agendamiento de tutoría para la FP-UNE. Ciudad del Este: Facultad Politécnica - Universidad Nacional del Este.

Ferrovial. (2024). ferrovial. Obtenido de <https://www.ferrovial.com/es/stem/teoria-de-la-informacion/#:~:text=%C2%BFQu%C3%A9%20es%20la%20teor%C3%ADa%20de%20transmisi%C3%B3n%20de%20una%20informaci%C3%B3n.>

Hernández, S., & Baptista, F. y. (2014).

Higuera, J., & Moreno, j. (2014). Implementación de aplicativo web 2.0 con fines educativos para el área de biotecnología en el colegio La Amistad I.E.D de Bogotá D.C. Bogotá: Universidad Católica.

Huet, P. (24 de 08 de 2022). OpenWebinars. Obtenido de <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>

IBM. (2023). IBM . Obtenido de <https://www.ibm.com/docs/es/controller/11.1.0?topic=settings-defining-user-roles>

Jesús. (21 de 02 de 2022). DONGEE. Obtenido de <https://www.dongee.com/tutoriales/que-es->

hosting/?utm_source=google&utm_medium=cpc&utm_campaign=tutoriales&utm_id=que-es-hosting&utm_term=que-es-hosting&tm=tt&ap=gads&aaid=adaFnZvkSdpvp&gad_source=1&gclid=Cj0KCQjw7Py4BhCbARIsAMMx-_K42nxizISFHIESMCqgPRVDSY

Jiménez, J. D. (20 de 01 de 2019). OpenWebinars. Obtenido de <https://openwebinars.net/blog/que-es-html5/>

Lelann. (1981). Capítulo II Marco Teórico.

Mendoza, S. (2022). Implementación de un sistema para la gestión de tutorías para los estudiantes de la unidad educativa particular Jonh Pierre del Hierro Melchiade. MANABÍ : UNIVERSIDAD ESTATAL DEL SUR DE MANABÍ .

Merchán, E., & Merchán, A. (2014). Contrucción e implementación de un objeto virtual ded aprendizaje en la plataforma moodle, para facilitar el proceso de enseñanza - aprendizaje de la asignatura de ciencias básicas I . Aguachica: Universidad Popular del Cesar, Seccional Aguachica.

Mundo Henry. (22 de 06 de 2011). Obtenido de https://blog.soyhenry.com/que-es-css-y-para-que-sirve/?gad_source=1&gclid=Cj0KCQjw7Py4BhCbARIsAMMx-_ILS4Wwym6icSCHN50OsZ1aWCm_ZCa6PgP6j6IT0qAs8JG7XS1TNV8aAlkbEALw_wcB

Niño Rojas, V. M. (2011). Metodología de la Investigación. Bogotá: Ediciones de la U.

Oracle. (24 de 11 de 2020). Oracle. Obtenido de <https://www.oracle.com/co/database/what-is-database/>

Ortiz Granja, D. (2015). El constructivismo como teoría y Método de enseñanza. Cuenca: Universidad Politécnica Saleciana.

Padilla, I., & Mayoral, V. (2017). Las tutorías académicas en el fortalecimiento del álgebra en estudiantes de octavo grado en una escuela distrital de Barranquilla. Barranquilla: Universidad del Atlántico.

Palella, & Martins. (2008).

PMOinformatica.com. (18 de 02 de 2021). PMOinformatica.com. Obtenido de <https://www.pmoinformatica.com/2021/02/diagrama-de-casos-de-uso-definicion-elementos-ejemplos-como-hacer.html>

Portal Único del Estado Colombiano. (s.f.). Obtenido de <https://www.minambiente.gov.co/politica-de-proteccion-de-datos-personales/#:~:text=Ley%20de%20Protecci%C3%B3n%20de%20Datos,de%20naturaleza%20p%C3%ABlica%20o%20privada.>

Portal Único del Estado Colombiano. (s.f.). Obtenido de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=87419>

Quimbayo, J., Llanos, J., Alarcón, Á., & Borrero, S. (2019). Desarrollo de una aplicación web de seguimiento y deserción para los estudiantes de la corporación Universitaria del Huila - CORHUILA. Huila: Corporación Universitaria del Huila.

Rincón Páez, R. (2020). UNIVERSIDAD POPULAR DEL CESAR SECCIONAL AGUACHICA. Obtenido de <https://aguachica.unicesar.edu.co/historia/>

Sampieri, R., Fernández, D. C., & Baptista, D. P. (1997). Metodología de la investigación . Panamericana Formas e Impresos S.A.

Secretaria de salud. (30 de Octubre de 2017). Gobierno de México. Obtenido de <https://www.gob.mx/salud/articulos/tutorias-academicas#:~:text=La%20tutor%C3%ADa%20acad%C3%A9mica%20es%20un,rezago%20y%20el%20abandono%20escolar.>

Souza, I. d. (09 de 2020). rockcontent blog. Obtenido de <https://rockcontent.com/es/blog/php/>

Tamayo, & Tamayo. (2006).

Tanenbaum. (2001). Capítulo II Marco Teórico .

Terán, M. (06 de 2020). Universidad San Francisco de Quito. Obtenido de https://www.usfq.edu.ec/sites/default/files/2020-06/pea_013_0007.pdf

Valtx. (s.f.). Obtenido de <https://www.valtx.pe/blog/metodologias-para-el-desarrollo-de-software-que-son-y-para-que-sirven>

Villa, H., & Atehortua, S. (2023). Aplicación Web de Tutorías Académicas. Medellín: Tenológico de Antioquia - Institución Universitaria.

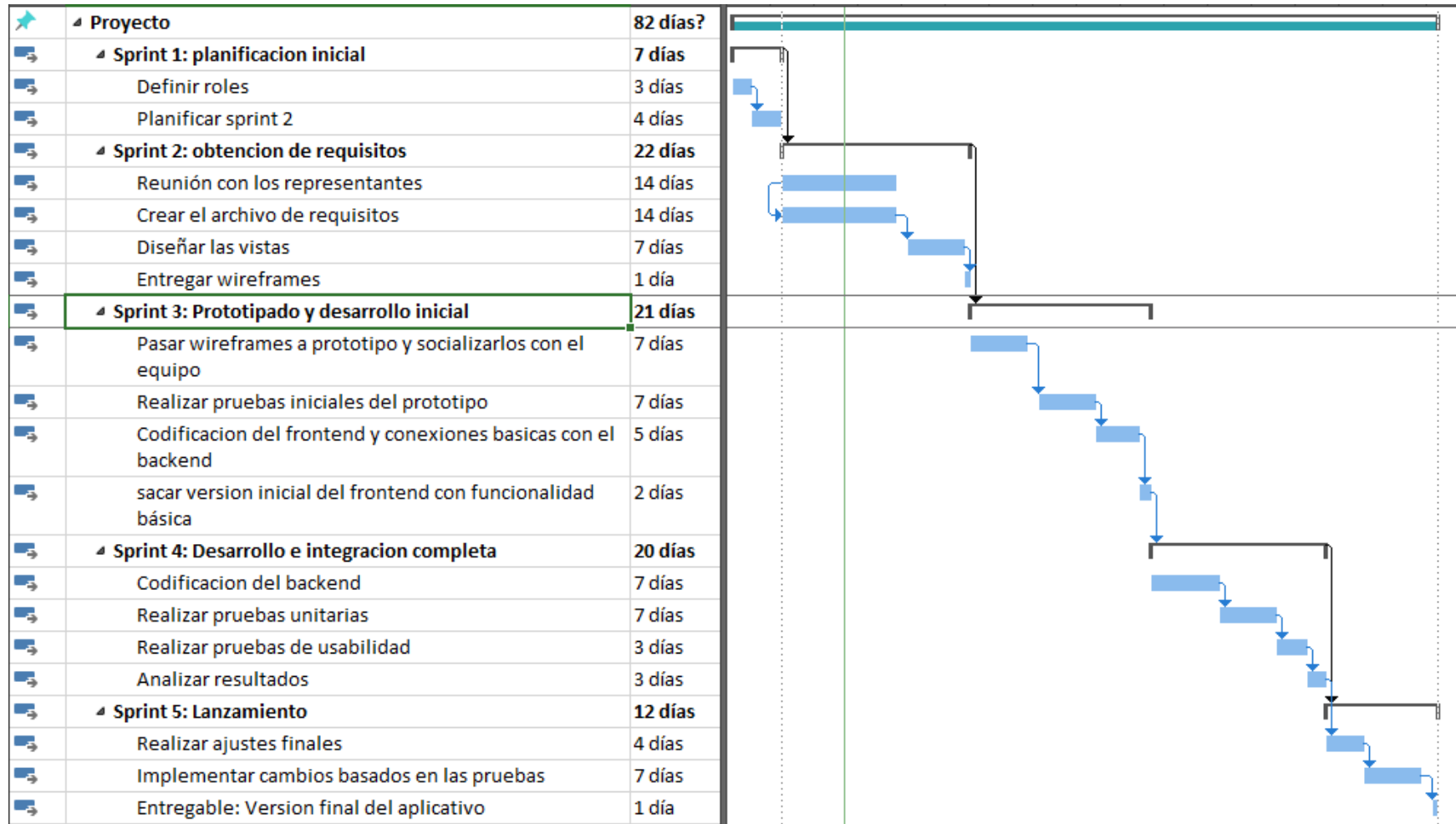
Zettler, K. (2024). ATLISSIAN. Obtenido de <https://www.atlassian.com/es/microservices/microservices-architecture/distributed-architecture#:~:text=Un%20sistema%20distribuido%20es%20un,error%20centrales%20de%20un%20sistema.>

Zettler, K. (2024). ATlassian. Obtenido de <https://www.atlassian.com/es/microservices/microservices-architecture/distributed-architecture#:~:text=Un%20sistema%20distribuido%20es%20un,error%20centrales%20de%20un%20sistema.>

Zulay, V. T. (2023). Aplicación web para gestionar la solicitud de tutorías académicas por los estudiantes en la Universidad Iberoamericana del Ecuador. Quito: UNIVERSIDAD IBEROAMERICANA DEL ECUADOR.

8 ANEXOS

Anexo 1: Cronograma de actividades.



Fuente: Los autores.

Anexo 2: Fotos entrevista a representantes, docentes de tutorías.



Fuente: los autores

Anexo 3: Entrevista a docentes representantes.

Para mejor manejo, las entrevistas fueron grabadas y luego transcritas de manera manual para un mejor análisis.

Docente: Leidy Ximena Cortés Velásquez.

1. ¿Qué tipo de autenticación cree usted que debería utilizarse?

Respuesta: Dentro del contexto de ser un aplicativo netamente académico y en un entorno como lo es nuestra universidad, creo que lo más acertado para una forma de autenticarse serían usuarios y contraseñas, que de por sí ya han sido creados y que hoy en día tienen los estudiantes con sus apellidos que son los que usan para acceder al portal.

2. Existen políticas específicas para la seguridad de las contraseñas (longitud mínima, caracteres especiales, etc.) ¿cree usted que deberíamos implementar alguna o algunas?

Respuesta: Por ese lado no veo algo de rigurosidad, siento que con el hecho de que cumpla con los caracteres mínimos y algún signo o carácter especial ya sería digo yo que suficiente.

3. En sentido de registro ¿Quién debería registrar a los estudiantes y docentes en el sistema?

Respuesta: Creo que sería mejor que lo tuviese otra persona, por cuestiones de orden y organización, alguien que pueda desempeñar digamos que un rol adicional,

coordinador o administrador si en dado caso se gestionará un rol como tal, así se evitan mal entendidos de duplicación de usuarios y se tiene mejor control de la cantidad real de las personas en plataforma.

4. ¿Qué información debería ser necesaria para registrar a cada usuario (nombre, correo electrónico, rol, etc.)?

Respuesta: Primer nombre, primer apellido, el correo institucional sería bueno que se maneje también ahí, el rol, pero ya eso en sentido de mantener el registro organizado y gestionado por otra persona como te dije antes.

5. ¿Qué rol debería tener la capacidad de activar o desactivar cuentas de usuarios, el administrador o algún otro rol?

Respuesta: Igual que lo del registro, es que siento que eso debería manejarse así por cuestiones de orden, no veo eso viable o bien visto de que se puedan activar o desactivar por sí mismo, o que tampoco sea algo que netamente deba realizar el docente.

6. ¿Qué tipo de datos deberían dejar ser modificados (nombres, roles, contraseña, etc.)?

Respuesta: Los básicos, ósea, los mismos que se pidieron para el registro de la persona, el primer nombre, el primer apellido, no sé si la contraseña porque si de algo estamos seguros es que empezarían a olvidarla y así empezar a modificarla a cada rato. Pero no debería ser algo que hiciera cada persona, eso debería hacerlo la persona que vaya a hacer los registros, las activaciones y eso.

7. ¿Quién debe ser responsable de activar o desactivar un curso en el sistema?

Respuesta: En esto si no sabría, diría yo que el coordinador de la carrera o bueno, eso debería ser del mismo rol que te digo, sea admin, sea coordinador, como le vayan a llamar, pero podría ser también una función para ese rol.

8. ¿cree usted que se debería poder filtrar o buscar usuarios según criterios específicos (nombre, rol, etc.)?

Respuesta: Pues no es algo que pese en este o no, pero si sería algo de comodidad en el momento en que el aplicativo tenga una población amplia en un futuro.

9. ¿En qué situaciones un docente optaría por eliminar a un estudiante de un curso?

Respuesta: En ninguna, como te digo, eso debería ser una función del rol de coordinador o administrador que vayan a colocar.

10. ¿Qué información sobre los estudiantes tendría que ser visible para el docente?

Respuesta: Nombre, apellido y el estado digo yo, pero como tales principales si o si, el nombre y el apellido.

11. ¿Qué información sobre los cursos debe ser visible para los estudiantes?

Respuesta: El nombre del curso y el docente, yo creo que con eso ya el estudiante puede guiarse.

12. ¿Quiénes serán los usuarios que necesitarán acceso real al sistema?

Respuesta: Pues en este caso serían, estudiantes, docentes y el rol que te digo, un adicional, pueden llamarlo coordinador o administrador, orientador, como mejor les parezca, pero que se encargue de las funciones que hablamos arriba.

13. ¿Es necesario el hecho de que se hagan validaciones para garantizar que los datos ingresados sean correctos?

Respuesta: No estaría de más, pero sino, no creo que sea algo que sea problema.

Docente: Callenata López Baleta.

1. ¿Qué tipo de autenticación cree usted que debería utilizarse?

Respuesta: Bueno, lo de usuario y contraseña obviamente, se les coloca que un fragmento de código que tome las primeras letras del primer nombre, primer apellido y que de ello se genere el usuario, ya lo que es contraseña pues se manejaría por primer ingreso una temporal o sino que cada estudiante tenga de contraseña su documento, pero ese tipo sería el mejor, usuario y contraseña.

2. Existen políticas específicas para la seguridad de las contraseñas (longitud mínima, caracteres especiales, etc.) ¿cree usted que deberíamos implementar alguna o algunas?

Respuesta: Hoy en día sí, hay una norma que nos habla de ello, pero si, que cumpla con los 8 caracteres mínimos, caracteres especiales, cifrado, que ya eso sería un toque de ustedes como programadores, pero si los básicos.

3. En sentido de registro ¿Quién debería registrar a los estudiantes y docentes en el sistema?

Respuesta: Administrador, eso es netamente función de administrador, por normativa y ética se puede decir así, eso es función netamente del administrador, para evitar malos registros y mantener un buen manejo de información en el programa.

4. ¿Qué información debería ser necesaria para registrar a cada usuario?

Respuesta: Como campos obligatorios para un registro siempre son, primer nombre, primer apellido, el correo, en este caso institucional para seguir con ese orden y el rol que va a desempeñar la persona dentro del programa o bueno, aplicativo, y una contraseña, ya con eso cumplirían con esa información, de ahí en adelante el campo que ustedes como programador quieran hacer obligatorio.

5. ¿Qué rol debería tener la capacidad de activar o desactivar cuentas de usuarios, el administrador o algún otro rol?

Respuesta: El administrador, eso es propio del administrador.

6. ¿Qué tipo de datos deberían dejar ser modificados?

Respuesta: Esos casi siempre son los mismos de registro, pero como tal esos campos para edición los asignan ustedes, hay programas que dejan editar todo, algunos solo los nombres y lo que es correo y eso no lo dejan modificar por seguridad.

7. ¿Quién debe ser responsable de activar o desactivar un curso en el sistema?

Respuesta: Administrador, todo eso que me dices de activar, editar, registrar, etc., eso es función netamente del administrador.

8. ¿Cree usted que se debería poder filtrar o buscar usuarios según criterios específicos (nombre, rol, etc.)?

Respuesta: No es algo que sea necesario, pero más que todo por comodidad de acceso a información, cuando hay un alto volumen es cuando se implementan esas cosas, pero son cositas digamos no tan importantes para el funcionamiento real del aplicativo siento yo.

9. ¿En qué situaciones un docente optaría por eliminar a un estudiante de un curso?

Respuesta: No, en ninguna, eso debería ser propio del administrador, pero no debería aun así existir una razón o como dices tu una situación para ello.

10. ¿Qué información sobre los estudiantes tendría que ser visible para el docente?

Respuesta: Yo creo que el nombre y el apellido, ya con eso uno identifica el estudiante si está o no en las tutorías.

11. ¿Qué información sobre los cursos debe ser visible para los estudiantes?

Respuesta: igual, el nombre del docente, el apellido del docente y el nombre de la asignatura.

12. ¿Quiénes serán los usuarios que necesitarán acceso real al sistema?

Respuesta: Estudiante, claramente, el docente y el administrador, ya con esos tres cubres los roles del software.

13. ¿Es necesario el hecho de que se hagan validaciones para garantizar que los datos ingresados sean correctos?

Respuesta: Casi siempre se colocan para validar que, si sea la contraseña, de que si se al usuario obviamente, pero pues ya eso es cuestión de normativa y de ustedes como programador si lo ven necesario y obligatorio para el aplicativo.

Anexo 4 Fotos entrevista Dirección de Programa



Anexo 5 Entrevista Dirección de Programa

Directora: Ing. Lizeth Badillo Durán.

- 1. ¿Cómo es el funcionamiento hoy en día de las tutorías dentro de la seccional en la carrera de ingeniería de sistemas?**

Respuesta: Al día de hoy, las tutorías se brindan de manera presencial dentro de los espacios en la seccional, se especifica un horario claramente contrario a las clases de esas mismas asignaturas que son asignadas, por ejemplo, si calculo 2 se está dictando en la jornada de la mañana digamos que 6am u 8am las tutorías entonces estarían estipuladas y asignadas en la tarde cosa que el estudiante tenga el tiempo de organizarse y pueda asistir a ellas.

- 2. ¿Cómo realiza actualmente el seguimiento de los estudiantes que asisten a tutorías?**

Respuesta: En las tutorías, cada docente cuenta con una lista digámoslo así, en blanco claramente, donde los estudiantes la llenan al asistir con datos como su nombre, carrera, semestre y su correo institucional.

- 3. ¿Lleva algún tipo de registro (manual o digital) de los estudiantes que atiende en tutorías?**

Respuesta: El listado que antes hablábamos, es como nuestro registro propio para sacar luego un análisis de cuantos estudiantes realmente están asistiendo a esos espacios asignados de esas asignaturas de tutorías.

- 4. ¿Qué asignaturas hoy en día priman o tienen asignados estos espacios de tutorías?**

Respuesta: Hoy tenemos asignadas en tutorías las asignaturas Calculo 1, 2 y 3, por el lado de las competencias matemáticas, por el lado de lógica y programación

tenemos fundamentos de programación, programación 1, 2 y 3. Que al día de hoy son como las asignaturas que más tienen índice de reprobación.

5. A nivel del desarrollo de un aplicativo como este ¿Cuáles deberían ser los roles indispensables dentro del funcionamiento del software?

Respuesta: Bueno, roles serían estudiantes obviamente ya que ellos son los principales usuarios para el software seguido de los docentes, se convertiría en una herramienta propia para el docente sin embargo no quitaría el espacio presencial de estas tutorías ya que ustedes mismos lo dicen, esto es una herramienta de apoyo y por principio claro de todo software debería tener un administrador, alguien que tenga control como su nombre lo dice administrativo del aplicativo.

6. ¿Qué información sobre los estudiantes tendría que ser visible para el docente?

Respuesta: Nombre, apellido... creo que con eso ya el docente identificaría al estudiante para saber así si está o no registrado en las tutorías de la asignatura.

7. ¿Qué rol debería tener la capacidad de activar o desactivar cuentas de usuarios, el administrador o algún otro rol?

Respuesta: Eso ya debería ser una función propia y sola del administrador, eso es primordial para generar y mantener un buen control del funcionamiento del aplicativo.

8. ¿Quién debería registrar a los estudiantes y docentes en el sistema?

Respuesta: Administrador, también es función propia del rol que debe cumplir el administrador así se mantendría un control de los estudiantes que están dentro de las tutorías.

9. ¿Qué información debería ser necesaria para registrar a cada usuario?

Respuesta: Los datos obligatorios para un registro en softwares institucionales siempre son primer nombre, primer apellido, el correo institucional y una contraseña, ya con eso se mantiene un registro correcto de los estudiantes.

10. De los datos anteriormente mencionados ¿Cuáles deberían dejar ser modificados (nombres, roles, contraseña, etc.)?

Respuesta: Debería hacerlos en administrador claro está, pero yo digo que esos mismos, nombre, apellido y la contraseña, si es caso el correo mismo también sueles suceder que por errores de digitación se dan normalmente los malos registros.