

PROYECTO DE GRADO APLICADO

IMPLEMENTACIÓN DE UN MÓDULO DE CALIFICACIONES
PARA INSTITUCIONES DE EDUCACIÓN BÁSICA SECUNDARIA Y MEDIA

SERGEI LEONARDO MONTOYA CARO

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA UNAD
INGENIERÍA DE SISTEMAS
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA
BOGOTÁ DC, COLOMBIA
2021

IMPLEMENTACIÓN DE UN MÓDULO DE CALIFICACIONES
PARA INSTITUCIONES DE EDUCACIÓN BÁSICA SECUNDARIA Y MEDIA

SERGEI LEONARDO MONTOYA CARO

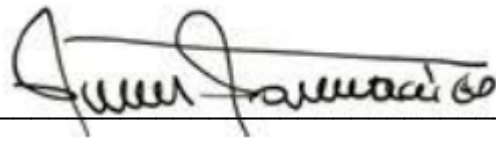
Trabajo de grado presentado como requisito
para optar al título de ingeniero de sistemas

Luis Abdul Samaca Chavarriaga
Magíster en E-Learning y Redes Sociales

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA UNAD
INGENIERÍA DE SISTEMAS
ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA
BOGOTÁ DC, COLOMBIA
2021

NOTA DE ACEPTACIÓN

Aprobado en cumplimiento de los requisitos exigidos por la Universidad Nacional Abierta y A Distancia UNAD para optar por el título de Ingeniero de Sistemas.



Director de la opción de grado

Luis Abdul Samaca Chavarriaga



Jurado

Yina Alexandra Gonzalez Sanabria

Bogotá DC, 03 de diciembre de 2021

DECLARACIÓN DE DERECHOS DE PROPIEDAD INTELECTUAL

Los autores de la presente propuesta manifestamos que conocemos el contenido del Acuerdo 06 de 2008, Estatuto de Propiedad Intelectual de la UNAD, Artículo 39 referente a la cesión voluntaria y libre de los derechos de propiedad intelectual de los productos generados a partir de la presente propuesta. Asimismo, conocemos el contenido del Artículo 40 del mismo Acuerdo, relacionado con la autorización de uso del trabajo para fines de consulta y mención en los catálogos bibliográficos de la UNAD.

DEDICATORIA

La culminación de mi carrera profesional no habría sido posible sin el apoyo de mi familia, por esto les dedico mi título. Especialmente, expreso mi más profundo agradecimiento a:

Mi abuela, Transito Caro y mi madre, Patricia Caro, dos mujeres incansables que, con su amor y su esfuerzo me han ayudado a cumplir mis sueños, siempre buscaron la manera de sonreír ante las dificultades y ver la calma después de la tormenta, me enseñaron a perseverar, a no rendirme y luchar por mis metas.

Mi hija, Samantha Montoya quien se convirtió en el motor de mi vida, desde el día que me enteré que venía en camino, yo sabía que mi vida cambiaría abruptamente, y así fue, ella llegó para hacer de mi mundo un lugar mejor y hacer de mí, una mejor persona.

CONTENIDO

Lista de figuras	9
Lista de tablas	11
Lista de ecuaciones	12
Lista de abreviaturas	13
Lista de anexos	14
GLOSARIO	15
RESUMEN	16
INTRODUCCIÓN	18
1. PLANTEAMIENTO DEL PROBLEMA	19
1.1. Formulación De La Pregunta	19
2. JUSTIFICACIÓN	20
3. OBJETIVOS	21
3.1. Objetivo General	21
3.2. Objetivos Específicos	21
4. MARCO DE REFERENCIA	22
4.1. Marco Teórico	23
4.1.1. Evolución del software	23
4.1.2. Ingeniería del software	24
4.1.3. Fases de desarrollo en la ingeniería de software	24
4.1.4. Modelos para el diseño de software	25
4.1.5. Metodologías para el desarrollo de software	25
4.1.6. Paradigmas en el desarrollo de software	26
4.1.7. Boceto del software	27
4.1.8. Inclusión de interfaces gráficas en el software	28
4.2. Marco Conceptual	29
4.2.1. Programación orientada a objetos	29
4.2.2. Bases de datos relacionales	29
4.2.3. Aplicativo web	29
4.2.4. Gestión escolar	30

4.2.5. Procesos académicos	30
4.2.6. Procesos disciplinarios	30
4.2.7. Procesos administrativos	30
4.2.8. Plataforma	31
4.2.9. Módulo de calificaciones	31
4.3. Marco Contextual	31
5. DELIMITACIONES	33
5.1. Delimitación Tecnológica	33
5.2. Delimitación Cognitiva	33
5.3. Delimitación Temporal	34
5.4. Delimitación Financiera	34
5.5. Delimitación Legal	35
6. METODOLOGÍA	36
6.1. Modelo De Desarrollo En Cascada	36
6.2. Arquitectura De Software Modelo Vista Controlador	37
6.2.1. Patrones de diseño	37
6.3. Tecnologías Web	38
6.4. Seguridad informática	39
7. DISEÑO Y DESARROLLO DEL SOFTWARE	41
7.1. Modelo De Desarrollo	41
7.2. Análisis	41
7.2.1. Identificación de actores	42
7.2.2. Identificación de roles y permisos	43
7.2.3. Identificación de requerimientos	44
7.3. Diseño	45
7.3.1. Modelo relacional de la base de datos	46
7.3.2. Modelo lógico de la base de datos	47
7.3.3. Diagramas de casos de uso	48
7.3.4. Diagramas de actividades	51
7.3.5. Diagrama de clases	55
7.3.6. Diagrama de secuencia	56

7.4. Implementación	56
7.4.1. Diseño de interfaz de usuario.....	56
7.5. Prueba	61
7.5.1. Software versión beta	62
7.6. Servicio.....	66
7.6.1. Matriz de evaluación	66
7.6.2. Métricas de evaluación	67
7.6.3. Plan de evaluación	68
8. CONCLUSIONES	71
9. RECOMENDACIONES.....	72
10. REFERENCIAS BIBLIOGRÁFICAS	73
11. ANEXOS.....	77

LISTA DE FIGURAS

<u>Figura 1. Modelo relacional DB</u>	46
<u>Figura 2. Modelo lógico DB</u>	47
<u>Figura 3. Caso de uso: Autenticación</u>	48
<u>Figura 4. Caso de uso: Gestión de usuarios</u>	48
<u>Figura 5. Caso de uso: Calificación</u>	49
<u>Figura 6. Caso de uso: Calificar</u>	49
<u>Figura 7. Caso de uso: Matricular</u>	50
<u>Figura 8. Modelo en cascada</u>	41
<u>Figura 9. Wireframe: Inicio de sesión</u>	57
<u>Figura 10. Wireframe: Menú de herramientas</u>	57
<u>Figura 11. Wireframe: Informe de calificaciones</u>	58
<u>Figura 12. Wireframe: Barras de progreso</u>	58
<u>Figura 13. Wireframe: Edición de datos de usuario</u>	59
<u>Figura 14. Wireframe: Listado de usuarios</u>	60
<u>Figura 15. Wireframe: Registro de datos</u>	60
<u>Figura 16. Wireframe: Eliminación de registros</u>	61
<u>Figura 17. Inicio de sesión</u>	62
<u>Figura 18. Menú de herramientas</u>	63
<u>Figura 19. Barras de progreso</u>	64
<u>Figura 20. Edición de datos de usuario</u>	64
<u>Figura 21. Listado de usuarios</u>	65
<u>Figura 22. Registro de datos</u>	65
<u>Figura 23. Eliminación de registros</u>	66
<u>Figura 24. Diagrama de actividad: Autenticación</u>	51
<u>Figura 25. Diagrama de actividad: Gestión de usuarios</u>	52
<u>Figura 26. Diagrama de actividad: Calificación</u>	53
<u>Figura 27. Diagrama de actividad: Matricular</u>	54
<u>Figura 28. Diagrama de clases</u>	55

<u>Figura 29. Relación entre los módulos del MVC</u>	37
<u>Figura 30. Modelo cliente-servidor</u>	39
<u>Figura 31. Diagrama de secuencia</u>	56
<u>Figura 32. Interrelación entre los elementos del patrón MCV</u>	38

LISTA DE TABLAS

<u>Tabla 1. Cronograma inicial</u>	34
<u>Tabla 2. Presupuesto inicial</u>	35
<u>Tabla 3. Identificación de actores</u>	42
<u>Tabla 4. Roles y permisos para el usuario final</u>	43
<u>Tabla 5. Requerimientos funcionales y no funcionales</u>	44
<u>Tabla 6. Matriz de evaluación del software</u>	67
<u>Tabla 7. Métricas de evaluación</u>	67
<u>Tabla 8. Plan de pruebas</u>	69
<u>Tabla 9. Pruebas no funcionales</u>	69

LISTA DE ECUACIONES

Ecuaciones generales

76

LISTA DE ABREVIATURAS

Abreviaturas generales

76

LISTA DE ANEXOS

<u>Anexo 1. Encuesta de levantamiento de información</u>	77
<u>Anexo 2. Matriz de descripción de actividades</u>	78
<u>Anexo 3. Manual técnico</u>	78
<u>Anexo 4. Manual de usuario</u>	78

GLOSARIO

Internet: Conjunto de redes de comunicaciones interconectadas, descentralizado, que utiliza protocolos TCP/IP para su funcionamiento, lo que garantiza que las redes físicas heterogéneas que la componen constituyen una red lógica única de alcance global.

Servidor: Dispositivo informático o un conjunto de estos, que permite centralizar servicio para que sean utilizados y accesibles, atendiendo peticiones de otros programas o dispositivos llamados clientes.

Cliente: Es un dispositivo informático o software que accede a un servidor y recupera servicios o datos de este, por medio de peticiones estandarizadas y transmitidas a estos clientes.

Protocolo de internet: Normas estandarizadas que permiten enviar y recibir datos entre varios ordenadores o dispositivos de una manera universal. Por medio de estos protocolos es posible controlar y permitir la comunicación y transferencia de datos entre dos puntos.

Lenguaje de programación: Son lenguajes formales compuestos por una serie de instrucciones que le permite a un programador escribir un conjunto de órdenes, acciones consecutivas, datos o algoritmos en la creación de programas que controlan de manera lógica y física el comportamiento de una máquina.

RESUMEN

El presente proyecto aplicado tiene como objetivo principal, presentar el desarrollo de un módulo de calificaciones, por medio de una solución de software, de manera que puede ser integrado al sistema de gestión y administración integral de los diferentes procesos académicos, disciplinarios y administrativos, existentes o en desarrollo, para ser implementado y usado en instituciones educativas, enfocado en el nivel de educación secundaria, básica y media.

Se desarrolla este software teniendo en cuenta una serie de exigencias, requerimientos, condiciones y necesidades de las instituciones educativas, donde se definen los diferentes grados, niveles y asignaturas, así mismo permite vincular a los docentes con sus asignaturas y grados correspondientes y a los estudiantes con los grados o niveles y asignaturas a cursar, gestionar las calificaciones de las diversas asignaturas correspondientes al nivel en curso y el histórico de las asignaturas y grados previamente cursados, parametrizando y definiendo el tipo de calificación como numérica de 0 a 5, todo esto configurado y establecido por medio de una base de datos relacional.

Se desarrolla este software usando tecnologías web, buscando que sea accesible para cualquier usuario y desde cualquier lugar, siendo alojado en la nube o un servidor web en internet y que tenga un alto índice de compatibilidad y escalabilidad, lo que permite facilitar el acceso a la información, facilita la integración y permite la creación de otros módulos de acuerdo con las necesidades de la institución.

El desarrollo de este proyecto facilita el manejo y la centralización de la información, especialmente, teniendo en cuenta que, en los últimos años, la información y el acceso a esta, se ha convertido en uno de los factores claves para el crecimiento de cualquier organización, por lo que, adaptarse a las nuevas tecnologías para su manejo y gestión, es una necesidad latente y urgente.

Este proyecto tecnológico, está enfocado en el mejoramiento de los procesos que componen a las instituciones educativas, para lo que se busca desarrollar uno de los diferentes módulos, que integrados logren subsanar todas las necesidades de la gestión escolar, administrativa, académica y disciplinaria de las instituciones educativas.

Palabras claves: plataforma, gestión escolar, procesos académicos, procesos disciplinarios, procesos administrativos, módulo de calificaciones.

INTRODUCCIÓN

El uso de soluciones de software para la administración de la información ha sido un tema de gran relevancia en diferentes ámbitos, especialmente a nivel corporativo o empresarial, sin embargo, esta misma digitalización ha permitido evidenciar la importancia del uso de software para gestionar los diferentes procesos administrativos y educativos en las instituciones de educación, adicionalmente el uso de este tipo de soluciones informáticas han ayudado en el proceso de aprendizaje en los estudiantes, gracias a que brinda la oportunidad de realizar un monitoreo y control muy preciso, por medio del análisis de la información. De hecho, es importante notar que el uso de software educativo, tanto de uso libre como de uso comercial y para satisfacer diferentes tipos de necesidades ha presentado un gran crecimiento en las últimas décadas.

Los gobiernos en prácticamente todo el mundo se encuentran gestionando campañas, que buscan fomentar la migración de procesos físicos documentales a soluciones digitales, lo que a su vez busca disminuir los índices de afectación al medio ambiente y también una disminución en el uso de recursos de oficina como papel, entre otros, adicionalmente, se evita el uso de grandes espacios para el almacenamiento del archivo, lo que puede significar menores costos para las organizaciones.

Aunque las grandes corporaciones y empresas son pioneras y han tenido mayor participación en la era digital, la aceptación del uso de software en las instituciones educativas ha crecido significativamente, gracias a que han logrado comprender que el uso de programas informáticos es un complemento a los diferentes procesos tanto educativos, como administrativos.

Por esto, el presente proyecto busca brindar solución a una serie de necesidades de las instituciones educativas, por medio del desarrollo de un programa informático enfocado en la gestión académica para la educación secundaria, en los niveles básica y media y con esto incentivar el uso del software educativo, así expandir los beneficios de las tecnologías de la información.

1. PLANTEAMIENTO DEL PROBLEMA

Por lo general las entidades de educación carecen de alguna solución de software que sea integral y centralizado, que permita gestionar los diferentes procesos de estas instituciones, en los ámbitos; académico, disciplinario y administrativo, así como la comunicación oportuna de los diferentes actores del proceso educativo.

Especialmente se hace evidente esta carencia en los grados de educación secundaria, ya que en la básica primaria no se considera tan necesario debido a las cortas edades de los estudiantes y siendo las entidades de educación superior las que generalmente hacen uso de software para facilitar algunos procesos educativos, sin embargo, estas soluciones no manejan de manera integral y centralizada todo el volumen informático de las instituciones.

Las organizaciones que manejan una gran cantidad de datos enfrentan diferentes retos, por lo que tener una solución que permita centralizar la información y gestionar de manera integral todos los procesos de las instituciones además de generar un canal oficial de comunicaciones integrado, se traduce en un factor clave de su crecimiento y optimiza todo el funcionamiento, generando mayor efectividad y eficiencia, economizando y aprovechando de mejor manera los recursos.

1.1. Formulación De La Pregunta

¿Cómo implementar un módulo de calificaciones para una institución educativa en los niveles de básica secundaria y media?

2. JUSTIFICACIÓN

La necesidad del manejo de la información de las organizaciones es cambiante y evoluciona a la par de las dinámicas digitales, por esto, las instituciones educativas de básica y media tienen la necesidad de implementar soluciones informáticas que optimicen y faciliten diferentes procesos, con el propósito de generar mayor eficiencia y calidad a nivel general.

Actualmente hay diferentes herramientas en el mercado que permiten manejar y gestionar el volumen de información para diferentes tipos de organizaciones, sin embargo, la oferta existente no se encuentra focalizada en las instituciones educativas, por lo que si una de estas instituciones requiere digitalizar todos sus procesos, se verá obligada a adquirir herramientas que en la mayoría de ocasiones no cumplen con las necesidades, requerimientos y expectativas de estas instituciones, lo que se traduce en mayores costos y además hace esta tarea más compleja, por esto y gracias al presente proyecto aplicado se satisface la necesidad de este tipo de organizaciones donde se centralizan todos los procesos y la información de manera integral con un software desarrollado y pensado para las entidades educativas.

Por esto, por medio del presente proyecto se desarrolla el módulo que se considera como el más oportuno e importante, el módulo que se encargaría de gestionar los procesos académicos, las calificaciones e informes individuales e institucionales, se debe desarrollar de manera modular, siendo el primer módulo, creado a modo de piloto para ser integrado a un sistema integrado de gestión de los procesos institucionales.

3. OBJETIVOS

3.1. Objetivo General

Desarrollar una solución informática para gestionar las calificaciones y los procesos académicos de las entidades educativas de básica secundaria y media, de manera modular y priorizando la accesibilidad, la integridad de la información y la seguridad.

3.2. Objetivos Específicos

- ✓ Analizar las necesidades del público objetivo de las instituciones de educación en el nivel de secundaria, de tal manera que se puedan identificar los requerimientos funcionales y no funcionales con los que debe contar el software.
- ✓ Planificar el diseño, estructura y herramientas, para el desarrollo del módulo de gestión académica y calificaciones para los estudiantes de secundaria.
- ✓ Codificar el módulo de gestión académica y calificaciones para instituciones de básica y media, siguiendo los lineamientos del diseño designado, de acuerdo con la información obtenida del análisis y la planeación, integrando las herramientas necesarias para cumplir con los requerimientos acordados.
- ✓ Implementar un plan de pruebas y evaluación del funcionamiento del software.

4. MARCO DE REFERENCIA

Hay una gran cantidad de software o herramientas basadas en software, tanto privativas como libres con un altísimo potencial para descubrir y explotar muchas habilidades y capacidades de los estudiantes, así como complementar el proceso de enseñanza-aprendizaje, lo que aporta de manera significativa a la satisfacción de los usuarios, como explica Luis Pimiento en su artículo La Evolución del Software: “La satisfacción de las necesidades de los usuarios seguirá dependiendo del software, ya sea que le permita utilizar herramientas que faciliten el trabajo o tareas diarias o software que permita editar y gestionar todo aquello que se pueda incluir en la tecnología informática.”, sin embargo, cuando se desarrolla software, se hace para cumplir ciertos requerimientos específicos, por esto, podemos encontrar diferentes herramientas que sumadas entre sí, pueden cumplir con el conjunto de necesidades de un usuario de manera integral, pero esto supone generalmente mayores costos de licenciamiento, soporte o administración, así sean herramientas privativas o libres.

Aunque en el mercado actual encontramos soluciones de software parciales, que cuentan con algunas herramientas o funciones de gestión, son por lo general insuficientes para cubrir todos los procesos fundamentales y de una manera transversal e integral, especialmente en el país, donde se ha dado prioridad a las grandes empresas y poco se han enfocado esfuerzos en las entidades de educación secundaria, por esto nace la necesidad urgente de cubrir todos estos requerimientos para las organizaciones de este sector.

Por esto nace la idea de diseñar, desarrollar e implementar un sistema modular integral de gestión escolar, donde se define iniciar con la creación del módulo de gestión académica y de calificaciones inicialmente y como modelo piloto. Finalmente, y teniendo en cuenta la reflexión de Richardas (2015) que nos orienta acerca de los diferentes patrones de arquitectura en el desarrollo de software, se define para el desarrollo del proyecto, la arquitectura MVC, debido a que, para la intencionalidad y las características de este desarrollo, es la que mejor se adapta, inicialmente como base para diseñar la separación de las diferentes capas que componen la aplicación; esto es, la capa de interfaz, la capa de datos y la capa lógica.

4.1. Marco Teórico

4.1.1. Evolución del software

En la actualidad el software está presente en prácticamente todos los aspectos de la sociedad, una ingente cantidad de organizaciones en diferentes sectores depende del uso de este para su funcionamiento, sin embargo, no siempre fue así, realmente el ámbito del desarrollo de software ha tenido un auge y un crecimiento exponencial a través de las últimas décadas, donde su expansión ha sido absoluta y su evolución puede resumirse de la siguiente manera:

- **1975 - 1980:** Surgen los primeros programas de propósito general y específico, estos son de apoyo en los diferentes procesos de las instituciones educativas.
- **1985:** Surgen diferentes desarrollos que buscan potencializar aún más los programas a nivel educativo y afianzar más el soporte que la tecnología aporta a los procesos académicos.
- **1990 - 1995:** Surgen diferentes programas y herramientas que hacen el uso de internet para su funcionamiento.
- **2000:** En esta era nace el aprendizaje virtual, programas de aprendizaje vía online, plataformas virtuales.
- **2005:** Surgen las plataformas sociales, es decir herramientas informáticas virtuales; como blogs, wikis y herramientas informáticas que fomentan el aprendizaje colaborativo.
- **2010:** Se expande aún más el uso del software libre, surgen plataformas virtuales y software de comunicación virtual, donde ciertos grupos interactúan entre sí exponiendo temas comunes, frecuentemente en temas de investigación, educación, científicos, medicina, entre otros aspectos.
- **2015:** El desarrollo del software crece vertiginosamente con la evolución de las plataformas móviles y actualmente la digitalización y la globalización han incentivado el uso de aplicativos informáticos para todo tipo de organizaciones.

4.1.2. Ingeniería del software

Por medio de esta rama de las ciencias es posible crear software confiable y de calidad, haciendo uso de métodos y técnicas de ingeniería, como lo expone Bohem: “es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos”.

La creación de software es un proceso esencialmente creativo que busca sistematizar, optimizar o automatizar diferentes procesos, así como gestionar, procesar o manipular datos e información para diversos tipos de organizaciones y en muchos ámbitos, haciendo uso de herramientas, técnicas y metodologías de la manera más eficiente para la obtención de resultados óptimos y la consecución de objetivos, mediante la aplicación de ingeniería al diseño y desarrollo del software.

4.1.3. Fases de desarrollo en la ingeniería de software

La ingeniería de software es también conocida como desarrollo o producción de software y puede dividirse en tres partes fundamentalmente:

- **Definición:** Esta fase, se centra en la concepción de la idea, se identifica que se requiere hacer y con qué finalidad, se define el comportamiento, las funcionalidades, entre otros aspectos, basándose en una serie de datos obtenidos, procesados y analizados en el levantamiento de la información.
- **Desarrollo:** Esta fase, se centra en el diseño estructural de los datos, identificando el comportamiento y la manera en que se implementan las funciones dentro de la arquitectura del software.
- **Mantenimiento:** Esta fase, se centra en el control y monitoreo del software diseñado, corrigiendo errores o realizando mejoras de acuerdo a las necesidades y requerimientos cambiantes por el contexto y el cliente.

4.1.4. Modelos para el diseño de software

Definir métodos o modelos de diseño de software permite tener un plan de inicio a final, y permite en cada una de sus etapas saber qué es lo que se va a hacer o que es lo que se debe hacer en cada proceso para lograr los resultados requeridos de manera óptima. Hay diferentes modelos y entre ellos se analizan los siguientes:

- ❖ **Modelo lineal secuencial:** Este modelo tiene un enfoque sistemático y secuencial, que progresa por medio de una serie de etapas; análisis, diseño, codificación, pruebas y mantenimiento, este modelo también es conocido como ciclo de vida básico o modelo cascada.
- ❖ **Modelo de prototipado:** Este modelo presenta un enfoque más dinámico, inicia por la recolección de requisitos y definición de objetivos globales del software acordados con el cliente, posteriormente se diseña al menos un prototipo que se evalúa con el cliente y de manera conjunta y reiterativa se definen cambios para afinar detalles y obtener el mejor resultado.
- ❖ **Programación extrema:** Este modelo busca agilizar el proceso de desarrollo, por medio de la potencialización de las relaciones interpersonales, con estrategias de trabajo colaborativo y teniendo retroalimentación activa por parte del cliente.

Posteriormente a la evaluación de las diferentes alternativas y modelos de desarrollo propuestos, se decide seleccionar el modelo lineal secuencial, también conocido como desarrollo en cascada, teniendo en cuenta que por medio de este modelo se puede definir una serie de etapas que permite planificar y controlar de mejor manera el proceso de desarrollo del software.

4.1.5. Metodologías para el desarrollo de software

Usar una metodología para el desarrollo o programación de una solución basada en software es fundamental, debido a que permite optimizar recursos, minimizar los riesgos, optimizar el tiempo de desarrollo del proyecto, definiendo procesos organizados y eficientes, gracias a las diferentes técnicas y métodos organizativos que las componen.

Seleccionar la metodología más adecuada para un proyecto, dependerá principalmente de las necesidades propias de cada desarrollo, en la actualidad hay dos grupos principales de metodologías para el desarrollo de software:

- Metodologías tradicionales: Se caracterizan principalmente por su rigidez en la definición de los requisitos desde el inicio del proyecto, en las metodologías tradicionales los ciclos son poco flexibles, no permiten cambios sobre la marcha, su organización es lineal, es decir, se definen etapas y cada etapa sucede posterior a la finalización de la etapa anterior, las principales metodologías clásicas o tradicionales son:
 - Waterfall, también llamada Cascada.
 - Prototipado.
 - Espiral.
 - Incremental.
 - RAD, por sus siglas Rapid Application Development.
- Metodologías ágiles: Son las más utilizadas en la actualidad, sin embargo, las metodologías tradicionales son también ampliamente utilizadas, ya que cada proyecto tiene sus propias necesidades y condiciones, las metodologías ágiles propenden por dinamizar el desarrollo gracias a su flexibilidad y agilidad, basándose en la metodología incremental, donde se agregan pequeños cambios y funciones a la aplicación final en cada ciclo, en lugar de realizar cambios grandes, lo que permite tener ciclos más cortos y rápidos, así mismo se van probando sus funciones bajo la marcha, logrando evaluar los avances casi en tiempo real, realizando correcciones y agregando nuevas funciones de acuerdo con las necesidades, esto también permite dividir el trabajo en grupos independientes, las principales metodologías ágiles son:
 - Kanban.
 - Scrum.
 - Lean
 - Programación extrema, también llamada XP.

4.1.6. Paradigmas en el desarrollo de software

Para que una máquina o una computadora pueda realizar una tarea u operación, esta debe ser programada y cargada en su memoria, para lo que se debe definir un algoritmo apropiado, el que debe ser expresado en un lenguaje que la máquina pueda comprender, generalmente en un sistema binario.

Al enfoque del proceso de programación se le conoce como paradigma de programación, actualmente se reconocen principalmente los siguientes:

- **Paradigma imperativo, también llamado paradigma por procedimientos:** Este es el más utilizado desde prácticamente los inicios de la programación y en la actualidad, está enfocado en las instrucciones y estrechamente ligado con la arquitectura de la máquina, por lo que puede ser implementado de manera eficiente, los principales lenguajes son C, Pascal y Basic.
- **Paradigma funcional:** Se caracteriza por el uso de expresiones y funciones, en este paradigma un programa es un conjunto de funciones compuestas por funciones más simples, donde una función puede llamar a otra y el resultado de una función puede usarse como argumento de otra función, el lenguaje más reconocido de este paradigma es Lisp.
- **Paradigma declarativo, también conocido como paradigma de programación lógica:** Basado en la idea de que un programa implementa relaciones antes que correspondencias, es decir que, las relaciones son más generales y fundamentales que las correspondencias, este paradigma de programación es de más alto nivel que la funcional y la imperativa, el lenguaje más reconocido es Prolog.
- **Paradigma orientado a objetos:** Se basa en los conceptos de clases de objetos y objetos, donde un objeto puede ensamblarse con un conjunto de operaciones que se definen para estos o les pertenecen y una clase puede identificarse como un conjunto de objetos que comparten una serie de operaciones, este paradigma ha alcanzado gran popularidad a día de hoy y algunos de los lenguajes más reconocidos son Java y C++.

4.1.7. Boceto del software

En las etapas iniciales de producción de un software, encontramos la etapa de producción del boceto, o conocido ampliamente como el wireframe, éste, consiste en el diseño de la estructura general del aplicativo, generalmente sin logotipos o tipografías, debido a que en esta etapa no se busca mostrar el aspecto visual final, lo que se busca es identificar la estructura, el orden o la composición general de

cada uno de los elementos que componen la interfaz del aplicativo, por medio de la que va a interactuar el usuario con las herramientas del software, para diseñar un wireframe de manera correcta, su diseño se debe focalizar en la estructura y la jerarquía de los elementos, por lo general se diseñan en blanco y negro o escala de grises, los tonos más neutros posibles, buscando centrar la atención en la estructura en sí.

4.1.8. Inclusión de interfaces gráficas en el software

La interfaz gráfica de usuario es un software base que permite la interacción del usuario por medio de elementos gráficos como botones, ventanas, iconos o enlaces, entre otros. La inclusión de la primera interfaz de usuario grafica en una maquina se atribuye a Xerox, con su computadora Xerox Alto a principios de los 70, posteriormente se masifico el uso e inclusión de interfaces graficas en los sistemas operativos de los computadores de uso doméstico y corporativo por parte de Microsoft y Apple, con el auge de la incorporación de gráficos en la interacción de los usuarios con las maquinas, los desarrolladores de software y grandes corporaciones y marcas que se dedicaban a crear soluciones de software también apostaron por implementar esta tecnología, debido a las grandes ventajas que suponía acercar a los usuarios de una manera sencilla al complejo lenguaje de las maquinas.

Esto, significo un vertiginoso avance en el consumo y desarrollo de la tecnología, que aún vemos hoy y que seguirá creciendo en los próximos años, gracias a que permitió disminuir significativamente las brechas que separaban a la tecnología del grueso de los consumidores y minimizó las curvas de aprendizaje necesarias para el uso de los computadores de la época, que funcionaban a base de comandos que se ingresaban por medio de líneas de texto.

Fue tan importante la inclusión de interfaces graficas de usuarios, que, a día de hoy, es una cualidad de facto en la producción de software a excepción de algunos desarrollos muy específicos y que definen índices diminutos comparados con el grueso del software que se desarrolla en la actualidad.

4.2. Marco Conceptual

La producción de software es una ciencia compleja y utiliza una serie de criterios, estándares y fundamentos de la ingeniería de software para crear productos industriales y a su vez utiliza bases de la ingeniería como métodos, técnicas y herramientas para desarrollar productos innovadores regidos por metodologías y buenas prácticas, adicionalmente y para el desarrollo del proyecto actual, es necesario comprender ciertos conceptos claves que permiten identificar el público objetivo y sus necesidades.

4.2.1. Programación orientada a objetos

Cuando se busca solucionar una problemática por medio de un software, una de las primeras decisiones que se deben tomar, es seleccionar tipo de modelo por medio del cual se va a codificar el aplicativo, en el caso de este proyecto, se define el uso del modelo de programación orientada a objetos, debido a que favorece el desarrollo e implementación de sistemas complejos, los sistemas desarrollados con este modelo de programación son construidos desde objetos auto-contenidos que interactúan entre ellos, lo que genera una modularidad que favorece su diseño, implementación y mantenimiento.

4.2.2. Bases de datos relacionales

La tecnología moderna y más ampliamente utilizada para las bases de datos, es el modelo relacional, este modelo se encarga de tres aspectos de la información fundamentalmente; la estructura de datos, la manipulación de datos y la integridad de datos, para lo que es fundamental identificar los términos estructurales más importantes, como los atributos, la cardinalidad, el grado, el dominio, la clave primaria, y para operar los datos se usa el lenguaje estructurado SQL.

4.2.3. Aplicativo web

Las herramientas o el software que puede ser utilizado por los usuarios accediendo a un servidor web a través de internet o de una intranet por medio de un navegador, es decir, este tipo de software se codifica en uno o varios lenguajes que pueden ser interpretados por los navegadores web, los aplicativos web

suponen una gran cantidad de ventajas, permite la centralización de los servicios y la información, puede ser ejecutado en múltiples plataformas, independientemente del sistema operativo, posee una gran facilidad para actualizar y mantener los aplicativos, en comparación con el software de escritorio y permite llegar a muchos más usuarios.

4.2.4. Gestión escolar

Es un proceso sistemático, que busca fortalecer las instituciones educativas y sus proyectos, facilitando los diferentes procesos pedagógicos, administrativos, disciplinarios, directivos y de seguimiento y control, haciendo uso de la información para generar valor y políticas de calidad.

4.2.5. Procesos académicos

Es la gestión de los diferentes procesos que afectan de manera directa el historial académico de un estudiante, donde se definen parámetros, métricas y criterios de evaluación, así como niveles de desempeño.

4.2.6. Procesos disciplinarios

Es la gestión de los diferentes procesos que evalúan los índices comportamentales, disciplinarios y de convivencia, para lo que se definen diversos criterios que permiten evaluar a los estudiantes de manera individual.

4.2.7. Procesos administrativos

Es la gestión de los diferentes procesos de índole administrativa y directiva en una organización, puntualmente para el desarrollo del proyecto, estos procesos abarcan la gestión de los datos, la información, los usuarios, roles y permisos, así como la data financiera y presupuestal.

4.2.8. Plataforma

Sistema base que permite el funcionamiento de diferentes servicios, herramientas o módulos de software o hardware integrados o compatibles, las plataformas digitales han adquirido una gran importancia en la actualidad, gracias a las ventajas que supone integrar diferentes servicios y soluciones de software traspasando fronteras en pro de la globalización.

4.2.9. Módulo de calificaciones

Sistema modular con herramientas de software integradas que permiten generar, monitorear y gestionar los subprocesos educativos, necesarios para calificar las actividades académicas y brindar un repositorio de estas calificaciones para su visualización y procesamiento por los diferentes actores del proceso.

4.3. Marco Contextual

En la actualidad el uso de las TIC es, además de innovador, sumamente necesario para evolucionar y adaptarse al mercado en prácticamente cualquier sector económico, especialmente en el sector de la educación, donde los paradigmas se han modificado y la manera de enseñar y aprender ha evolucionado a través de los últimos años, esto a su vez permite explotar diferentes ventajas que pueden beneficiar a los estudiantes y a las instituciones educativas de básica y media.

Las instituciones educativas en Colombia, en su mayoría carecen de una solución que permita mantener toda su información, archivo y base de datos centralizada y disponible para todo el personal que pertenece a estas, fenómeno que se acentúa especialmente en las instituciones de índole oficial o estatal, sin embargo, hay unas pocas que hace uso de algún software o servicio en la nube que les permite tener esa información accesible para los interesados, pero estas soluciones son mayormente extranjeras, dificultando su comunicación y soporte, en comparación con proveedores nacionales.

Hay pocas empresas que han desarrollado soluciones de software que mitigan parcialmente muchas de estas necesidades, sin embargo, no son muy conocidas

nacionalmente, tienden a manejar costos elevados en comparación con las alternativas extranjeras, no tienen presencia en todo el territorio nacional, o tienen funciones muy limitadas, contrario a lo requerido por las instituciones educativas en este segmento.

Considerando los niveles de secundaria, básica y media, como los niveles de salida de la educación básica, previa a la educación superior y así mismo a los estudiantes de estos niveles, con un mayor grado de digitalización o con mayor conocimiento de las herramientas informáticas e internet, se selecciona a este grupo poblacional para la primera fase, donde se desarrollará un módulo de calificaciones para instituciones de educación básica secundaria y media.

5. DELIMITACIONES

La propuesta ha sido diseñada para ser implementada en diferentes instituciones educativas de índole oficial y privado, enfocado en los niveles de secundaria, básica y media, incluyendo únicamente el módulo de administración académica, donde principalmente se encuentran alojadas las herramientas para calificar, generación de informes académicos, gestionar la base de datos de estudiantes, docentes, directivos y acudientes, gestionar los cursos y las asignaturas, así como la creación de usuarios y asignación de permisos, se busca centralizar la información y garantizar su disponibilidad, accesibilidad e integridad.

5.1. Delimitación Tecnológica

Debido a que se trata de una solución de software, es necesario contar con algún dispositivo tecnológico para visualizar e interactuar con el programa y teniendo en cuenta que se busca tener disponibilidad y accesibilidad absoluta de la información en tiempo real, este software y sus datos, como la información de la institución se encuentran alojados en la red, por lo que también es necesario contar con conexión a internet para su correcto uso, sin embargo, para el desarrollo y especialmente teniendo en cuenta las comunidades de escasos recursos y zonas alejadas del territorio nacional, se tiene destinado desarrollar el aplicativo para ser usado en diferentes tipos de dispositivos como computadores, tablets o smartphones y se definen planes para facilitar la adquisición de dispositivos a bajo costo, por medio de alianzas, para estas poblaciones, de igual manera también se puede hacer uso de los equipos tecnológicos que las instituciones y el estado tiene a disposición a nivel local, municipal o regional y las bibliotecas públicas y privadas.

5.2. Delimitación Cognitiva

Se tiene claro que no todas las personas, especialmente las poblaciones de mayor edad y las poblaciones de índole rural, cuentan con un amplio conocimiento de las herramientas informáticas disponibles, por esto, se focaliza el desarrollo para los niveles de educación secundaria, ya que estos rangos de edad son más diestros en el uso de este tipo de herramientas, de igual manera se busca desarrollar una solución que sea altamente intuitiva y fácil de utilizar con una curva de aprendizaje muy baja, además, la propuesta se encuentra en concordancia con las campañas de digitalización que el gobierno ha venido haciendo en los últimos años y la

adquisición del software trae consigo una formación y capacitación para todos los actores que integran la comunidad.

5.3. Delimitación Temporal

Para el desarrollo del proyecto se establece un cronograma de actividades inicial, habiendo cumplido el cronograma hasta el cuarto mes, de acuerdo a lo planificado, el plan de mantenimiento y el plan de promoción y ventas se encuentra programado para ser iniciado el quinto mes:

Tabla 1: Cronograma inicial.

CRONOGRAMA DE ACTIVIDADES					
ACTIVIDAD	MES 1	MES 2	MES 3	MES 4	MES 5
Análisis y planeación					
Levantamiento de requerimientos					
Maquetación y bosquejos					
Diseño gráfico					
Codificación y desarrollo					
Pruebas y evaluación					
Mantenimiento y mejoras					
Promoción y ventas					

5.4. Delimitación Financiera

Para el desarrollo del proyecto se define un presupuesto inicial, el presupuesto se cumple de acuerdo a lo planificado a excepción del costo del Alojamiento en servidor web, debido a que este rubro se encuentra considerado en implementación en el servidor definitivo y se realiza en cuanto se inicie la comercialización del software.

Tabla 2: Presupuesto inicial.

RECURSO	DESCRIPCIÓN	PRESUPUESTO
Equipo Humano	01 desarrollador	\$6,000,000 (Total)
	01 diseñador grafico	\$2,400,000 (Total)
	01 director del proyecto	\$8,000,000 (Total)
	01 vendedor	\$1,600,000 (Total)
Equipos y Software	03 ordenadores (alta gama)	\$5,700,000 (Total)
	01 ordenador (gama media)	\$1,200,000 (Total)
Viajes y Salidas de Campo	Visitas a diferentes instituciones para el levantamiento de información.	\$600,000 (Total)
Materiales y suministros	Alojamiento en servidor web	\$1,100,000 (Anual)
TOTAL (COP)		\$ 26,600,000

5.5. Delimitación Legal

Con el auge de las aplicaciones informáticas, es necesario regular y controlar el desarrollo del software, lo que da a lugar diferentes discusiones en el plano nacional e internacional, en primer lugar, sobre la mejor manera para proteger jurídicamente el componente intelectual a nivel del desarrollo o creación de los aplicativos, para lo que se definen y regulan los derechos de autor y las patentes de invención, en segundo lugar, sobre la forma y las regulaciones necesarias para proteger jurídicamente la información y los datos contenidos y manipulados por medio de estas aplicaciones, para lo que se define la ley de protección de datos personales.

6. METODOLOGÍA

Una vez que se logran identificar las necesidades de las instituciones y se cotejan con los datos obtenidos por medio del levantamiento de información, se realiza un análisis que permite definir los requerimientos con que debe cumplir la solución de software.

Para este proyecto que busca desarrollar un módulo de calificaciones para las instituciones educativas de básica y media se determina usar un conjunto de metodologías que soporta la gestión de proyectos, donde se determinan principalmente; el alcance, para lo que se definen los objetivos, los recursos, donde se analiza y se define el personal, el capital, los bienes y equipos necesarios para este desarrollo, se establecen de igual manera el inicio, las diversas etapas de evaluación, donde se controla y verifica la evolución del proyecto para tomar medidas oportunas ante cualquier eventualidad o problema que surjan y la etapa final, donde se revisará que la solución de software cumpla de manera satisfactoria y se realice la entrega formal, para el paso a comercialización y posteriormente el proceso de mantenimiento y mejoras.

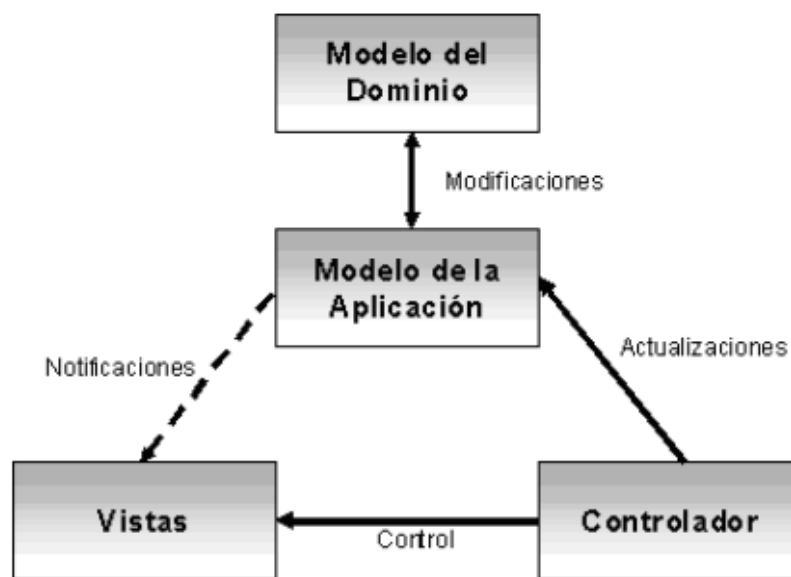
6.1. Modelo De Desarrollo En Cascada

Según Carrión et al. (2003) en el desarrollo de un software educativo se deben considerar dos elementos fundamentales la estructura y la teoría de aprendizaje, lo que permite generar una sinergia positiva entre las instituciones y la solución a desarrollar, teniendo en cuenta este principio, se decide optar por la metodología de desarrollo en cascada, lo que permite analizar las necesidades de las instituciones, para posteriormente levantar los requerimientos con que debe cumplir el software, tomando como base estos requisitos, se procede a diseñar el contenido, la interfaz y las bases de datos, ya teniendo esta información se procede a codificar la aplicación y antes de desplegar y dar acceso a los usuarios finales, se realiza el proceso de pruebas y evaluación, donde se busca corregir errores y buscar oportunidades de mejora, esto para generar procesos de mantenimiento y evaluación.

6.2. Arquitectura De Software Modelo Vista Controlador

Para complementar el modelo de desarrollo seleccionado, se establece el Modelo Vista Controlador (MVC), como modelo de arquitectura de software base, donde se separan los datos de la aplicación o el software, la interfaz de usuario y la lógica de control en tres componentes distintos, esto también permite mayor maniobrabilidad y flexibilidad para su desarrollo, además de tratarse de un modelo muy maduro y que ha demostrado su validez a lo largo de muchos años en todo tipo de aplicaciones, en una gran cantidad de lenguajes y plataformas de desarrollo, incluyendo por supuesto las plataformas web.

Figura 29. Relación entre los módulos del MVC. Ernesto Bascón. (2004). Recuperado de: El patrón de diseño Modelo-Vista-Controlador Universidad Católica Boliviana.

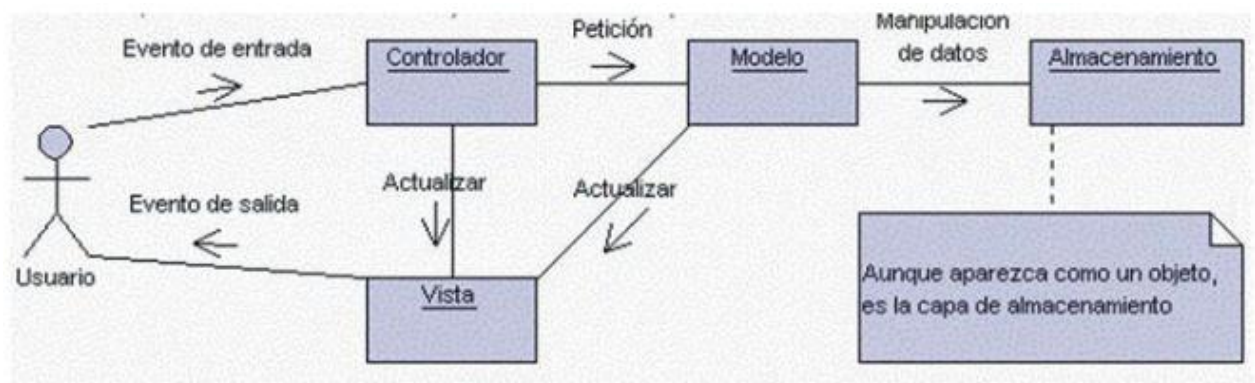


6.2.1. Patrones de diseño

Incorporar patrones de diseño en el proyecto, supone una gran optimización de los recursos, ya que estos patrones de diseño se conocen por ser soluciones generales, reutilizables y aplicables a diversas problemáticas de diseño de software. Generalmente, el uso de patrones de diseño evita volver constantemente a la etapa primaria de desarrollo, ya que se alimenta de errores previos de experiencias de otros desarrolladores, que ensamblaron una serie de plantillas

que ayudan a identificar condiciones de error y problemas en el código que pueden no ser evidentes en un momento dado, son soluciones que funcionan y que han sido usados y probados por una gran cantidad de desarrolladores, haciendo que este tipo de proyectos sean menos propensos a errores, actualmente se conocen hasta 23 patrones de diseño, clasificados en tres categorías principales: Creacional, Estructural y de Comportamiento.

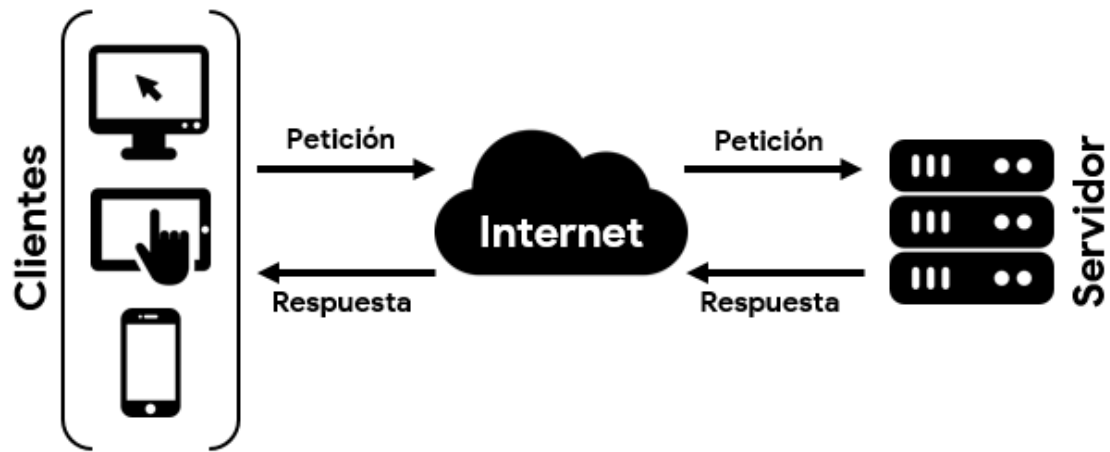
Figura 32. Interrelación entre los elementos del patrón MCV. Yanette Díaz, Yenisleidy Fernández. (2012). Patrón Modelo-Vista-Controlador. Recuperado de: Revista Digital De Tecnologías De La Información Universidad Tecnológica De La Habana.



6.3. Tecnologías Web

Se desarrolla y codifica la solución de software haciendo uso de tecnologías web, teniendo en cuenta los últimos estándares y versiones de estas tecnologías para garantizar la seguridad e integridad de la información; HTML5, CSS3, JavaScript, SQL, AJAX, JSON, PHP7, integrando bases de datos relacionales y alojado en la nube en un servidor web, se decide decantarse por el uso de estas tecnologías debido a su amplio uso, su nivel de madurez y con el propósito de maximizar la compatibilidad y el acceso, ya que gracias a que se encuentra alojada en un servidor, solo se necesita de un dispositivo conectado a internet y un navegador, adicionalmente facilitan su mantenimiento, actualización y escalabilidad.

Figura 30. Modelo cliente-servidor. Elaboración propia.



6.4. Seguridad informática

En la actualidad el mayor activo de las organizaciones es la información, los datos personales y corporativos que cobran aún mayor importancia en torno a las soluciones de software que gestionan y administran dicha información, y proporcional a su importancia es el riesgo, ya que los datos se encuentran constantemente amenazados y hay una creciente y permanente cantidad de ataques, que buscan aprovechar hasta la más mínima vulnerabilidad, buscando diferentes propósitos, desde robo de información, hasta la corrupción de los datos con la intención de destruir o afectar negativamente la imagen de una organización.

Uno de los ataques informáticos más ampliamente conocido es la inyección SQL, infiltración de código intruso, haciendo uso de alguna vulnerabilidad, realizando así operaciones en las bases de datos, a fin de alterar el funcionamiento normal del programa, dependiendo de la gravedad de la vulnerabilidad y de la complejidad del ataque, la ejecución del código puede comprometer la información o puede permitir al atacante obtener el control completo de la base de datos con los riesgos que esto supone.

Para impedir o prevenir este tipo de ataques, el software toma los registros ingresados por el usuario y por medio de diferentes métodos se formatearán las entradas, esto para evitar que un usuario pueda ingresar fragmentos de código

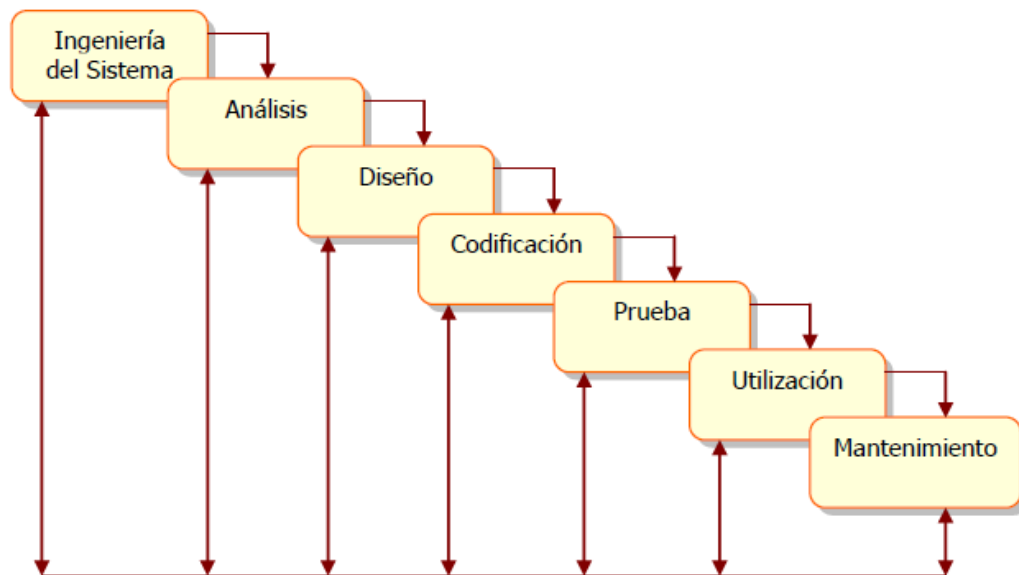
que puedan suponer un riesgo, adicionalmente cada campo de registro de datos tendrá un formato predefinido y las consultas u operaciones de la base de datos se harán por medio de sentencias preparadas y consultas parametrizadas.

7. DISEÑO Y DESARROLLO DEL SOFTWARE

7.1. Modelo De Desarrollo

Fundamentar este desarrollo en una metodología de gestión de proyectos, maximiza la oportunidad de conseguir los objetivos preestablecidos para proporcionar un beneficio a la institución y minimizar los riesgos. Para esto se deben definir tres aspectos claves; el alcance para determinar las fases del desarrollo del proyecto, el tiempo para establecer los periodos de desarrollo para cada fase y el costo para identificar y delimitar los topes mínimos y máximos de la inversión económica.

Figura 8. Modelo en cascada. Aparicio Pilar. (2012). Recuperado de: Modulo de ingeniería de software UNAD.



7.2. Análisis

Para culminar el proyecto de manera óptima fue fundamental abonar esfuerzos en esta etapa primaria, donde inicialmente se definen las herramientas para el levantamiento de información, y se incluye el estudio de viabilidad, donde se evalúa el costo, la rentabilidad y la factibilidad del proyecto de software, por medio

de estos datos es posible generar un pliego de condiciones, un plan, una estimación financiera y una propuesta comercial para el cliente final.

Para identificar el valor de la inversión final fue necesario definir clara y detalladamente los requisitos o requerimientos, esto a su vez permite identificar la complejidad y envergadura del proyecto:

7.2.1. Identificación de actores

Se identifican los actores, aquellas personas que se requieren para desarrollar el proyecto desde su planeación hasta la puesta en marcha y comercialización, así mismo, se describen las funciones principales.

Tabla 3. Identificación de actores.

ACTOR	FUNCION
Director del proyecto	▪ Levantamiento de información. ▪ Consolidación y análisis de información. ▪ Realizar estudio de viabilidad. ▪ Planificación del proyecto. ▪ Definición de presupuesto. ▪ Definición de requisitos. ▪ Seguimiento y control en el desarrollo. ▪ Definición de estrategias de mercadeo.
Diseñador gráfico	▪ Creación de bocetos o prototipos. ▪ Definición de fuentes tipográficas. ▪ Definición de paletas de colores corporativos. ▪ Creación de interfaz gráfica. ▪ Creación de iconos y plantillas de diseño.
Desarrollador	▪ Definir arquitectura de software. ▪ Codificación del software. ▪ Creación e implementación de la base de datos. ▪ Crear documentación del software. ▪ Definir plan de pruebas y evaluación. ▪ Corrección de errores. ▪ Implementación y despliegue del software.
Vendedor	▪ Definir plan de mercadeo. ▪ Presentación de catálogo y demo. ▪ Definición de planes de atención, venta y

	servicio post venta.
Administrador	▪ Creación, modificación y eliminación de usuarios. ▪ Gestión de credenciales de acceso. ▪ Asignación de roles y permisos. ▪ Gestión del módulo. ▪ Mantenimiento base de datos.
Usuario final	▪ Uso de las herramientas incorporadas en el software de acuerdo con el rol designado. ▪ Responsabilizarse del uso de las credenciales de acceso.

7.2.2. Identificación de roles y permisos

Se identifican los roles y se definen los permisos requeridos para cada tipo de usuario, de acuerdo con el rol que desempeña cada uno en la institución.

Tabla 4. Roles y permisos para el usuario final.

ROL	PERMISOS Y ACCESOS
Administrador	▪ Posee permisos absolutos en el software, puede modificar toda la información de la base de datos, crear, modificar, eliminar y asignar permisos, roles y credenciales de acceso.
Directivo	▪ Posee permisos limitados, puede crear usuarios y credenciales de acceso para docentes, estudiantes y acudientes, puede matricular a los docentes y estudiantes y crear y asignar asignaturas a los cursos.
Docente	▪ Posee permisos limitados, puede visualizar los listados de estudiantes por curso asignado y registrar las calificaciones de las asignaturas que le corresponden.
Estudiante	▪ Posee permisos limitados, puede visualizar sus registros de calificaciones e imprimir sus informes académicos.
Acudiente	▪ Posee permisos limitados, puede visualizar los registros de calificaciones e imprimir los informes académicos de su hijo o estudiante acudido y puede firmar el recibido o informado de este registro académico.

7.2.3. Identificación de requerimientos

Se identifican los requerimientos funcionales y no funcionales con los que debe cumplir el software para garantizar que satisface las necesidades propias del proceso.

Tabla 5. Requerimientos funcionales y no funcionales.

REQ. FUNCIONALES	REQ. NO FUNCIONALES
▪ Cuenta con interfaz gráfica de usuario. ▪ Cuenta con un menú de opciones que permite al usuario navegar entre las herramientas. ▪ Debe permitir al acudiente firmar o aceptar el informado del informe de calificaciones. ▪ Muestra las calificaciones registradas en cada periodo académico y en cada asignatura. ▪ Permite la descarga e impresión de los informes académicos individuales. ▪ Tiene herramientas de gestión y administración de credenciales. ▪ Permite editar la información personal de cada usuario. ▪ Permite tener un histórico de notas. ▪ Debe permitir definir la cantidad de horas de clase semana por área. ▪ Permite definir a un docente como director de curso para cada grado. ▪ Permite a los acudientes ver la	▪ Debe mostrar y especificar errores de sintaxis, errores del aplicativo o errores del servidor. ▪ Genera un log o fichero de registro donde se listan los errores, fallas o indecencias con el funcionamiento. ▪ Funciona desde un servidor web con una disponibilidad del 99%. ▪ Cuenta con credenciales únicas de acceso para garantizar la seguridad. ▪ Las interacciones realizadas con cada usuario quedan registradas en el sistema. ▪ Permite acceder a la información desde cualquier ubicación. ▪ Debe realizar de manera automática el cálculo del promedio acumulado. ▪ Guarda la información del periodo actual y de periodos anteriores. ▪ La base de datos se encuentra protegida. ▪ Se puede hacer un backup de la base de datos de acuerdo a las necesidades.

información del estudiante del cual es responsable. ▪ Permite al administrador asignar roles y permisos de acuerdo con su papel en la institución.	▪ Cada usuario puede ver información y secciones del aplicativo de acuerdo a su rol y únicamente la información autorizada y definida previamente.
---	--

7.3. Diseño

Por medio de esta fase se formula la solución de software específica, teniendo en cuenta las exigencias y necesidades, las tareas y las estrategias definidas en la fase previa, para esto el programador se encarga de diseñar la arquitectura, modela las bases de datos y el plan de diseño general, detallando los componentes concretos, las interfaces, los entornos de trabajo, entre otros componentes del software y los planes de pruebas.

7.3.1. Modelo relacional de la base de datos

Figura 1. Modelo relacional DB. 2021. Elaboración propia.

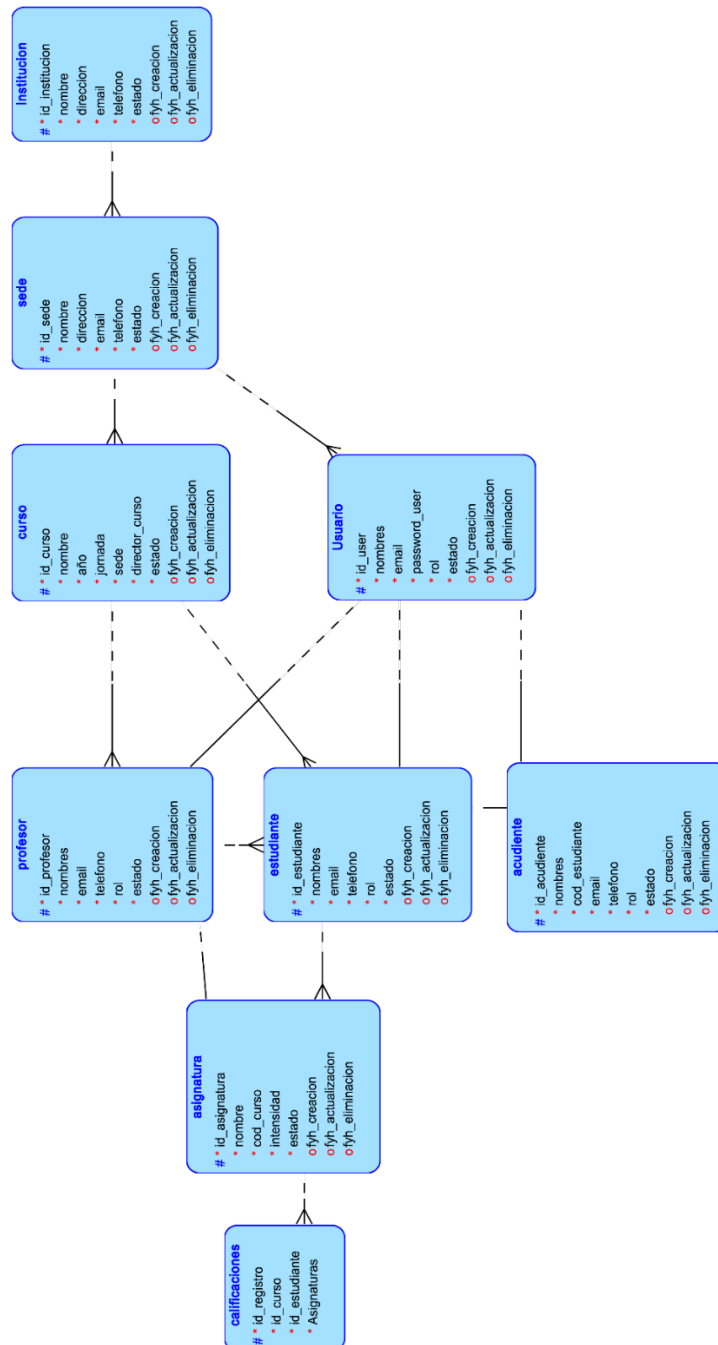
Ver en: [Ver imagen](#) o [Ver pdf](#)



7.3.2. Modelo lógico de la base de datos

Figura 2. Modelo lógico DB. 2021. Elaboración propia.

Ver en: [Ver imagen](#) o [Ver pdf](#)



7.3.3. Diagramas de casos de uso

Figura 3. Caso de uso: Autenticación. Elaboración propia.

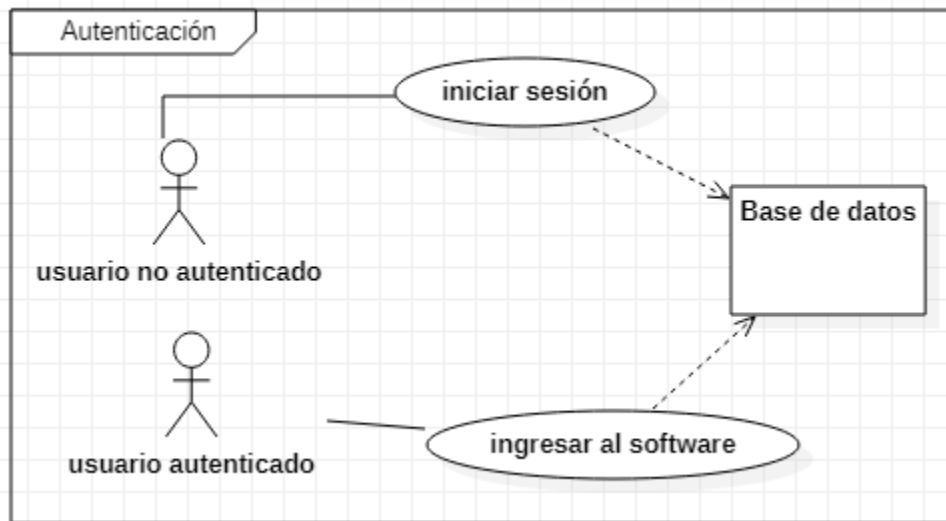


Figura 4. Caso de uso: Gestión de usuarios. Elaboración propia.

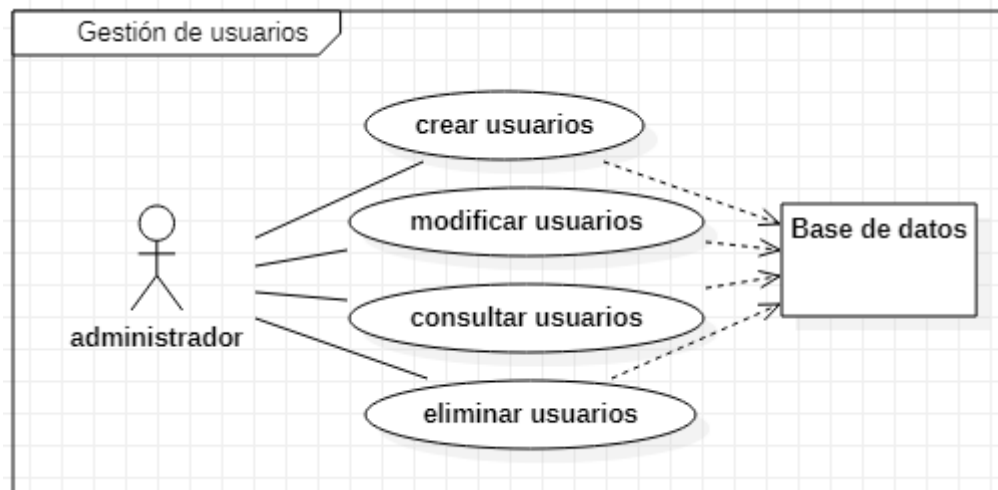


Figura 5. Caso de uso: Calificación. Elaboración propia.

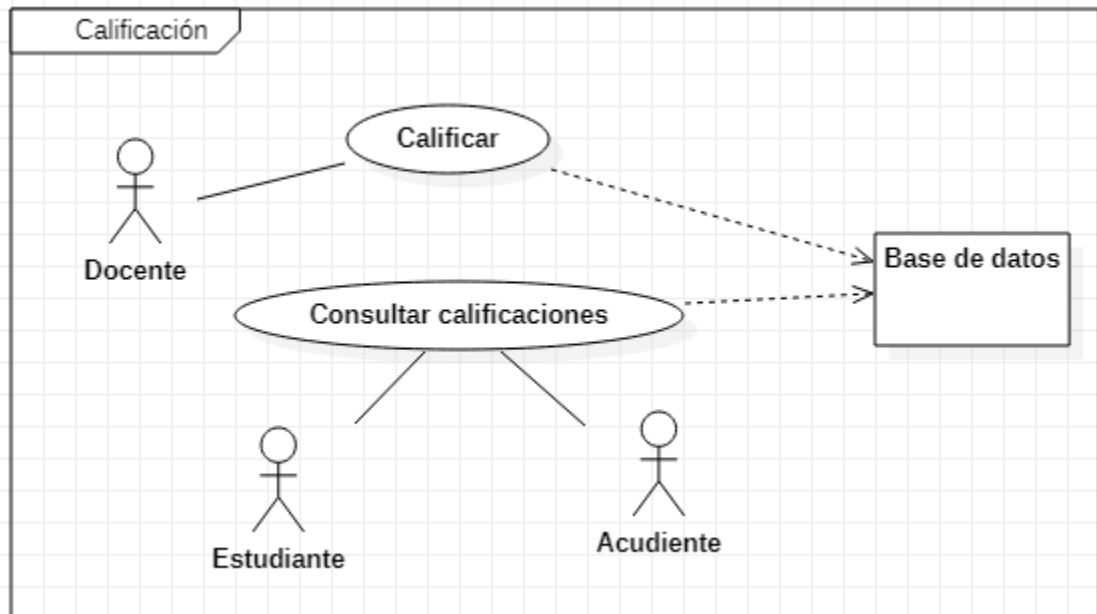
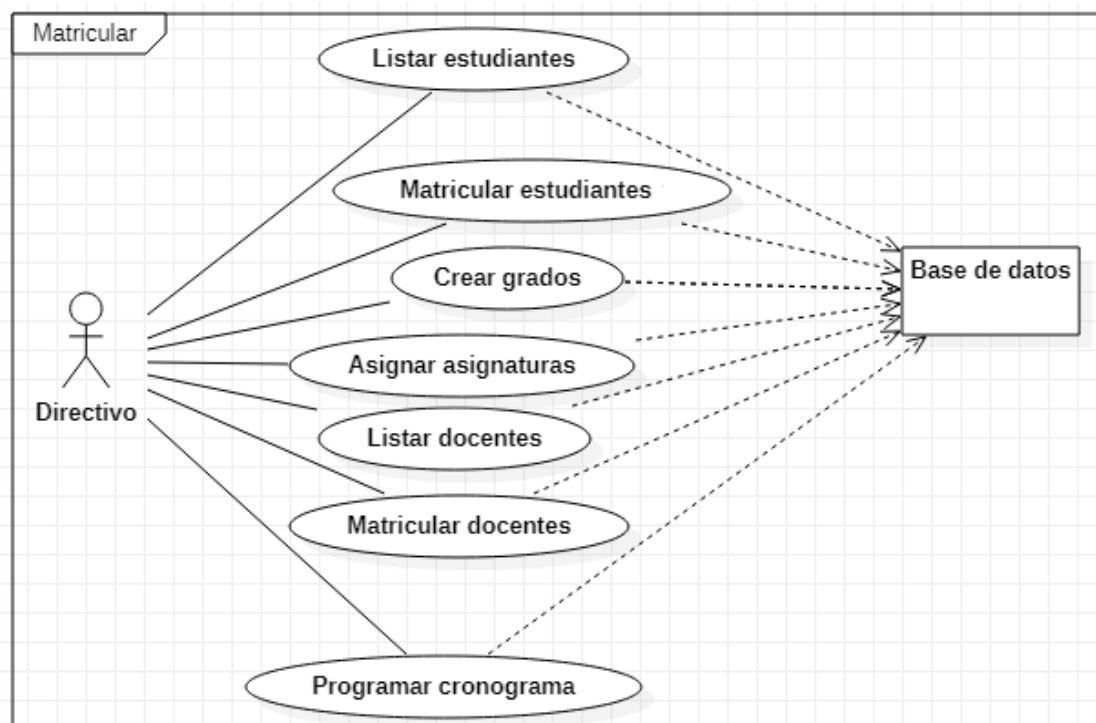


Figura 6. Caso de uso: Calificar. Elaboración propia.



Figura 7. Caso de uso: Matricular. Elaboración propia.



7.3.4. Diagramas de actividades

Figura 24. Diagrama de actividad: Autenticación. Elaboración propia.

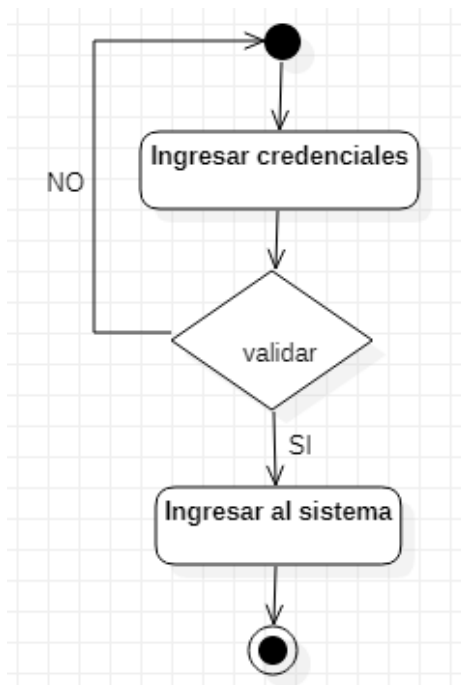


Figura 25. Diagrama de actividad: Gestión de usuarios. Elaboración propia.

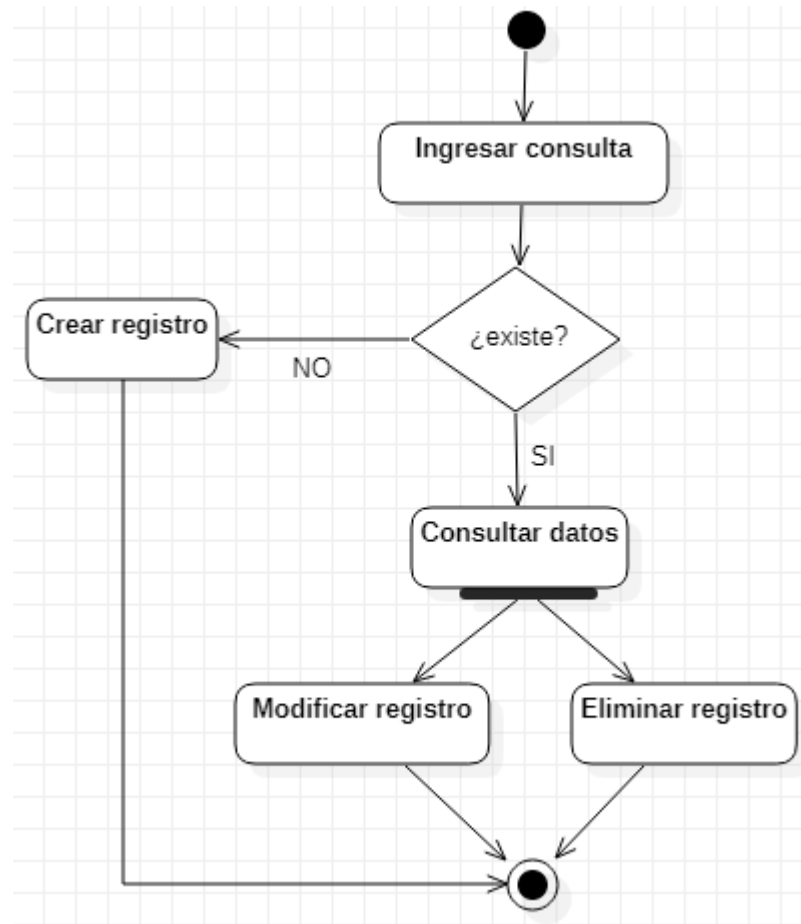


Figura 26. Diagrama de actividad: Calificación. Elaboración propia.

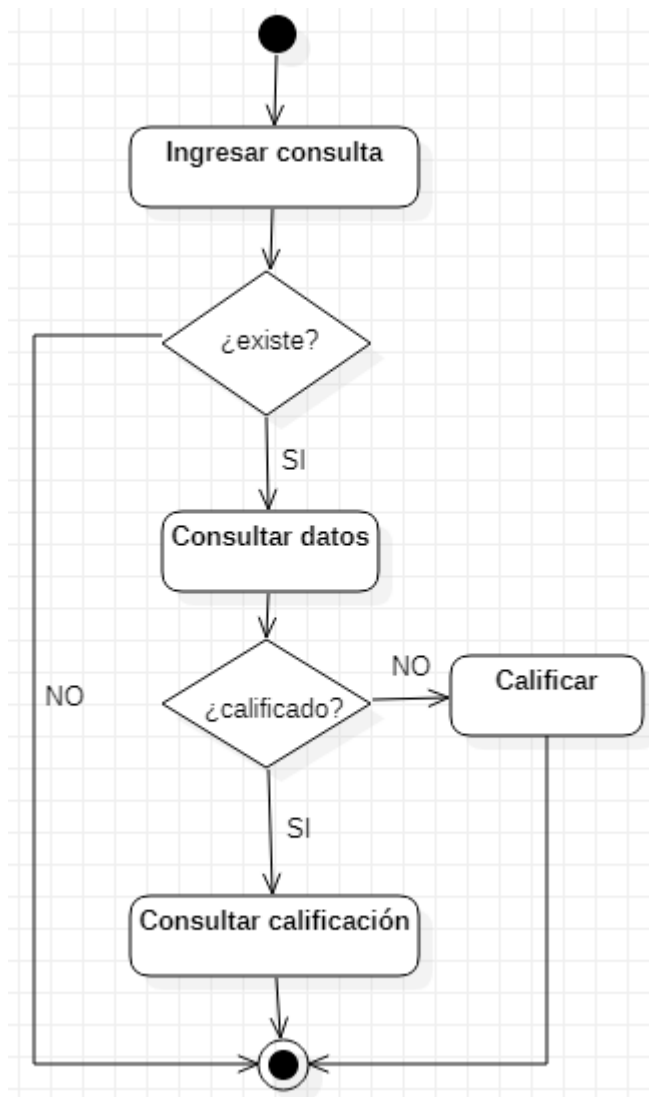
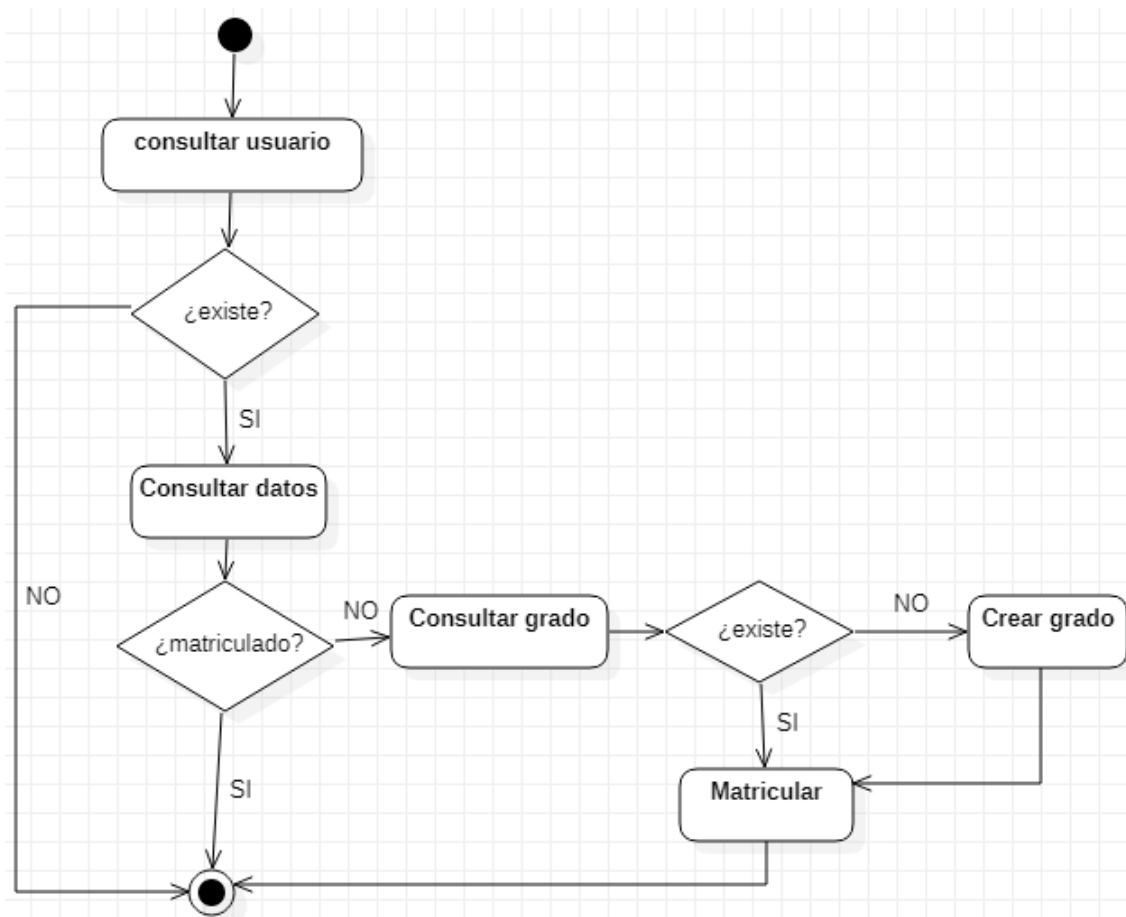
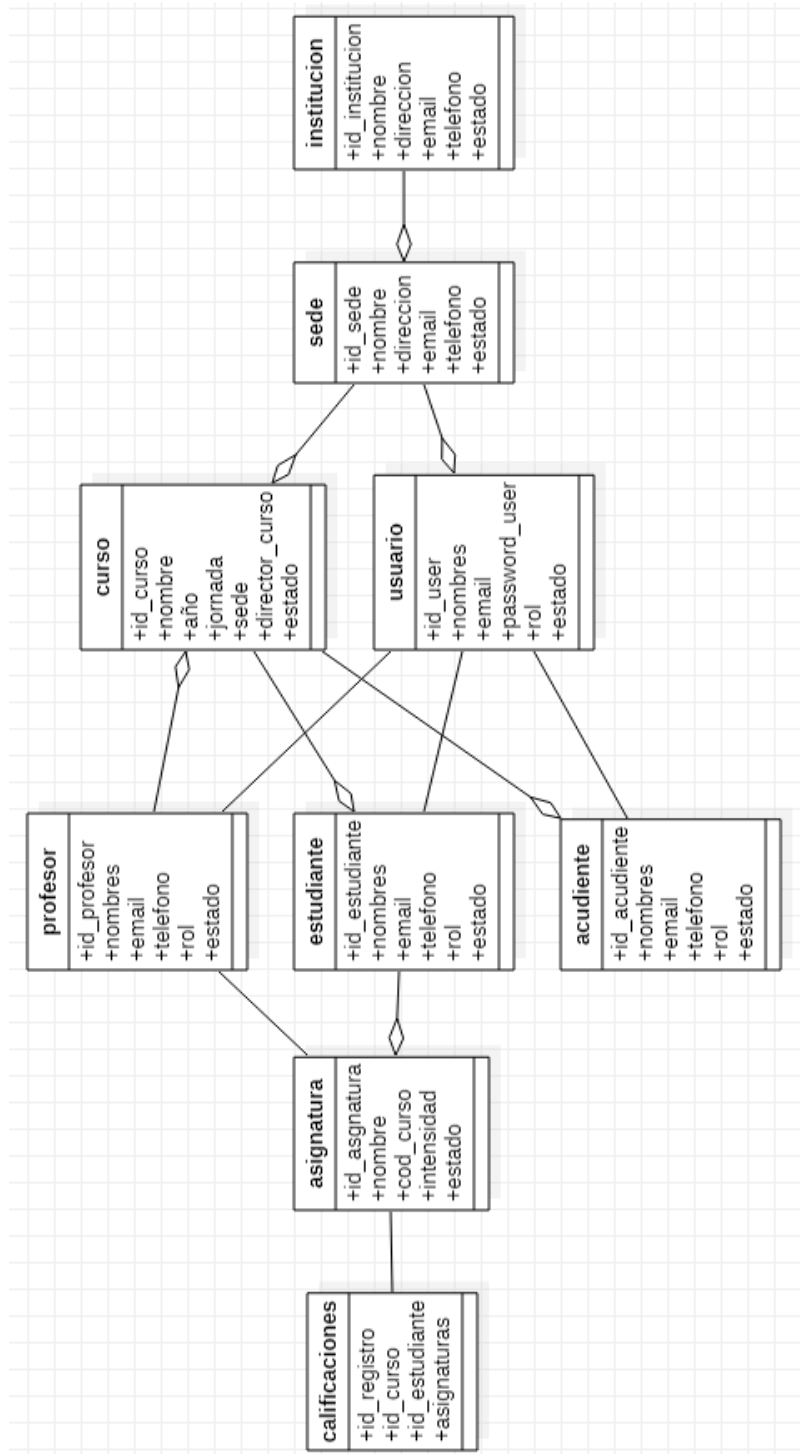


Figura 27. Diagrama de actividad: Matricular. Elaboración propia.



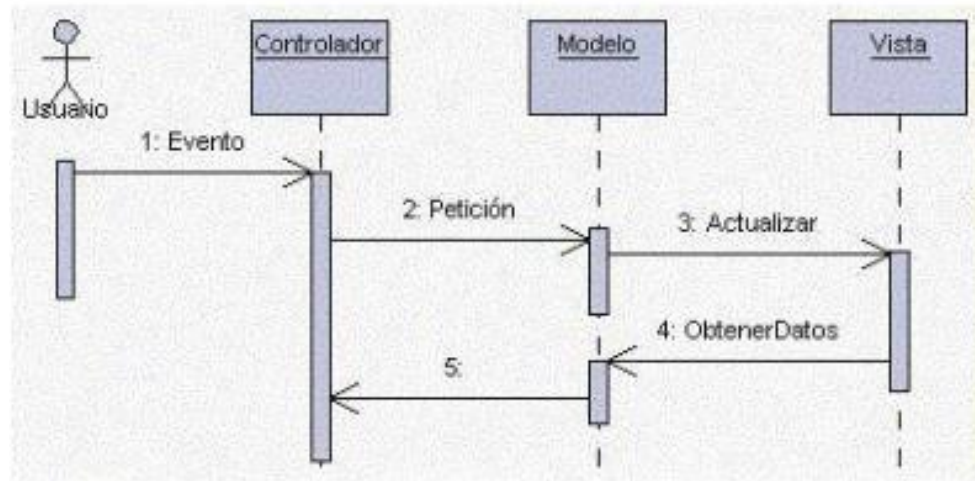
7.3.5. Diagrama de clases

Figura 28. Diagrama de clases. Elaboración propia. Ver en: [Ver imagen](#) o [Ver pdf](#)



7.3.6. Diagrama de secuencia

Figura 31. Diagrama de secuencia. Yanette Díaz, Yenisleidy Fernández. (2012). Patrón Modelo-Vista-Controlador. Recuperado de: Revista Digital De Tecnologías De La Información Universidad Tecnológica De La Habana.



7.4. Implementación

En esta fase se realiza la codificación y el desarrollo del software, se toma la información obtenida en las etapas de análisis y diseño, y esta se traduce al lenguaje de programación definido, el proyecto se divide en diversos componentes que a su vez se desarrollan por separado y se integran paulatinamente, hasta tener un producto final que se procede a probar y evaluar, para garantizar que cumpla con los requerimientos acordados.

Para realizar la implementación, inicialmente se diseña un borrador o bosquejo de la interfaz y se define la interacción del usuario con ésta, para usar cada una de las herramientas definidas por el rol de cada usuario.

7.4.1. Diseño de interfaz de usuario

Una fase fundamental para el desarrollo e implementación del software es la planificación del diseño de la interfaz de usuario:

Módulo de inicio de sesión donde se ingresan las credenciales de acceso y se validan con las credenciales que se encuentran definidas en la base de datos conectada con el software.

Figura 9. Wireframe: Inicio de sesión. Elaboración propia.

Aplicativo para la gestión educativa





☐ Recordar credenciales

INGRESAR

Se muestran las herramientas de administración principales, a las que pueden acceder los usuarios con mayores privilegios, como los directivos o el usuario administrador.

Figura 10. Wireframe: Menú de herramientas. Elaboración propia.

Herramientas administrativas

AÑADIR PROFESOR 	LISTAR PROFESORES 	AÑADIR CURSO 
AÑADIR ESTUDIANTE 	LISTAR ESTUDIANTES 	AÑADIR ASIGNATURA 
AÑADIR ACUDIENTE 	LISTAR ACUDIENTES 	_BLANK 

Se muestra el borrador del esquema del informe académico o informe de calificaciones que pueden visualizar e imprimir los estudiantes y acudientes.

Figura 11. Wireframe: Informe de calificaciones. Elaboración propia.

Nombre del colegio

Sede: *** | Jornada: ***

Estudiante

estudiante de prueba

Curso:

Código:

Año lectivo:

Asignatura	IHS	I	II	III	IV	R	Acum	Desempeño
string	int	double	double	double	double	double	double	string
string	int	double	double	double	double	double	double	string
string	int	double	double	double	double	double	double	string
string	int	double	double	double	double	double	double	string
string	int	double	double	double	double	double	double	string

FIRMA Y NOMBRE

FIRMA Y NOMBRE

Firma: director de curso

Firma: acudiente

Se muestra un resumen grafico con barras de progreso, permitiendo que los estudiantes y acudientes comprendan fácilmente el avance y progreso en cada una de las asignaturas matriculadas.

Figura 12. Wireframe: Barras de progreso. Elaboración propia.

INICIO

ACTUALIDAD

MENSAJES

REPORTES

prueba

MATEMATICAS

CIENCIAS NATURALES

ASIGNATURA

ACADÉMICO

Se muestra la pantalla donde se pueden modificar los datos de acceso del usuario.

Figura 13. Wireframe: Edición de datos de usuario. Elaboración propia.

Editar usuario

Nombre completo

directivo de prueba

Email

probando@prueba.test

Contraseña

.....

Rol

Directivo



ACTUALIZAR

Se muestra una lista de usuarios, los listados son clasificados por los diferentes roles, donde se muestran los datos básicos de cada usuario y permite modificar y eliminar los registros por quienes cuentan con privilegios superiores.

Figura 14. Wireframe: Listado de usuarios. Elaboración propia.

Mostrar

10

Registros

Buscador de estudiantes

ión	No.	Código	Nombre	Email
	1	10002	docente de prueba	probando@prueba.tes

Mostrando 1 a 1 de 1 Registros

Anterior

1

Siguien

Muestra una pantalla emergente donde se pueden ingresar datos.

Figura 15. Wireframe: Registro de datos. Elaboración propia.

ir curso

O

SO

CERRAR

LISTAR PROFESSORES

Muestra la confirmación de modificación o eliminación de registros.

Figura 16. Wireframe: Eliminación de registros. Elaboración propia.

The wireframe shows a mobile application interface for 'lemyApp'. At the top is a navigation bar with icons and labels: 'lemyApp', 'INICIO', 'ACTUALIDAD', 'MENSAJES', 'REPORTES', and 'CERRAR'. Below the navigation bar, the text '@, directivo de prueba' is displayed. The main content area features a confirmation dialog titled '¿Desea eliminar el acudiente?'. This dialog contains two input fields: the first is labeled 'acudiente de prueba' and the second contains the value '10004'. At the bottom of the dialog are two buttons: 'ELIMINAR' and 'CANCELAR'.

lemyApp INICIO ACTUALIDAD MENSAJES REPORTES CERRAR

@, directivo de prueba

¿Desea eliminar el acudiente?

acudiente de prueba

10004

ELIMINAR CANCELAR

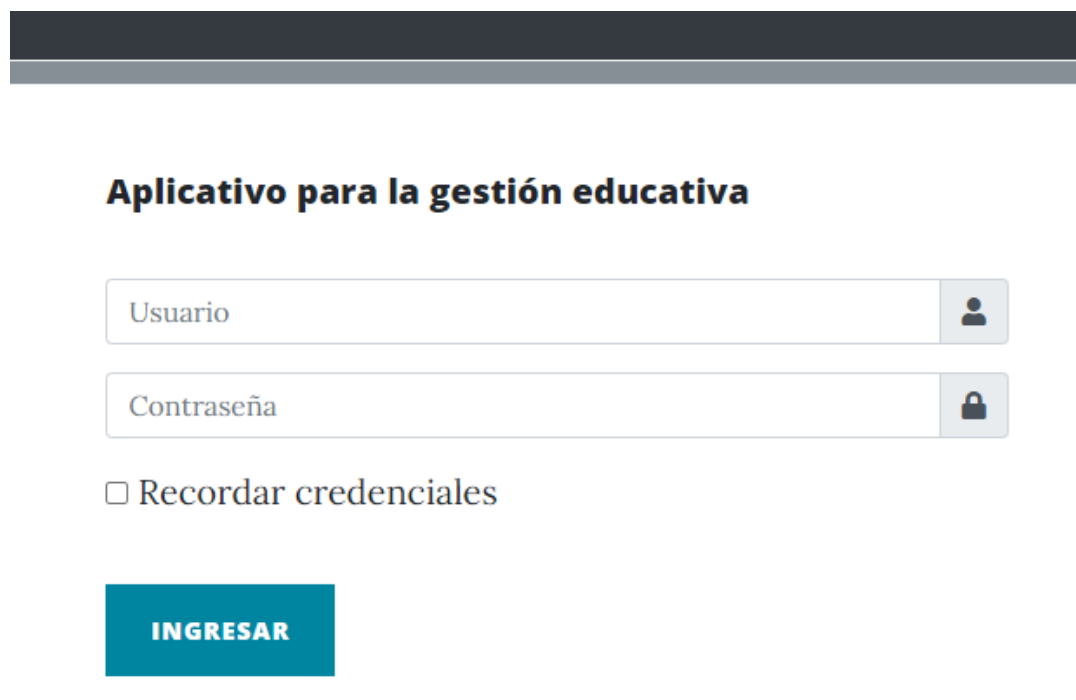
7.5. Prueba

En esta etapa el producto de software se integra con un entorno de pruebas con características similares a las que tendría el entorno en que va a funcionar y se seleccionan algunos usuarios finales, para que utilicen el software de la manera en que requieren que funcione, evaluando que cumpla con las exigencias ya definidas, además de los estándares de calidad y funcionamiento mínimos requeridos para tener una experiencia óptima.

7.5.1. Software versión beta

Proceso de autenticación, vista de ingreso de credenciales de acceso.

Figura 17. Inicio de sesión. Elaboración propia.



The image shows a login interface for an educational management application. At the top, there is a dark header bar. Below it, the title "Aplicativo para la gestión educativa" is displayed in a bold, dark font. The form consists of two input fields: "Usuario" (User) and "Contraseña" (Password). The "Usuario" field has a user icon on the right, and the "Contraseña" field has a lock icon on the right. Below these fields is a checkbox labeled "Recordar credenciales" (Remember credentials). At the bottom of the form is a teal button labeled "INGRESAR" (Log In).

Aplicativo para la gestión educativa

Usuario

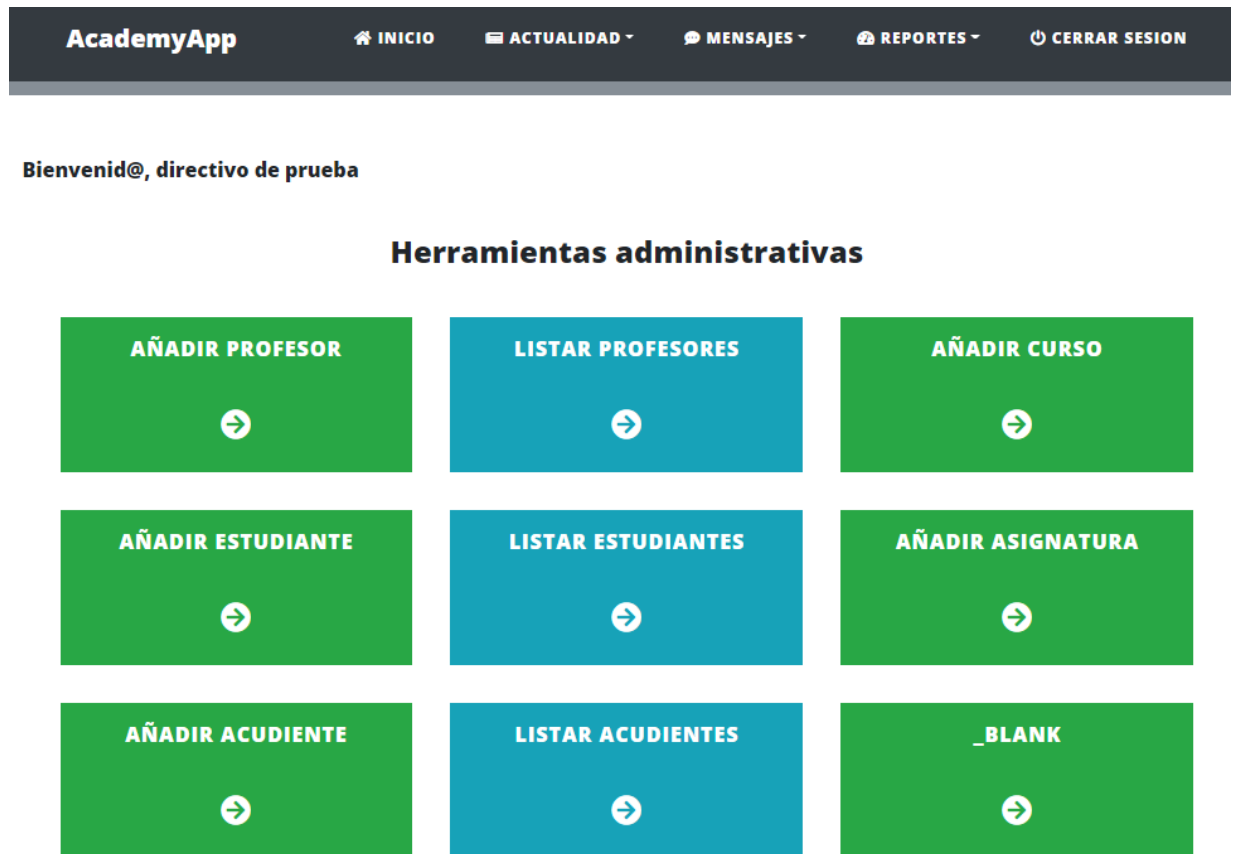
Contraseña

☐ Recordar credenciales

INGRESAR

Proceso de consulta de herramientas administrativas, vista del dashboard.

Figura 18. Menú de herramientas. Elaboración propia.



Proceso de consulta de registros de calificaciones, vista de barras de progreso.

Figura 19. Barras de progreso. Elaboración propia.



Proceso de modificación de datos en registro de usuario autenticado.

Figura 20. Edición de datos de usuario. Elaboración propia.

The screenshot shows the AcademyApp interface for editing a user profile. The navigation bar is the same as in Figure 19. The user is greeted as 'Bienvenid@, admin de prueba'. The main content area is titled 'Editar usuario'. It contains four input fields: 'Nombre completo' (with the value 'directivo de prueba'), 'Email' (with the value 'probando@prueba.test'), 'Contraseña' (with masked characters '*****'), and 'Rol' (a dropdown menu with 'Directivo' selected). Below these fields is a teal button labeled 'ACTUALIZAR'.

Proceso de consulta y listado de registros de usuarios.

Figura 21. Listado de usuarios. Elaboración propia.

AcademyAppINICIOACTUALIDADMENSAJESREPORTESCERRAR SESION

Bienvenid@, directivo de prueba

Listado de profesores

Mostrar 10 Registros

Buscador de estudiantes

Acción	No.	Código	Nombre	Email
	1	10002	docente de prueba	probando@prueba.test

Mostrando 1 a 1 de 1 Registros

Anterior 1 Siguiente

Proceso de creación de registros de cursos en la base de datos.

Figura 22. Registro de datos. Elaboración propia.

AcademyAppINICIOACTUALIDADMENSAJESREPORTESCERRAR SESION

Bienvenid@, directivo de prueba

Añadir curso

Curso

Curso

CERRAR

Proceso de eliminación de usuarios, eliminación de usuario con perfil de acudiente.

Figura 23. Eliminación de registros. Elaboración propia.

The screenshot shows the AcademyApp interface. At the top is a dark navigation bar with the app name and several menu items: INICIO, ACTUALIDAD, MENSAJES, REPORTES, and CERRAR SESION. Below the navigation bar, the user is greeted as 'Bienvenid@, directivo de prueba'. The main content area displays a confirmation dialog titled '¿Desea eliminar el acudiente?'. This dialog contains two input fields: the first field contains the text 'acudiente de prueba' and the second field contains the number '10004'. At the bottom of the dialog are two buttons: a red button labeled 'ELIMINAR' and a blue button labeled 'CANCELAR'. The footer of the page indicates 'Copyright © UNAD'.

7.6. Servicio

Finalmente, el producto de software finalizado es aprobado y continúa en etapa productiva para ser utilizado por los usuarios finales para lo que fue desarrollado, es decir, el software se entrega para su uso, generando planes de mantenimiento, evaluación y mejora, con el propósito de corregir erros menores que hayan podido quedar posteriormente a la etapa de prueba e implementar posibles mejoras de acuerdo con la dinámica del mercado y las exigencias de los clientes.

7.6.1. Matriz de evaluación

Se define una serie de elementos que permiten estructurar un plan de evaluación y ayudan a definir los lineamientos de la estrategia de mejoramiento continuo.

Tabla 6. Matriz de evaluación del software.

DETERMINANTE	DESCRIPCIÓN
Funcionalidad	Se evalúa la manera en que el producto de software gestiona el conjunto de funciones con las que cuenta y que estas cumplan con las necesidades y requerimientos para los que fue diseñado y desarrollado.
Confiabilidad	Se evalúa la capacidad que tiene el software, para mantener su funcionamiento y ejecución en condiciones óptimas o normales dentro de periodos de tiempo establecidos.
Usabilidad	Se evalúa el esfuerzo que deben realizar los usuarios, independientemente de los roles o permisos asignados para utilizar las funciones que componen al producto de software, así como la curva de aprendizaje necesaria para usarlo.
Eficiencia	Se evalúa la relación que hay entre el funcionamiento del producto de software con la cantidad de recursos lógicos y físicos usados para su ejecución con respecto al tiempo.
Mantenibilidad	Se evalúa el esfuerzo que se necesita invertir para realizar cualquier modificación al producto de software, ya sea para corregir errores, realizar parches de seguridad o para incrementar su funcionalidad.
Portabilidad	Se evalúa la capacidad que tiene el producto de software para ser transferido de un ambiente o entorno a otro, sin que se vea deteriorado su funcionamiento, sus componentes, ni su usabilidad.

7.6.2. Métricas de evaluación

Se define una serie de métricas que permiten evaluar las características fundamentales del software.

Tabla 7. Métricas de evaluación.

CARACTERÍSTICAS	SUBCARACTERÍSTICAS	MÉTRICAS
Funcionalidad	Ejecución	Cuenta con facilidad para la instalación y puesta en marcha.
Funcionalidad	Sincronización	El funcionamiento es en tiempo real y sincronizado

		con la DB
Confiabilidad	Seguridad	Los protocolos y estándares de seguridad y autenticación son de alto nivel.
Confiabilidad	Disponibilidad	Alta tolerancia a fallos y facilidad de restauración.
Usabilidad	Intuitivo	Cuenta con una interfaz de usuario intuitiva, amigable y fácil de utilizar
Usabilidad	Aprendizaje	La curva de aprendizaje es baja.
Eficiencia	Recursos	Cada proceso del aplicativo debe hacer uso dinámico y optimizado de los recursos necesarios.
Eficiencia	Respuesta	Debe hacer uso solo de los recursos que necesita cada proceso, así los índices de eficiencia son altos.
Mantenibilidad	Preventivo	Permite la ejecución de rutinas de mantenimiento preventivo periódicas definidas.
Mantenibilidad	Correctivo	Facilita la corrección de incidencias o fallas.
Portabilidad	Homologación	Permite escalar y ser modificado para complementarse o reemplazar su código con nuevas instancias y tecnologías compatibles.
Portabilidad	Multiplataforma	Al tratarse de un aplicativo web, su ejecución debe garantizar en cualquier sistema operativo y navegador web.

7.6.3. Plan de evaluación

Se define un plan de evaluación, con una serie de pruebas de índole funcional y no funcional para garantizar que el software funcione correctamente y cumpla con los requerimientos del cliente, de acuerdo con sus necesidades.

Tabla 8. Pruebas funcionales.

PRUEBAS FUNCIONALES	RESULTADOS
Prueba de seguridad y acceso, para impedir cualquier acceso sin el uso de las credenciales asignadas.	En cuanto se intenta acceder directamente a las secciones de gestión, el sistema muestra una advertencia y redirige al usuario a sección de autenticación.
Prueba de funcionamiento de todas las herramientas incorporadas en el software.	Se simula cada uno de los procesos dentro del sistema, encontrando funcionamiento óptimo.
Prueba de estabilidad y disponibilidad del software y la información.	Se genera gran cantidad de solicitudes de carga, donde se evidencia caída del servicio después de 500 solicitudes simultáneas, sin embargo, el software se encuentra instalado en un servidor de pruebas con especificaciones básicas.
Prueba de reporte de errores, para identificar cuando se presente una incidencia.	El sistema genera un registro de errores, para consulta y análisis.
Prueba de integridad y seguridad en los datos y la información de la base de datos y a nivel institucional.	Se intenta extraer la base de datos sin acceder al módulo de administración sin éxito.
Prueba de integración, para verificar que los componentes se encuentren integrados y en óptimo funcionamiento.	Se intenta integrar con una página web de prueba, sin encontrar novedades.

Tabla 9. Pruebas no funcionales.

PRUEBAS NO FUNCIONALES	RESULTADOS
Prueba de carga, donde se evalúa el tiempo que tarda el software en cargar la interfaz, la base de datos y las herramientas en diferentes condiciones.	La carga de los recursos en servidor local es menor a 3 segundos, la carga en servidor de pruebas es ligeramente superior.
Prueba de estrés, donde se evalúa el funcionamiento del software en las condiciones de mayor exigencia.	Se realiza prueba de estrés, encontrando lentitud en la carga de los recursos a partir de 200 solicitudes y caída intermitente a partir de 500 solicitudes.
Prueba de escalabilidad, donde se evalúa la viabilidad de implementar y desplegar el software en escenarios de carga creciente de datos y trabajo.	La carga de recursos mejora con el incremento del hardware y el módulo se desarrolla con el propósito de ser modular, por lo que es escalable.

Prueba de portabilidad, donde se evalúa el comportamiento del software cuando se mueve a diferentes ambientes.	Se realiza portabilidad del ambiente del servidor local al ambiente del servidor de pruebas.
Prueba de configuración, para verificar que todos los parámetros estén configurados correctamente.	Se establecen diferentes configuraciones para verificar el comportamiento del software con alta tolerancia a fallos.

8. CONCLUSIONES

El desarrollo del presente proyecto y, por consiguiente, de este software, ha permitido aplicar diferentes tecnologías y conocimientos de Ingeniería de Sistemas, para aportar al contexto social, económico, ambiental y cultural, identificando y solucionando problemáticas, y soportando diferentes tipos de organizaciones.

Para desarrollar el proyecto de software fue fundamental realizar un levantamiento y análisis de información para definir de manera correcta y oportuna los requerimientos con los que este cumple.

Es importante comprender ampliamente las necesidades específicas del público objetivo, para estructurar y planificar de manera correcta y oportuna la propuesta de solución de software que cumple con los requerimientos exigidos.

El desarrollo del proyecto sigue unos lineamientos definidos y designados en cada una de las etapas, siempre en concordancia con la información obtenida y analizada en las primeras etapas de planificación.

La generación de un plan de pruebas y evaluación ha permitido encontrar falencias y vulnerabilidades, lo que a su vez permite generar planes de mejora continua.

9. RECOMENDACIONES

Extender los estudios relacionados con el ámbito de seguridad informática, debido a que hay diferentes individuos y organizaciones que constantemente buscan encontrar y explotar vulnerabilidades en los diferentes servicios web.

Planificar y desarrollar módulos complementarios e integrados que permitan generar una oferta de servicios integral de acuerdo con las necesidades cambiantes de las instituciones de educación.

Generar planes de evaluación y mejoramiento continuo, buscando estar a la vanguardia de las tecnologías de la información en la sociedad actual, contribuyendo al desarrollo y evolución de la humanidad, con responsabilidad social y ambiental.

Definir estrategias de desarrollo de software, como factor fundamental en la sociedad actual, generado valor en diferentes áreas y ámbitos, además de aportar significativamente a la digitalización y globalización, a la revolución 4.0.

10. REFERENCIAS BIBLIOGRÁFICAS

Plataforma de gestión académica, disciplinaria y administrativa. 2020. Proyecto GADA. Recuperado de: <http://gada.studiodti.com/>

Córdoba J. F (2006). Patentes Sobre Software. Desafío a La Propiedad Intelectual en la Sociedad de la Información. Publicado en Hacia una comprensión humana del derecho. Estudios en homenaje a Roberto Suarez Franco", Temis, pp. 123-144. Recuperado el 30 de noviembre de 2016 de: https://www.unisabana.edu.co/fileadmin/Documentos/Derecho/CEDEPI/Patentabilidad_d_el_Software.pdf

Pimiento E, L.E. (2009). La Evolución del Software. Recuperado de: <http://www.monografias.com/trabajos73/evolucion-software/evolucion-software.shtml>

Omatos S., A. (208). Software libre y Educación. Recuperado de: http://legalidad.aomatos.com/software_libre_y_educacin.html

Date, C.J. Introducción a los Sistemas de Bases de datos (2001). Ed. Pearson Education. 7ª edición. Recuperado el 23 de diciembre de 2016 de: <http://www.freelibros.org/ingenieria/introduccion-a-los-sistemas-de-bases-de-datos-7ma- edicion-c-j-date.html>

Richardas, M. Software Architecture Patterns (2015). Ed. O'Rally Media. Recuperado el 25 de diciembre de 2016 de <http://www.twirpx.com/file/1936162/>

Somerville, I. (2011). Ingeniería de software. México: Editorial Pearson Educación S.A.

Aparicio, A., Pilar, A. M. (2012). Módulo de Ingeniería de Software. UNAD.

GALVIS, A., (2000) Ingeniería de software educativo. Universidad de Los Andes. Ediciones UNIANDES. Colombia.

SÁNCHEZ, Jaime. (1993). Informática Educativa. Santiago de Chile: Editorial Universitaria.

Campderrich Falgueras, B. (2003). Ingeniería del software. Barcelona: Editorial UOC.

Moreno N., J. J. arquitecturas de software. Recuperado de: <http://pegasus.javeriana.edu.co/~mad/Arquitecturas%20de%20SW.pdf>

Murillo, J.L. (2009). Software educativo libre o gratuito para utilizar en el aula. Recuperado de: <http://www.edulibre.info/software-educativo-libre-o>

Noragueda, C. (2015). Programas de software libre para labores educativas. Recuperado de: <https://hipertextual.com/2015/06/interesantes-programas-de-software-libre-educativo>

Danforth, S. Tomlinson, C. Type theories and Object Oriented Programming (1998). System Technology Lab. P. 44

Cataldi, Z. (2000). Metodología de diseño, desarrollo y evaluación de software educativo. Recuperado de: <http://www.iidia.com.ar/rgm/tesistas/cataldi-tesisdemagistereninformatica.pdf>

Quintero, H., Portillo, L., Luque, R. Gonzales, M. (2005). Desarrollo de software educativo: una propuesta metodológica. Recuperado de: <https://www.redalyc.org/pdf/993/99318837004.pdf>

García P, Carlos I. (). Software de Libre Disponibilidad. Su utilización en el Estado. Recuperado de: <http://www.monografias.com/trabajos18/software-libre-estado/software-libre-estado.shtml>

Cobo, J. C. (2009). Conocimiento, creatividad y software libre: una oportunidad para la educación en la sociedad actual. Recuperado de: <http://www.uoc.edu/uocpapers/8/dt/esp/cobo.pdf>

Caro P., M. F., Toscazo M., R. E., Hernandez R., F. M., David L., M. E. (2009). DISEÑO DE SOFTWARE EDUCATIVO BASADO EN COMPETENCIAS. Recuperado de: <http://www.scielo.org.co/pdf/cein/v19n1/v19n1a05.pdf>

Castro D., R., (2008). El software educativo en el entorno de los medios de enseñanza. Recuperado de: <http://www.redalyc.org/pdf/1815/181518069007.pdf>

Booch, G; Maksimchuk, Robert; Object Oriented Analysis and Design with Applications (2007). Tercera Edición.

Culebro J., M., Gómez H., W. G., Torres S., S. (2006). Software libre vs software propietario. Ventajas y desventajas. Recuperado de: <http://www.rebelion.org/docs/32693.pdf>

Montes, D., Duque, V. (2009). CLASIFICACION DEL SOFTWARE EDUCATIVO. Recuperado de: <http://ucosweducativo.blogspot.com.co/2009/06/clasificacion-del-software-educativo.html>

Gonzales P., D. (2004). Software libre en los institutos. Recuperado de: https://www.cs.upc.edu/~tonis/daniel_gonzalez_pinyero.pdf

Tic portal. (2018). Gestión de proyectos. Recuperado de: <https://www.ticportal.es/glosario-tic/gestion-proyectos>

Abreviaturas

Abreviatura	Término
IHS	Intensidad horaria semanal
I	Primer periodo académico
II	Segundo periodo académico
III	Tercer periodo académico
IV	Cuarto periodo académico
R	Periodo académico de nivelación
Acum	Promedio acumulado
Aprox	Aproximadamente

Ecuaciones

Ecuación	Descripción
$\% \text{ Desempeño} = \frac{100 (Acum)}{5}$	Para calcular el porcentaje de desempeño, se toma el acumulado, se multiplica por 100 y el total se divide por 5, la calificación máxima posible.
$\text{if } (\% \text{ Desempeño}) \geq 4.0$	Desempeño Alto: El nivel de desempeño es alto cuando el promedio acumulado es superior a 4.0.
$\text{if } (\% \text{ Desempeño}) \geq 3.0 < 4.0$	Desempeño Medio: El nivel de desempeño es medio cuando el promedio acumulado es menor a 4.0 y superior a 3.0.
$\text{if } (\% \text{ Desempeño}) < 3.0$	Desempeño Bajo: El nivel de desempeño es bajo cuando el promedio acumulado es menor a 3.0.
$Acum = \frac{I + II + III + IV + R}{5}$	El promedio acumulado se calcula sumando la calificación obtenida en los periodos académicos y dividiendo el total por la cantidad de periodos.

11. ANEXOS

Anexo 1. Encuesta de levantamiento de información.

PREGUNTA	VALORACIÓN
¿Cuál es su nombre?	¿Útil para un requerimiento?
R:	SI_ NO
¿Cuál es su rol en la institución?	¿Útil para un requerimiento?
R:	SI_ NO
¿Cuáles tipos de usuario hay en la institución?	¿Útil para un requerimiento?
R:	SI_ NO
¿Cada cuánto registran las calificaciones de manera global?	¿Útil para un requerimiento?
R:	SI_ NO
¿Cuándo evalúan las asignaturas necesarias para cada grado?	¿Útil para un requerimiento?
R:	SI_ NO
¿Cuántas horas semanales de clase tiene cada asignatura?	¿Útil para un requerimiento?
R:	SI_ NO
¿Cada cuánto se realiza la entrega de informes académicos?	¿Útil para un requerimiento?
R:	SI_ NO
¿Cómo los acudientes hacen seguimiento a los estudiantes?	¿Útil para un requerimiento?
R:	SI_ NO
¿Cómo se presentan los resultados académicos de los estudiantes?	¿Útil para un requerimiento?
R:	SI_ NO
¿Cómo se realiza la matriculación de los docentes y estudiantes?	¿Útil para un requerimiento?
R:	SI_ NO
¿Cómo se asignan las asignaturas o materias académicas?	¿Útil para un requerimiento?
R:	SI_ NO
¿Cuáles son los índices de evaluación y desempeño?	¿Útil para un requerimiento?
R:	SI_ NO
¿Cómo se genera el registro de calificaciones en la institución?	¿Útil para un requerimiento?
R:	SI_ NO

Anexo 2. Matriz de descripción de actividades.

Actividad	Frecuencia	Complejidad	% Error
Matriculación de estudiantes	Anual	Media	5% Aprox
Registro de calificaciones	Quincenal	Alta	10% Aprox
Matriculación de docentes	Anual	Media	3% Aprox
Asignación de cursos	Anual	Media	5% Aprox
Asignación de asignaturas	Esporádico	Baja	1% Aprox
Elaboración de boletines académicos	Semestral	Media	5% Aprox
Informe académico a padres	Semestral	Media	5% Aprox

Anexo 3. Manual técnico.

- Ver y descargar el manual técnico en: [Ver pdf](#)



Anexo 4. Manual de usuario.

Ver y descargar el manual de usuario en: [Ver pdf](#)

