UNIVERSIDAD DEL BIO-BIO

Facultad de Ciencias Empresariales

Departamento Sistema de Informaciones

MEMORIA PARA OPTAR A TÍTULO DE INGENIERO DE EJECUCION EN COMPUTACION E INFORMATICA

“Desarrollo de sistema web para procesos educativos enfocado en la gestión académica para la Escuela Particular Enrique Molina Garmendia.”

Alumno : Nicolás Eliecer Castro Villa.

Profesor Guía : Sra. Elizabeth Grandon Toledo

CONCEPCIÓN, Marzo 2021

ÍNDICE:

CAPÍTULO I: DEFINICIÓN DE LA INSTITUCIÓN **7**

1.1 Descripción de la empresa. 7

1.2 Descripción de la problemática. 8

1.3 Justificación del proyecto. 9

CAPÍTULO II : PLANIFICACIÓN DEL PROYECTO **9**

2.1 Objetivos del proyecto 9

2.1.1 Objetivo general 9

2.1.2 Objetivos específicos 9

2.2 Ambiente de ingeniería de software 10

2.2.1 Metodología de desarrollo 10

2.2.2 Arquitectura 11

2.3 Definiciones, Siglas y Abreviaciones 12

2.3.1 Estándares de documentación 12

2.3.2 Técnicas y notaciones 12

CAPÍTULO III: ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE **13**

3.1 Alcances 13

3.2 Objetivos del software 14

3.2.1 Objetivos generales 14

3.2.2 Objetivos específicos 14

3.3 Descripción Global del Producto 14

3.3.1 Interfaz de Usuario 14

3.3.2 Interfaz de Hardware 15

3.3.3 Interfaz de software 16

3.3.4 Interfaz de Comunicación 16

3.4 Requerimientos específicos 16
3.4.1 Requerimientos funcionales 16
3.4.2 Interfaces externas de entrada 25
3.4.3 Interfaces externas de salida 25
3.4.4 Atributos del producto 26
CAPÍTULO IV ESTUDIO DE FACTIBILIDAD 26
4.1 Factibilidad Técnica 27
4.2 Factibilidad operativa 27
4.3 Factibilidad económica 28
4.4 Conclusión de factibilidad 29
CAPÍTULO V: ITERACIONES 29
5.1 Iteración 1 29
5.1.1 Análisis 29
5.1.1.1 Diagrama de flujo de datos 29
Diagrama de contexto 29
Diagrama de nivel superior 30
Diagrama de casos de uso 31
5.1.1.2 Especificación de los casos de uso y descripción 32
5.1.2 Diseño 50
5.1.2.1 Modelo Entidad Relación 50
5.1.2.2 Diseño de la base de datos 51
5.1.2.3 Diseño interfaz y navegación. 52
5.1.3 Codificación 55
5.1.4 Pruebas 61
5.1.4.1 Elementos de prueba 61
5.1.4.2 Especificación de las pruebas 61

5.1.4.3 Calendario de las pruebas 62
5.1.4.4 Detalle de las pruebas 62
5.1.4.5 Conclusión de las pruebas 64
5.2 Iteración II 65
5.2.1 Diagrama de flujo de datos 65
5.2.1.1 Diagrama de contexto 65
5.2.1.2 Diagrama de nivel superior 66
5.2.1.3 Diagrama de casos de uso 67
5.2.2 Especificación de casos de uso 68
5.2.3 Diseño 77
5.2.3.1 Modelo entidad relación 77
5.2.3.2 Diseño de la base de datos 78
5.2.3.3 Diseño de interfaz y navegación 78
5.2.4 Codificación 80
5.2.4 Pruebas 85
5.2.4.1 Elementos de prueba 85
5.2.4.2 Especificación de las pruebas 85
5.2.4.3 Calendario de las pruebas 86
5.2.4.4 Detalle de las pruebas 86
5.1.4.5 Conclusión de las pruebas 88
CAPÍTULO VI: PLAN DE CAPACITACIÓN Y PUESTA EN MARCHA 89
Plan de capacitación y entrenamiento. 89
Plan de implementación y puesta en marcha. 90
Resumen esfuerzo requerido. 91
8.1 Calcular UAW (Unadjusted actor weights) 91
8.2 Calcular UUCW (Unadjusted use case weights) 92
8.3 Calcular TCF (Technical Complexity Factors) 93

8.4 Calcular ECF (Environmental complexity factor) 94

8.5 Calcular UCP (use case points) 95

8.5.1 Calcular UUCP (Unadjusted use case points) 95

8.5.2 Calcular TCF (Technical complexity factor) 95

8.5.3 Calcular ECF (Environmental complexity factor) 95

8.5.4 Calcular UCP (Use case points) 95

8.6 Estimación final 95

CAPÍTULO VII: CONCLUSIONES **96**

CAPÍTULO VIII: BIBLIOGRAFÍA **97**

ANEXOS **98**

ÍNDICE DE TABLAS

Tabla 1: Navegadores soportados 20
Tabla 2: Requerimientos funcionales 28
Tabla 3: Requerimientos no funcionales. 29
Tabla 4: Interfaces externas de entrada 29
Tabla 5: Interfaces externas de salida 30
Tabla 7: Costo hardware 32
Tabla 8: Costo software 32
Tabla 9: Caso de uso 1 Iniciar sesión 37
Tabla 10: Caso de uso 2 Gestionar administrador 39
Tabla 11: Caso de uso 3 Gestionar administrativo 40
Tabla 12: Caso de uso 4 Gestionar alumno. 42
Tabla 13: Caso de uso 5 Gestionar profesor 43
Tabla 14: Caso de uso 6 Gestionar profesor 45
Tabla 15: Caso de uso 7 Ver calificación. 46
Tabla 16: Caso de uso 8 Ingresar calificación. 47
Tabla 17: Caso de uso 9 Modificar calificación. 48
Tabla 18: Caso de uso 10 Eliminar calificación. 49
Tabla 19: Caso de uso 11 Ver asistencia. 50
Tabla 20: Caso de uso 12 Ingresar asistencia. 51
Tabla 21: Caso de uso 13 Modificar asistencia 52
Tabla 22: Caso de uso 14 Eliminar asistencia 53
Tabla 23: Elementos de pruebas. 65
Tabla 24: Especificación de las pruebas. 66
Tabla 25: Calendario pruebas. 66
Tabla 26: Elemento de prueba iniciar sesión. 66

Tabla 27: Elemento de prueba registrar alumno. 67
Tabla 28: Elemento de prueba registrar calificación. 68
Tabla 29: Caso de uso 15 Generar certificado alumno regular. 72
Tabla 30: Caso de uso 16 Generar certificado notas. 73
Tabla 31: Caso de uso 17 Ver calendario. 74
Tabla 32: Caso de uso 18 Ingresar calendario. 75
Tabla 33: Caso de uso 19 Modificar calendario. 76
Tabla 34: Caso de uso 20 Eliminar calendario. 76
Tabla 35: Caso de uso 21 Ver documentos. 77
Tabla 36: Caso de uso 22 Ingresar documento. 78
Tabla 37: Caso de uso 23 Modificar documento. 79
Tabla 38: Caso de uso 24 Eliminar documento. 80
Tabla 39: Elementos de prueba iteración 2. 89
Tabla 40: Especificación de las pruebas iteración 2. 90
Tabla 41: Calendario de pruebas iteración 2. 90
Tabla 42: Elemento de prueba Ingresar calendario. 90
Tabla 43: Elemento de prueba Ingresar documento. 91
Tabla 44: Elemento de prueba Generar certificado alumno regular. 92
Tabla 45: Elemento de prueba Certificado notas. 92
Tabla 46: Calcular UAW. 95
Tabla 47: Cálculo UAW. 95
Tabla 48: Calcular UUCW. 96
Tabla 49: Cálculo UUCW. 97
Tabla 50: Cálculo TCF.. 98
Tabla 51: Cálculo ECF. 99

ÍNDICE DE FIGURAS.

Figura 1: Ciclo de vida metodología iterativa incremental 17

Figura 2 Diagrama modelo vista controlador 18

Figura 3: Diagrama de contexto iteración 1 37

Figura 4: Diagrama de nivel superior iteración 1 37

Figura 5: Diagrama de casos de uso iteración 1. 38

Figura 6: Modelo Entidad relación iteración 1 56

Figura 7: Diseño base de datos iteración 1 57

Figura 8: Login 58

Figura 9: Asignaturas profesor 58

Figura 10: Notas asignaturas profesor 59

Figura 11: Asistencia asignaturas profesor 59

Figura 12: Asignaturas alumnos 60

Figura 13: Conexión a la base de datos. 61

Figura 14: Ingresar alumno 62

Figura 15: Modificar alumno 62

Figura 16: Eliminar alumno 63

Figura 17: Ingresar notas 63

Figura 18: Modificar notas. 64

Figura 19: Eliminar nota. 64

Figura 20: Ingresar asistencia. 65

Figura 21: Modificar asistencia 66

Figura 22: Eliminar asistencia. 66

Figura 23: Diagrama de contexto iteración 2. 71

Figura 24: Diagrama de nivel superior iteración 2. 72

Figura 25: Diagrama de casos de uso iteración 2. 73

Figura 26: Modelo entidad relación iteración 2. 83
Figura 27: Diseño de la base de datos iteración 2. 84
Figura 28: Calendario. 84
Figura 29: Calendario. 85
Figura 30: Certificado alumno regular. 85
Figura 31: Certificado notas alumno. 86
Figura 32: Ingresar calendario. 86
Figura 33: Modificar calendario. 87
Figura 34: Eliminar calendario. 87
Figura 35: Ingresar documento. 88
Figura 36: Modificar documento. 89
Figura 37: Eliminar documento. 89
Figura 38: Certificado alumno regular. 90
Figura 39: Certificado notas. 90
Figura 40: Anexo Carta Gantt. 104

INTRODUCCIÓN

El año pasado una pandemia se generó esparciéndose por todo el mundo, lo que en nuestro país ha llevado al cese de algunas actividades esenciales, una de ellas es la educación. Las clases presenciales han sido suspendidas y sustituidas por clases "online", lo que ha llevado a los establecimientos del país a intentar sobrellevar esta nueva dinámica, pero también ha salido a la luz algunas fallas que hacen que esta situación se vuelva un poco más complicada.

En el siguiente informe se presenta la documentación del proceso de "desarrollo de sistema web para procesos educativos enfocado en la gestión académica para la Escuela Particular Enrique Molina Garmendia" la cual está orientada a los alumnos y profesores del establecimiento. Este proceso se llevó a cabo utilizando una metodología iterativa-incremental que permite la entrega de partes funcionales del sistema, además de utilizar una arquitectura modelo vista controlador que separa las operaciones del sistema de la vista del usuario.

En el Capítulo 1 del informe se presenta la institución y se describe la problemática a abordar en este proyecto. Seguido de esto se presenta la planificación del proyecto, quiere decir, los objetivos de este, su metodología y su arquitectura. Pasando al Capítulo 3 se encuentra la especificación de los requerimientos que tendrá el sistema los cuales son necesarios para su funcionamiento, además de nombrar los alcances que tendrá el proyecto, en otras palabras, lo que hará y lo que no hará el sistema. Los estudios de factibilidad que son presentados en el Capítulo 4, se componen de la factibilidad técnica, operativa y económica, los cuales son necesarios para determinar si el proyecto es factible o no. En el Capítulo 5 se presentan las iteraciones del proyecto, paso por paso, desde el análisis de cada iteración hasta las pruebas de esta. En el penúltimo capítulo se presentan el plan de capacitación y entrenamiento, en el cual se da a conocer cómo va a ser la capacitación de los usuarios en el manejo del software; también se presenta el plan de implementación y puesta en marcha, el cual describe cómo se implementará el sistema en la institución, y el resumen del esfuerzo requerido para el

desarrollo del sistema. Ya en el último capítulo, se presentan las conclusiones que se generaron a través del desarrollo del proyecto, como también se muestra la bibliografía y los anexos que fueron utilizados para el desarrollo de este informe.

CAPÍTULO I: DEFINICIÓN DE LA INSTITUCIÓN

1.1 Descripción de la empresa.

Colegio Enrique Molina Garmendia, fundado el 30 de septiembre de 1960, ubicado en la comuna de Hualpén, Octava región, cuenta con cursos desde kinder a octavo básico.

Antecedentes generales del establecimiento

- **Nombre Establecimiento: Escuela Particular Enrique Molina Garmendia**
- **Directora: Jacqueline Del Carmen Bustamante Vergara**
- **Tipo: Particular subvencionado**
- **Dirección: Holanda 2569**
- **Comuna: Hualpén**
- **Región: Región del Biobío**

Misión

"Se plantea como misión, tener un propósito fundamental, educar, promoviendo y asegurando una educación integral de nuestros alumnos (as), en lo intelectual, afectivo, moral, ético, físico, artístico y deportivo, como una forma de desarrollo total de la persona que ayude a su formación educacional con sentido de pertenencia a un grupo social de trabajo en equipo, de valores de responsabilidad, lealtad y compromiso, desde una perspectiva de sana competencia, diversión y formación para la vida"[1].

Visión

"Se aspira a mantener en el tiempo una educación de calidad, asumir y fortalecer el deporte escolar, a través del desarrollo de actividades físicas, artísticas, deportivas y culturales, generando en su desarrollo oportunidades de continuación"[1].

[1] (2021) ESCUELA PARTICULAR ENRIQUE MOLINA GARMENDIA | #SEA2021

1.2 Descripción de la problemática.

La Escuela Particular Enrique Molina Garmendia (Así se llama el establecimiento) cuenta con los recursos e infraestructura para su correcto funcionamiento presencial, utilizan un libro de clases para llevar el registro de las notas parciales, anotaciones de los alumnos, asistencia a clases, planificaciones de evaluaciones. Toda documentación que se requiera por parte del alumno se obtiene de manera presencial (certificado de alumno regular, certificado de notas, situación académica).

Debido a la catástrofe que afronta el país, el establecimiento ha visto suspendidas sus actividades presenciales, lo que ha demostrado lo frágil que es el sistema actual, ya que la gran mayoría de los procesos se hace de manera presencial y el establecimiento no contaba con las herramientas y recursos necesarios para afrontar este tipo de situaciones.

1.3 Justificación del proyecto.

En la actualidad, debido al estado de catástrofe (pandemia) que atraviesa el país, el establecimiento no puede funcionar normalmente, por lo que los puntos señalados en la situación actual no pueden llevarse a cabo. Es por esto que el establecimiento se ha visto obligado a optar por alternativas que muchas veces no son la mejor opción de comunicación (Zoom, Correo Electrónico, Google Meets, Google Drive, etc), para funcionar de la manera más normal posible.

El establecimiento carece de un sistema corporativo online que permita la administración del material pedagógico (presentaciones, guías, clases), calificaciones de los estudiantes (libro de clases), calendarización de evaluaciones, obtención de certificados (notas, alumno regular), entre otros.

Es por esto que la planificación del proyecto se centra en apoyar los diferentes procesos administrativos y pedagógicos del establecimiento con el desarrollo de un sistema web que cubra algunas de las falencias provocadas por la pandemia.

CAPÍTULO II : PLANIFICACIÓN DEL PROYECTO

2.1 Objetivos del proyecto

2.1.1 Objetivo general

Implementar un sistema web para la gestión académica de la escuela particular Enrique Molina Garmendia.

2.1.2 Objetivos específicos

- Desarrollar una plataforma web tipo intranet que permita gestionar procesos administrativos (certificados de alumno regular, licencias médicas de alumnos, calendario de evaluaciones) y gestionar los procesos de evaluación del estudiante (notas del alumno tanto parciales como finales, informe de notas, registro de asistencia, libro de anotaciones positivas y negativa, etc.).
- Desarrollar una plataforma tipo Moodle que permita reunir material pedagógico (guías, evaluaciones, presentaciones, etc.) que sean descargables por los estudiantes.

2.2 Ambiente de ingeniería de software

2.2.1 Metodología de desarrollo

La metodología a utilizar en este proyecto es la iterativa incremental, ya que esta metodología entrega el software en partes pequeñas llamadas iteraciones, donde cada iteración tiene varias etapas, iniciando por el análisis y terminando con las pruebas en cada iteración, permitiendo así que el proyecto crezca de forma incremental (ver Figura 1). La ventaja de esta metodología es que los usuarios no tienen que esperar hasta que el sistema esté completo para su uso, dado a que el primer incremento cumple con los requerimientos más importantes. Cada iteración genera una evolución del producto final a partir de los resultados de las iteraciones anteriores, así añadiendo nuevos objetivos y requisitos.

Se eligió esta metodología ya que al entregar pequeñas iteraciones, los usuarios pueden familiarizarse con la plataforma y de ser necesario algún cambio, este pueda ser detectado a tiempo, ya que es menos perjudicial afrontar cambios en los inicios de las iteraciones que al momento de la entrega final. En la figura 1 se muestra el ciclo de vida de la metodología iterativa incremental.

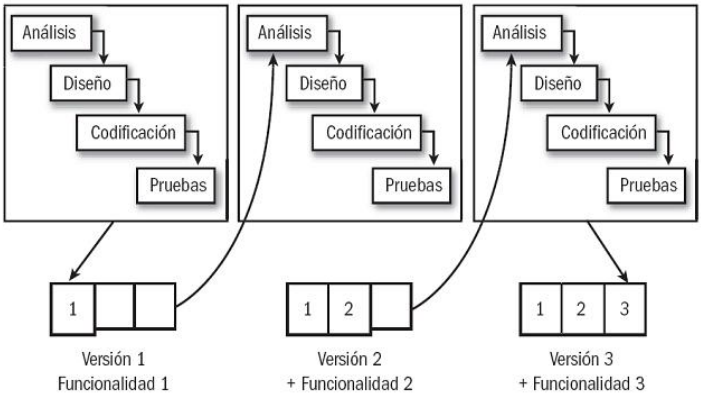

Figura 1:Ciclo de vida metodología iterativa incremental (http://alredsa.blogspot.com/2017/07/ciclo-de-vida-de-proyectos-clasico.html)

2.2.2 Arquitectura

La arquitectura de software a utilizar en este proyecto es el modelo vista controlador (MVC), el cual separa los datos de una aplicación, su interfaz y la lógica en tres componentes.

- **Modelo** : Representa la estructura de los datos, va a contener funciones que nos ayudan a mostrar, insertar, modificar y eliminar información de la base de datos.
- **Vista** : Es la información que se presenta al usuario, en otras palabras, es la parte donde se interactúa con el usuario y se especifican cosas como: formularios, posición de datos y cómo estos se desplegarán en la pantalla.
- **Controlador**: Es el intermediario entre el modelo y la vista y cualquier otro recurso necesario para procesar una solicitud HTTP, y con esto, poner a funcionar una página. La Figura 2 muestra en forma gráfica el diagrama modelo vista controlador.

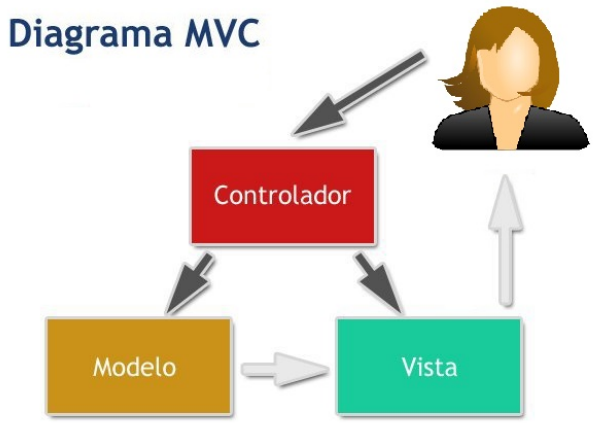

Figura 2 Diagrama modelo vista controlador
(https://itsoftware.com.co/content/modelo-vista-controlador-mvc-sirve/)

2.3 Definiciones, Siglas y Abreviaciones

2.3.1 Estándares de documentación

● Estándar para los requerimientos funcionales: Adaptación basada en IEEE Software requirements Specifications Std 830-1998.

2.3.2 Técnicas y notaciones

Algunas de las siguientes definiciones se encuentran en Pressman (2010) y Sommerville (2011):

● Hardware: conjunto de aparatos de una computadora.(dle.rae.es/hardware)

● Software: Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora. (dle.rae.es/software)

● Casos de Uso - Lenguaje unificado y modelado (UML): Se utiliza para entender más claro desde una notación gráfica, en el cual se describirán los casos de uso para así entender el sistema y disminuir la ambigüedad.

● Modelo Entidad-Relación (MER): Modelo lógico que expone las relaciones de las distintas entidades dentro del sistema.

● Modelo Relacional (MR): Muestra las relaciones de las tablas en la base de datos del sistema.

● Base de Datos (BD): Este término será utilizado cuando nos queramos referir a una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible.

● DFD: Diagrama de Flujo de Datos.

● Script: Conjunto de órdenes guardadas en un archivo de texto.

● Modelo: Representación de los datos que maneja el sistema.

● Vista: Interfaz de usuario, que compone la información que se envía al cliente.

● Controlador: Intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

CAPÍTULO III: ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

3.1 Alcances

- El sistema debe permitir desplegarse en distintas plataformas y sistemas operativos sin necesidad de una adaptación previa.
- El sistema debe ser amigable e intuitivo.
- El sistema será utilizado por administradores, administrativos, profesores y alumnos del establecimiento.
- El sistema debe permitir a los profesores la administración de las notas y promedios de las asignaturas que imparten respectivamente.
- El sistema debe permitir a los usuarios visualizar y obtener los diferentes certificados con los que cuenta el establecimiento (notas, alumno regular).
- El sistema debe permitir a los profesores administrar la asistencia a clases de los alumnos de un curso.
- El sistema no permite la administración de asistencia de los profesores al establecimiento.

3.2 Objetivos del software

3.2.1 Objetivos generales

El sistema permitirá la gestión de ciertas áreas del establecimiento, permitiendo al usuario visualizar (alumnos) y administrar (profesores) la situación académica (notas parciales, notas finales, promedios, etc.) de cada alumno, permitirá la obtención de certificados (alumno regular, concentración de notas) de manera digital.

3.2.2 Objetivos específicos

- Gestionar las asignaturas de los alumnos de modo que el profesor cuente con un rol dentro del sistema que le permita ingresar notas, agendar evaluaciones, ingresar asistencia de alumnos y distribuir material pedagógico.

- Gestionar los cursos del establecimiento de modo que se le permita al administrador ingresar alumnos nuevos, asignar profesores a las asignaturas de dicho curso y asignar profesor jefe al curso deseado.
- Gestionar el personal del establecimiento de modo que se le permita al administrador ingresar personal nuevo, editar su información y/o deshabilitarlos del sistema.
- Permitir la obtención de certificados (notas, alumno regular) de los estudiantes.

3.3 Descripción Global del Producto

3.3.1 Interfaz de Usuario

Características:

El sistema contiene:

- Login de ingreso al sistema para administradores, administrativos, alumnos y profesores.
- Módulo principal con información del usuario actual de la sesión.
- Módulo para administrar (crear, ver, editar, eliminar) los alumnos del sistema.
- Módulo para administrar (crear, ver, editar, eliminar) los profesores del sistema.
- Módulo para administrar (crear, ver, editar, eliminar) los administrativos del sistema.
- Módulo para administrar (crear, ver, editar, eliminar) los administradores del sistema.
- Módulo para obtener certificados (alumno regular, notas).
- Módulo para listar los cursos y asignaturas dictadas.
- Módulo para administrar (crear, ver, editar, eliminar) las notas y promedios de los alumnos.
- Módulo para administrar (crear, ver, editar, eliminar) la asistencia de los alumnos.

- Módulo para administrar (crear, ver, editar, eliminar) un calendario de eventos futuros de los alumnos.

Elementos de interfaz:

- Nombre del sistema.
- Logo del sistema.
- Menú de navegación principal.
- Submenús para navegar entre módulos.
- Elemento de regreso al inicio (botón “home”).

3.3.2 Interfaz de Hardware

- Entrada Estándar: Mouse y Teclado.
- Salida Estándar: Monitor.
- Tarjeta de Red: Wifi o Ethernet.

3.3.3 Interfaz de software

El sistema es independiente por lo cual no tendrá relaciones con otros sistemas de software externos excluyentes al sistema operativo y navegador.

La tabla a continuación muestra los navegadores que soportan el sistema.

Navegadores soportados

Nombre	**Abreviación**	**Versión**
Google Chrome	Chrome	
Microsoft Edge	Edge	88.0.705.63
Mozilla Firefox	Firefox	85.0.2
Opera	Opera	74.0.3911.75

Tabla 1: Navegadores soportados

3.3.4 Interfaz de Comunicación

Para que el cliente pueda interactuar con el software es necesario contar con los siguientes protocolos de comunicación:

- Protocolo TCP/IP: Este protocolo representa todas las reglas de comunicación para internet y basa su funcionamiento en las direcciones IP. Esto permite una comunicación fiable y segura de extremo a extremo.
- HTTP: El significado de esta sigla es HyperText Transfer Protocol (Protocolo de transferencia de hipertexto) es el método más común para intercambiar información en la web. La comunicación entre el servidor y el cliente se realizará a través del puerto TCP: 80 para el tráfico HTTP.

3.4 Requerimientos específicos

3.4.1 Requerimientos funcionales

A continuación se detallan los requerimientos funcionales que posee el sistema.

ID	NOMBRE	DESCRIPCIÓN
RF_01	Iniciar sesión	El sistema debe permitir al usuario ingresar mediante su rut y contraseña previamente creado.
RF_02	Ver Alumnos	El sistema debe permitir al usuario con el rol correspondiente (administrador - administrativo) ver los alumnos ingresados al sistema.

RF_03	Ingresar alumno	El sistema debe permitir al usuario con el rol correspondiente (administrador - administrativo) ingresar nuevos alumnos al sistema.
RF_04	Modificar alumno	El sistema debe permitir al usuario con el rol correspondiente (administrador - administrativo) modificar los datos ingresados de un alumno.
RF_05	Eliminar alumno	El sistema debe permitir al usuario con el rol correspondiente (administrador - administrativo) eliminar a un alumno ingresado en el sistema.
RF_06	Ver profesores	El sistema debe permitir al usuario con el rol correspondiente (administrador-administrativo) ver los profesores ingresados al sistema.
RF_07	Ingresar profesor	El sistema debe permitir al usuario con el rol correspondiente (administrador-administrativo) ingresar un nuevo profesor al sistema.
RF_08	Modificar profesor	El sistema debe permitir al usuario con el rol correspondiente (administrador-administrativo) modificar los datos ingresados de un profesor.

RF_09	Eliminar profesor	El sistema debe permitir al usuario con el rol correspondiente (administrador-administrativo) eliminar un profesor del sistema.
RF_10	Ver administradores	El sistema debe permitir al usuario con el rol correspondiente (administrador) ver los administradores ingresados al sistema.
RF_11	Ingresar administrador	El sistema debe permitir al usuario con el rol correspondiente (administrador) ingresar un nuevo administrador al sistema.
RF_12	Modificar administrador	El sistema debe permitir al usuario con el rol correspondiente (administrador) modificar los datos ingresados de un administrador.
RF_13	Eliminar administrador	El sistema debe permitir al usuario con el rol correspondiente (administrador) eliminar un administrador del sistema.
RF_14	Ver administrativos	El sistema debe permitir al usuario con el rol correspondiente (administrador) ver los administrativos ingresados al sistema.
RF_15	Ingresar administrativo	El sistema debe permitir al usuario con el rol correspondiente

		(administrador) ingresar un nuevo administrativo al sistema.
RF_16	Modificar administrativo	El sistema debe permitir al usuario con el rol correspondiente (administrador) modificar los datos ingresados de un administrativo.
RF_17	Eliminar administrativo	El sistema debe permitir al usuario con el rol correspondiente (administrador) eliminar un administrativo del sistema.
RF_18	Ver asignaturas	El sistema debe permitir al usuario con el rol correspondiente (administrador - administrativo) ver las asignaturas asignadas de un profesor.
RF_19	Ingresar asignatura	El sistema debe permitir al usuario con el rol correspondiente (administrador - administrativo) asignar una asignatura a un profesor.
RF_20	Modificar asignatura	El sistema debe permitir al usuario con el rol correspondiente (administrador - administrativo) modificar los datos de una asignatura asignada de un profesor.
RF_21	Eliminar asignatura	El sistema debe permitir al usuario con el rol correspondiente (administrador -

		administrativo) eliminar una asignatura asignada de un profesor.
RF_22	Ver calificaciones	El sistema debe permitir al usuario con el rol correspondiente (profesor - alumno) ver las calificaciones de una asignatura.
RF_23	Ingresar calificación	El sistema debe permitir al usuario con el rol correspondiente (profesor) ingresar en el sistema una calificación de una asignatura a un curso.
RF_24	Modificar calificación	El sistema debe permitir al usuario con el rol correspondiente (profesor) modificar los datos ingresados a una calificación de una asignatura a un curso.
RF_25	Eliminar calificación	El sistema debe permitir al usuario con el rol correspondiente (profesor) eliminar una calificación de una asignatura a un curso.
RF_26	Ver asistencia	El sistema debe permitir al usuario con el rol correspondiente (profesor - alumno) ver la asistencia de una asignatura del curso.
RF_27	Ingresar asistencia	El sistema debe permitir al usuario con el rol correspondiente (profesor) ingresar en el sistema una asistencia de una asignatura del curso.

RF_28	Modificar asistencia	El sistema debe permitir al usuario con el rol correspondiente (profesor) modificar los datos de una asistencia de una asignatura del curso.
RF_29	Eliminar asistencia	El sistema debe permitir al usuario con el rol correspondiente (profesor) eliminar del sistema una asistencia de una asignatura del curso.
RF_30	Ver calendario	El sistema debe permitir al usuario con el rol correspondiente (profesor-alumno) ver el calendario de futuros eventos.
RF_31	Ingresar calendario	El sistema debe permitir al usuario con el rol correspondiente (profesor) ingresar un nuevo evento en el calendario.
RF_32	Modificar calendario	El sistema debe permitir al usuario con el rol correspondiente (profesor) modificar un evento futuro ingresado en el calendario
RF_33	Eliminar calendario	El sistema debe permitir al usuario con el rol correspondiente (profesor) eliminar un evento futuro ingresado en el calendario
RF_34	Generar certificado alumno regular	El sistema debe permitir al usuario con el rol correspondiente (Alumno) generar certificado de alumno regular.

RF_35	Exportar certificado alumno regular	El sistema debe permitir al usuario con el rol correspondiente (Alumno) exportar los certificados (alumno regular, notas) del alumno correspondiente.
RF_36	Generar reportes de calificaciones	El sistema debe permitir al usuario con el rol correspondiente (Profesor, Alumno) generar reportes con las calificaciones pertinentes.
RF_37	Exportar reporte de calificaciones	El sistema debe permitir al usuario con el rol correspondiente (Profesor, Alumno) exportar a una planilla (pdf) las calificaciones pertinentes.
RF_38	Ver material pedagógico asignaturas	El sistema debe permitir al usuario con el rol correspondiente (profesor - alumno) ver el material pedagógico ingresado en una asignatura.
RF_39	Subir material pedagógico	El sistema debe permitir al usuario con el rol correspondiente (Profesor) subir material pedagógico a una asignatura que tenga asignada.
RF_40	Modificar material pedagógico.	El sistema debe permitir al usuario con el rol correspondiente (Profesor) modificar el material pedagógico ingresado en una asignatura que tenga asignada.
RF_41	Eliminar material	El sistema debe permitir al

	pedagógico	usuario con el rol correspondiente (Profesor) eliminar el material pedagógico ingresado en una asignatura que tenga asignada.
RF_42	Ver curso.	El sistema debe permitir al usuario con el rol correspondiente (Profesor) ver las estadísticas de su curso asignado, ya sea promedio generales de los alumnos o asistencia a clases.
RF_43	Cerrar sesión	El sistema debe permitir al usuario cerrar su sesión

Tabla 2: Requerimientos funcionales

Requerimientos no funcionales

ID	**Nombre**	**Descripción**
RNF_01	Tiempo de Respuesta a Solicitudes	El sistema debería responder de forma inmediata a las solicitudes, de acuerdo a los recursos y velocidad de las consultas a la base de datos
RNF_02	Estabilidad	El sistema deberá asegurar en su funcionalidad que no tendrá fallas y sus procesos serán continuos
RNF_03	Soporte de navegadores (Portabilidad)	La interfaz web del sistema debe funcionar en múltiples navegadores tales como: Mozilla Firefox, Google

		Chrome, como también para dispositivos móviles.
RNF_04	Intuitivo y Navegable	Los módulos deben ser entendibles por el usuario, con un lenguaje claro y en español.
RNF_05	Usabilidad	El tiempo de aprendizaje del sistema por un usuario 50 deberá ser menor o igual a 3 horas. El sistema debe proporcionar mensajes de error que sean informativos y orientados al usuario final.
RNF_06	Seguridad	El sistema debe asegurar la seguridad de los datos del usuario.
RNF_07	Eficiencia	Toda funcionalidad del sistema y transacción debe responder al usuario en menos de 10 segundos.

Tabla 3: Requerimientos no funcionales.

3.4.2 Interfaces externas de entrada

ID	Nombre del ítem	Detalle de datos	Medio de entrada
IE_01	Datos del administrativo	Nombres, Apellidos, Rut, Correo	Teclado
IE_02	Datos del alumno	Nombres, Apellidos, Rut, Correo, Dirección	Teclado
IE_03	Datos del profesor	Nombres, Apellidos, Rut, Correo	Teclado

Tabla 4: Interfaces externas de entrada

3.4.3 Interfaces externas de salida

ID	Nombre del item	Detalle de datos	Medio de salida
DE_01	Certificado de alumno regular	Documento que acredita la pertenencia al establecimiento.	PDF
DE_02	Certificado de notas	Documento que muestra y promedian las notas del alumno.	PDF

Tabla 5: Interfaces externas de salida

3.4.4 Atributos del producto

- **USABILIDAD-OPERATIVIDAD:** el sistema entrega mensajes informativos al usuario ante cualquier error ocurrido al ingresar, actualizar datos o en alguna situación en particular.
- **FUNCIONALIDAD-SEGURIDAD:** El sistema restringe el acceso a sus funciones mediante el mecanismo de autenticación (login). Los usuarios solo pueden ingresar al sistema mediante su Rut y contraseña, estos deben estar registrados en la base de datos por el administrador.
- **USABILIDAD-FACILIDAD DE APRENDIZAJE:** El sistema está diseñado para que los usuarios que hagan uso de este sistema requieren un esfuerzo mínimo para utilizar las distintas funciones que este provee.

CAPÍTULO IV ESTUDIO DE FACTIBILIDAD

El estudio de factibilidad se compone en 3 aspectos a considerar:

4.1 Factibilidad Técnica

El desarrollo del proyecto está a cargo del alumno Nicolás Castro Villa, memorista de la carrera Ingeniería de Ejecución en Computación e Informática.

Capacidades técnicas del desarrollador:

- Conocimiento intermedio en PHP.
- Conocimiento básico en HTML y CSS.
- Conocimiento intermedio en JavaScript.
- Conocimiento intermedio en SQL.
- Conocimiento intermedio en Framework Laravel.

Recursos de hardware disponibles:

- Computador de escritorio o portátil con las siguientes características:
    - Sistema operativo: Windows 10.
    - Procesador Intel Core i3 / Intel Core i5.
    - Disco duro: 1TB.
    - Ram: 8GB DRAM.
- Conexión a internet.

Actualmente el desarrollador cuenta con el equipo y recursos para cumplir con los requerimientos y objetivos para la implementación del sistema

4.2 Factibilidad operativa

El sistema ofrece una solución para los procedimientos administrativos de la Escuela Enrique Molina Garmendia, con el fin de optimizar, agilizar y organizar los procesos tales como registros, reportes, búsquedas, reservas de atención. Dentro de los beneficios que traerá el sistema se encuentran la drástica reducción de tiempo debido a que actualmente no cuentan

con un sistema digital interno, además de reducir lo mayormente posible en el contacto personal entre alumnos o apoderados con los profesores, también.

4.3 Factibilidad económica

Costo de Hardware:

El establecimiento al no contar con un servidor propio, se hará arriendo de un hosting que tiene un valor de:

	Cantidad	**Precio Unitario**	**Precio Total**
Hosting	24 (2 años)	$3.242	$77.800

Tabla 7: Costo hardware

Costo de Software:

Software	**Valor**
PHP 7.2.19	0
MySQL 5.7.24	0
Framework Laravel 7.29	0
Total	0

Tabla 8: Costo software

Costo de Recursos humanos:

Se estima que el sueldo promedio de un ingeniero en informática al primer año de egreso es de $1.047.984, el proyecto tiene una duración de 4 meses aproximadamente, por lo que el costo del ingeniero a realizar el proyecto es de $4.191.936. Sin embargo, al ser un proyecto que se realiza como un proyecto final de carrera, este no tiene costo asociado.

Costos de implementación y mantención:

- Los costos de codificación son nulos dado que este proyecto se desarrolla como proyecto final de título.
- Los costos de mantención del hosting vienen incluidos en el plan adquirido y corren por parte del cliente.

4.4 Conclusión de factibilidad

Podemos decir que, gracias al nulo costo de implementación debido a que se trata de un proyecto final de carrera y al bajo costo del hosting ($38.900 anual), se concluye que el proyecto es totalmente factible debido a la necesidad de contar con un sistema interno que sea de forma on-line.

CAPÍTULO V: ITERACIONES

5.1 Iteración 1

En esta primera iteración se desarrollarán los requerimientos funcionales que se consideran de mayor importancia, los cuales van desde el requerimiento funcional nº 1 hasta el nº 29 y nº 42

5.1.1 Análisis

5.1.1.1 Diagrama de flujo de datos

Diagrama de contexto

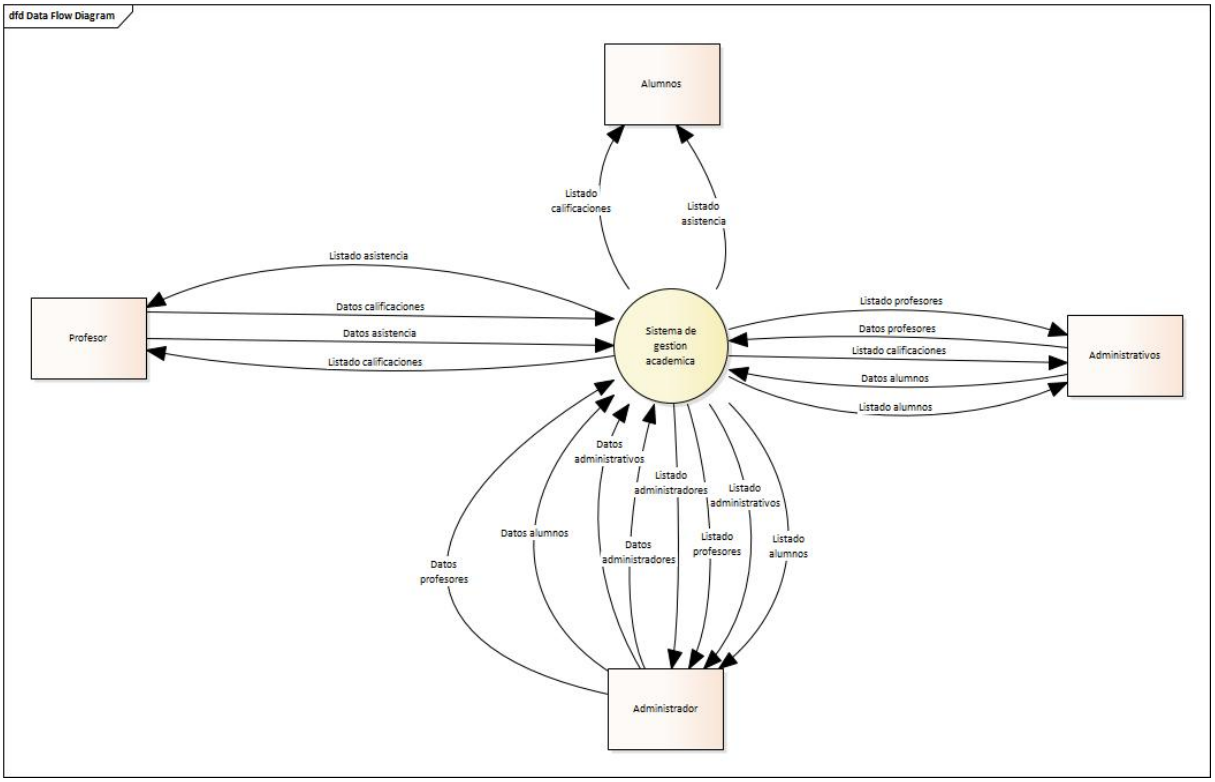

Figura 3: Diagrama de contexto iteración 1

Diagrama de nivel superior

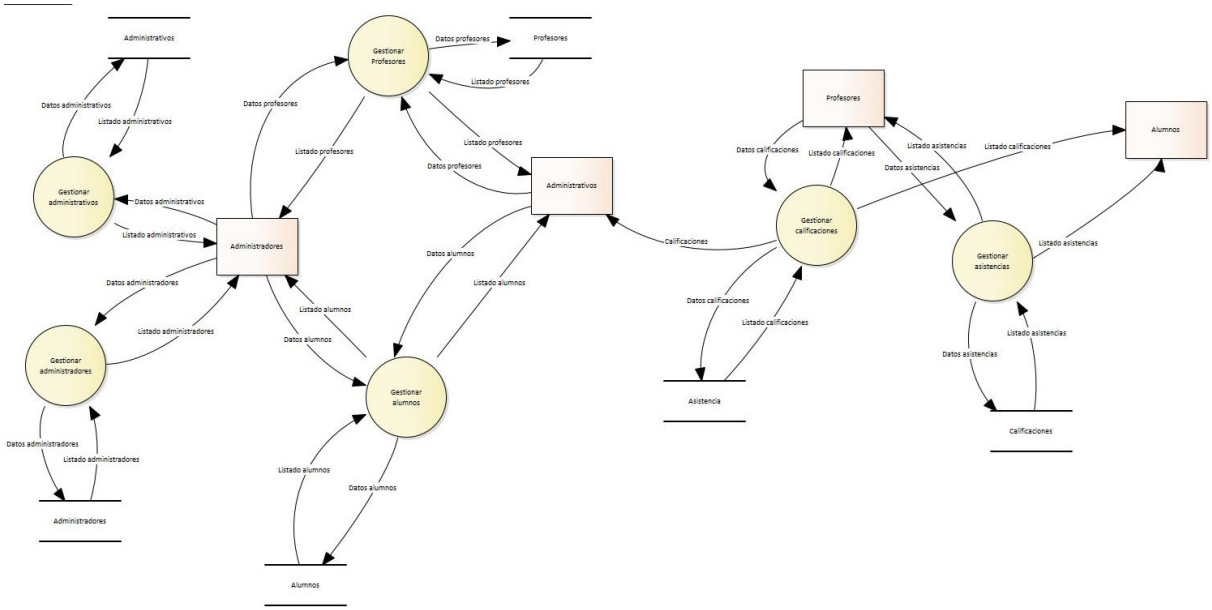

Figura 4: Diagrama de nivel superior iteración 1

Diagrama de casos de uso

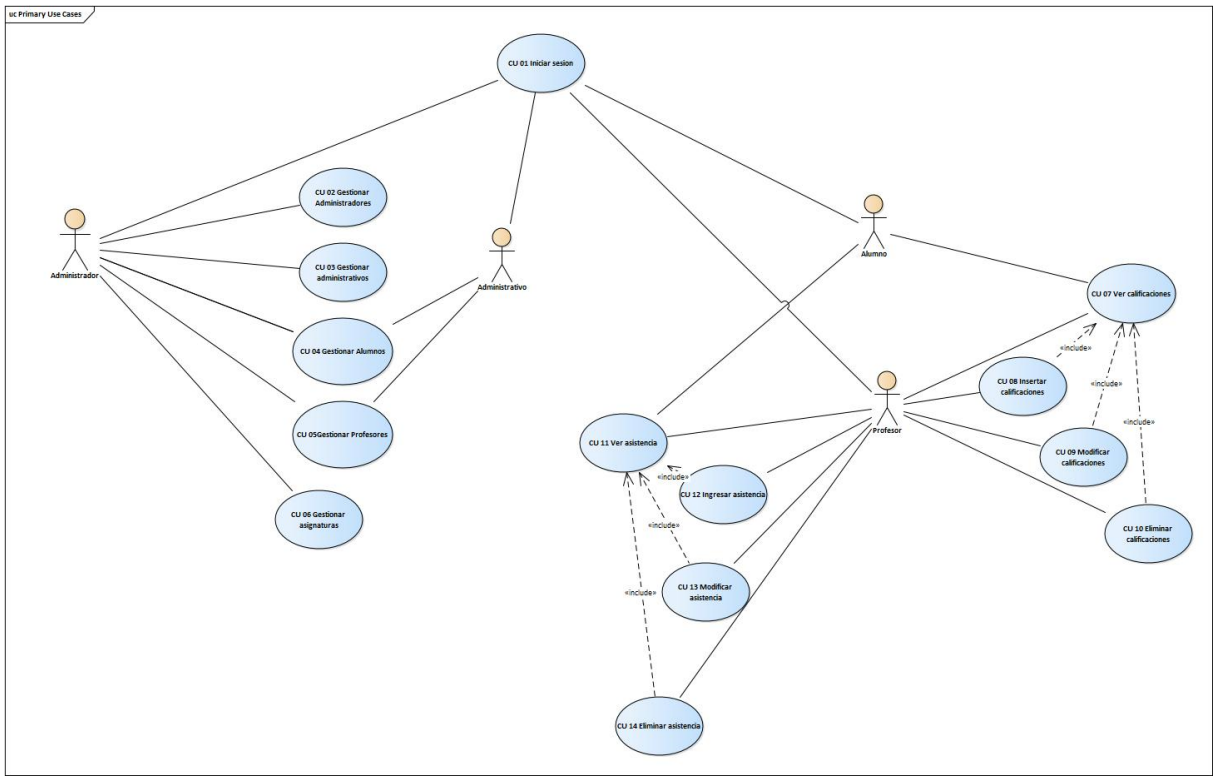

Figura 5: Diagrama de casos de uso iteración 1.

5.1.1.2 Especificación de los casos de uso y descripción

CU 01 Iniciar sesión

Casos de uso: Iniciar sesión	
Actor: Administrador, profesor, alumno	
Descripción: El actor inicia sesión en el software mediante un usuario y una contraseña	
Precondiciones: El usuario y la contraseña deben existir en la base de datos	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa su usuario y contraseña y le da click en el botón "Iniciar sesión"	2) El sistema verifica si el usuario y la contraseña en la base de datos coinciden con los ingresados, de ser así, el sistema devuelve la pantalla principal.
Escenario Alternativo	
Actor	**Sistema**
	2a) Si los datos no coinciden con los registrados en la base de datos, o no existen, el sistema devuelve el error en pantalla.
3) El actor deberá ingresar sus datos nuevamente.	
Postcondiciones: Inicio de sesión exitoso, se ingresa al sistema.	

Tabla 9: Caso de uso 1 Iniciar sesión

CU 02 Gestionar administrador

Casos de uso: Gestionar administrador	
Actor: Administrador	
Descripción: El actor puede ver, ingresar, modificar, eliminar un administrador	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de administrador	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de administrador	2) El sistema redirecciona al módulo de administrador y devuelve una vista con una lista de todos los administradores ingresados con sus respectivas opciones.
3) El actor selecciona el botón + para ingresar un nuevo administrador	4) El sistema despliega una ventana con los campos a rellenar para el ingreso del nuevo administrador
5) El actor llena los campos con la información pertinente y hace click en el botón “Guardar”	6) El sistema ingresa el nuevo administrador a la base de datos y redirecciona al módulo de administrador con la lista de los administradores registrados
7) El actor hace clic en el botón de “Modificar”	8) El sistema despliega una ventana con los datos registrador del administrador a modificar
9) El actor modificar los campos que desee modificar y hace click en “Guardar”	10) El sistema modifica la información que fue cambiada en los campos de la ventana que se desplegó y redirecciona al módulo de administrador con la lista de los administradores registrados
11) El actor hace clic en el botón de “Eliminar”	12) El sistema despliega una ventana con el nombre del administrador seleccionado y una advertencia de

	confirmación.
13) El actor confirma su selección de eliminación y hace click en el botón “Eliminar”.	14) El sistema elimina la información del administrador de la base de datos y redirecciona al módulo de administrador con la lista de los administradores registrados.
Escenario Alternativo	
Actor	**Sistema**
5a) El actor deja un campo en blanco y hace clic en “Guardar”	5b) El sistema informa de la información faltante al actor y que es obligatoria.
9a) El actor deja un campo en blanco y hace clic en “Guardar”	9b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se visualiza, busca, ingresa, modifica y/o elimina un administrador.	

Tabla 10: Caso de uso 2 Gestionar administrador

CU 03 Gestionar administrativos

Casos de uso: Gestionar administrativos	
Actor: Administrador	
Descripción: El actor puede ver, ingresar, modificar, eliminar un administrativo	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de administrador	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de administrativos	2) El sistema redirecciona al módulo de administrativos y devuelve una vista con una lista de todos los administrativos ingresados con sus respectivas opciones.
3) El actor selecciona el botón + para ingresar un nuevo administrativo	4) El sistema despliega una ventana con los campos a rellenar para el ingreso

	del nuevo administrativo
5) El actor llena los campos con la información pertinente y hace click en el botón “Guardar”	6) El sistema ingresa el nuevo administrativo a la base de datos y redirecciona al módulo de administrativos con la lista de los administrativos registrados
7) El actor hace clic en el botón de “Modificar”	8) El sistema despliega una ventana con los datos registrador del administrativo a modificar
9) El actor modificar los campos que desee modificar y hace click en “Guardar”	10) El sistema modifica la información que fue cambiada en los campos de la ventana que se desplegó y redirecciona al módulo de administrativos con la lista de los administrativos registrados
11) El actor hace clic en el botón de “Eliminar”	12) El sistema despliega una ventana con el nombre del administrativo seleccionado y una advertencia de confirmación.
13) El actor confirma su selección de eliminación y hace click en el botón “Eliminar”.	14) El sistema elimina la información del administrativo de la base de datos y redirecciona al módulo de administrativo con la lista de los administrativos registrados.
Escenario Alternativo	
Actor	**Sistema**
5a) El actor deja un campo en blanco y hace clic en “Guardar”	5b) El sistema informa de la información faltante al actor y que es obligatoria.
9a) El actor deja un campo en blanco y hace clic en “Guardar”	9b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se visualiza, busca, ingresa, modifica y/o elimina un administrativo.	

Tabla 11: Caso de uso 3 Gestionar administrativo

CU 04 Gestionar alumnos

Casos de uso: Gestionar alumnos	
Actor: Administrador	
Descripción: El actor puede ver, ingresar, modificar, eliminar a los alumnos	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de administrador	
Escenario principal	
Actor	**Sistema**
El actor ingresa al módulo de alumnos	El sistema redirecciona al módulo de alumnos y devuelve una vista con una lista de todos los alumnos ingresados con sus respectivas opciones.
El actor selecciona el botón + para ingresar un nuevo alumno	El sistema despliega una ventana con los campos a rellenar para el ingreso del nuevo alumno
El actor llena los campos con la información pertinente y hace click en el botón “Guardar”	El sistema ingresa el nuevo alumno a la base de datos y redirecciona al módulo de alumnos con la lista de los alumnos registrados
El actor hace clic en el botón de “Modificar”	El sistema despliega una ventana con los datos registrador del alumno a modificar
El actor modificar los campos que desee modificar y hace click en “Guardar”	El sistema modifica la información que fue cambiada en los campos de la ventana que se desplegó y redirecciona al módulo de alumnos con la lista de los alumnos registrados
El actor hace clic en el botón de “Eliminar”	El sistema despliega una ventana con el nombre del alumno seleccionado y una advertencia de confirmación.

El actor confirma su selección de eliminación y hace click en el botón “Eliminar”.	El sistema elimina la información del alumno de la base de datos y redirecciona al módulo de alumnos con la lista de los alumnos registrados.
Escenario Alternativo	
Actor	**Sistema**
5a) El actor deja un campo en blanco y hace clic en “Guardar”	5b) El sistema informa de la información faltante al actor y que es obligatoria.
9a) El actor deja un campo en blanco y hace clic en “Guardar”	9b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se visualiza, busca, ingresa, modifica y/o elimina un alumno.	

Tabla 12: Caso de uso 4 Gestionar alumno.

CU 05 Gestionar profesores

Casos de uso: Gestionar profesores	
Actor: Administrador	
Descripción: El actor puede ver, ingresar, modificar, eliminar un profesor	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de administrador	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de profesores	2) El sistema redirecciona al módulo de profesores y devuelve una vista con una lista de todos los profesores ingresados con sus respectivas opciones.
3) El actor selecciona el botón + para ingresar un nuevo profesor	4) El sistema despliega una ventana con los campos a rellenar para el ingreso

	del nuevo profesor
5) El actor llena los campos con la información pertinente y hace click en el botón “Guardar”	6) El sistema ingresa el nuevo profesor a la base de datos y redirecciona al módulo de profesores con la lista de los profesores registrados
7) El actor hace clic en el botón de “Modificar”	8) El sistema despliega una ventana con los datos registrador del profesor a modificar
9) El actor modificar los campos que desee modificar y hace click en “Guardar”	10) El sistema modifica la información que fue cambiada en los campos de la ventana que se desplegó y redirecciona al módulo de profesores con la lista de los profesores registrados
11) El actor hace clic en el botón de “Eliminar”	12) El sistema despliega una ventana con el nombre del profesor seleccionado y una advertencia de confirmación.
13) El actor confirma su selección de eliminación y hace click en el botón “Eliminar”.	14) El sistema elimina la información del profesor de la base de datos y redirecciona al módulo de profesores con la lista de los profesores registrados.
Escenario Alternativo	
Actor	**Sistema**
5a) El actor deja un campo en blanco y hace clic en “Guardar”	5b) El sistema informa de la información faltante al actor y que es obligatoria.
9a) El actor deja un campo en blanco y hace clic en “Guardar”	9b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se visualiza, busca, ingresa, modifica y/o elimina un profesor.	

Tabla 13: Caso de uso 5 Gestionar profesor

CU 06 Gestionar asignaturas

Casos de uso: Gestionar asignaturas	
Actor: Administrador - Administrativo	
Descripción: El actor puede ver, ingresar, modificar, eliminar una asignatura de un profesor	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de administrador o administrativo	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de profesores	2) El sistema redirecciona al módulo de profesores y devuelve una vista con una lista de todos los profesores ingresados con sus respectivas opciones.
3) El actor selecciona el botón "Asignaturas" para ver las asignaturas asignadas al profesor	4) El sistema devuelve una vista con las asignaturas asignadas al profesor seleccionado.
5) El actor selecciona el botón + para ingresar una nueva asignatura	6) El sistema despliega una ventana con los campos a rellenar para la asignación de la nueva asignatura
7) El actor llena los campos con la información pertinente y hace click en el botón "Guardar"	8) El sistema ingresa la nueva asignatura del profesor a la base de datos y redirecciona al módulo de las asignaturas del profesor con la lista de las asignaturas asignadas.
9) El actor hace clic en el botón de "Modificar"	10) El sistema despliega una ventana con los datos registrador la asignatura a modificar
11) El actor modificar los campos que desee modificar y hace click en "Guardar"	12) El sistema modifica la información que fue cambiada en los campos de la ventana que se desplegó y redirecciona al módulo de las asignaturas del profesor con la lista

	de las asignaturas asignadas.
13) El actor hace clic en el botón de “Eliminar”	14) El sistema despliega una ventana con el nombre de la asignatura seleccionada y una advertencia de confirmación.
15) El actor confirma su selección de eliminación y hace click en el botón “Eliminar”.	16) El sistema elimina la información de la asignatura del profesor de la base de datos y redirecciona al módulo de las asignaturas del profesor con la lista de las asignaturas asignadas.
Escenario Alternativo	
Actor	**Sistema**
5a) El actor deja un campo en blanco y hace clic en “Guardar”	5b) El sistema informa de la información faltante al actor y que es obligatoria.
9a) El actor deja un campo en blanco y hace clic en “Guardar”	9b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se visualiza, busca, ingresa, modifica y/o elimina una asignatura de un profesor.	

Tabla 14: Caso de uso 6 Gestionar profesor

CU 07 Ver calificaciones

Casos de uso: Ver calificaciones	
Actor: Profesor - Alumno	
Descripción: El actor puede ver una calificación	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor o alumno	
Escenario principal	
Actor	**Sistema**

1) El actor ingresa al módulo de calificaciones	2) El sistema redirecciona al módulo de calificaciones y devuelve una vista con una lista de las asignaturas que tiene.
3) El actor selecciona la asignatura que desee ver.	4) En el caso de ser profesor, el sistema devuelve una lista con todos los alumnos del curso con sus respectivas notas, en caso de ser alumno, devuelve las notas que tiene registradas en dicha asignatura.
Escenario Alternativo	
Actor	**Sistema**
Postcondiciones: Se visualiza una calificación.	

Tabla 15: Caso de uso 7 Ver calificación.

CU 08 Ingresar calificaciones

Casos de uso: Ingresar calificación.	
Actor: Profesor	
Descripción: El actor puede ingresar una calificación	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de calificaciones	2) El sistema redirecciona al módulo de calificaciones y devuelve una vista con una lista de las asignaturas que tiene.
3) El actor selecciona la asignatura que desee ver.	4) El sistema devuelve una lista con los alumnos del curso y sus respectivas notas y promedios parciales.
5) El actor selecciona el botón + para	6) El sistema despliega una ventana con

ingresar una nueva calificación.	los campos a rellenar para el ingreso de la nueva calificación.
7) El actor llena los campos con la información pertinente y hace click en el botón “Guardar”	8) El sistema ingresa la nueva calificación a la base de datos y devuelve la vista de la asignatura seleccionada.
9) El actor modificar los campos que desee modificar y hace click en “Guardar”	10) El sistema modifica la información que fue cambiada en los campos de la ventana que se desplegó y redirecciona al módulo de la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**
7a) El actor deja un campo en blanco y hace clic en “Guardar”	7b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se ingresa una calificación.	

Tabla 16: Caso de uso 8 Ingresar calificación.

CU 09 Modificar calificación.

Casos de uso: Modificar calificación.	
Actor: Profesor.	
Descripción: El actor puede modificar una calificación	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor.	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de calificaciones	2) El sistema redirecciona al módulo de calificaciones y devuelve una vista con una lista de las asignaturas que tiene.

3) El actor selecciona la asignatura que desee ver.	4) En el caso de ser profesor, la vista devuelve una lista con todos los alumnos del curso con sus respectivas notas, en caso de ser alumno, devuelve las notas que tiene registradas en dicha asignatura.
5) El actor hace clic en el botón de "Modificar"	6) El sistema despliega una ventana con los datos registrador de la calificación a modificar
7) El actor modificar los campos que desee modificar y hace click en "Guardar"	8) El sistema modifica la información que fue cambiada en los campos de la ventana que se desplegó y redirecciona al módulo de la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**
7a) El actor deja un campo en blanco y hace clic en "Guardar"	7b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se modifica una calificación.	

Tabla 17: Caso de uso 9 Modificar calificación.

CU 10 Eliminar calificación.

Casos de uso: Eliminar calificación.	
Actor: Profesor.	
Descripción: El actor puede eliminar una calificación	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor.	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de	2) El sistema redirecciona al módulo de

calificaciones	calificaciones y devuelve una vista con una lista de las asignaturas que tiene.
3) El actor selecciona la asignatura que desee ver.	4) En el caso de ser profesor, la vista devuelve una lista con todos los alumnos del curso con sus respectivas notas, en caso de ser alumno, devuelve las notas que tiene registradas en dicha asignatura.
5) El actor hace clic en el botón de "Eliminar"	6) El sistema despliega una ventana con la descripción de la nota y una advertencia de confirmación.
7) El actor confirma su selección de eliminación y hace click en el botón "Eliminar".	8) El sistema elimina la información de la calificación de la base de datos y redirecciona al módulo de la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**
Postcondiciones: Se elimina una calificación.	

Tabla 18: Caso de uso 10 Eliminar calificación.

CU 11 Ver asistencia

Casos de uso: Ver Asistencia	
Actor: Profesor - Alumno	
Descripción: El actor puede ver la asistencia.	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor o alumno	
Escenario principal	
Actor	**Sistema**

1) El actor ingresa al módulo de asistencia	2) El sistema redirecciona al módulo de asistencia y devuelve una vista con una lista de las asignaturas asignadas al profesor.
3) El actor selecciona la asignatura que desee ver.	4) Si el actor es profesor, el sistema devuelve un listado de las clases registradas con la asistencia del curso. Si el actor es alumno, el sistema devuelve un gráfico con el porcentaje de asistencia y un listado de las clases registradas de la asignatura.
Escenario Alternativo	
Actor	**Sistema**
Postcondiciones: Se visualiza la asistencia.	

Tabla 19: Caso de uso 11 Ver asistencia.

CU 12 Ingresar asistencia

Casos de uso: Ingresar Asistencia.	
Actor: Profesor.	
Descripción: El actor puede ingresar la asistencia.	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor.	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de asistencia	2) El sistema redirecciona al módulo de asistencia y devuelve una vista con una lista de las asignaturas asignadas al profesor.
3) El actor selecciona la asignatura que desee ver.	4) El sistema devuelve una vista con las clases registradas de la asignatura.

5) El actor selecciona el botón + para ingresar una nueva calificación.	6) El sistema despliega una ventana con los campos a rellenar para el ingreso de la asistencia.
7) El actor llena los campos con la información pertinente y hace click en el botón "Guardar"	8) El sistema ingresa la nueva asistencia a la base de datos y devuelve la vista de la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**
7a) El actor deja un campo en blanco y hace clic en "Guardar"	7b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se ingresa una asistencia.	

Tabla 20: Caso de uso 12 Ingresar asistencia.

CU 13 Modificar asistencia

Casos de uso: Modificar Asistencia.	
Actor: Profesor.	
Descripción: El actor puede modificar una asistencia.	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor.	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de asistencia	2) El sistema redirecciona al módulo de asistencia y devuelve una vista con una lista de las asignaturas asignadas al profesor.
3) El actor selecciona la asignatura que desee ver.	4) El sistema devuelve una vista con las clases registradas de la asignatura.
5) El actor hace clic en el botón de	6) El sistema despliega una ventana con

"Modificar"	los datos registrador de la asistencia a modificar
7) El actor modificar los campos que desee modificar y hace click en "Guardar"	8) El sistema modifica la información que fue cambiada en los campos de la ventana que se desplegó y redirecciona al módulo de la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**
7a) El actor deja un campo en blanco y hace clic en "Guardar"	7b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se modifica una asistencia.	

Tabla 21: Caso de uso 13 Modificar asistencia

CU 14 Eliminar asistencia

Casos de uso: Eliminar Asistencia.	
Actor: Profesor.	
Descripción: El actor puede modificar una asistencia.	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor.	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de asistencia	2) El sistema redirecciona al módulo de asistencia y devuelve una vista con una lista de las asignaturas asignadas al profesor.
3) El actor selecciona la asignatura que desee ver.	4) El sistema devuelve una vista con las clases registradas de la asignatura.
5) El actor hace click en el botón de	6) El sistema despliega una ventana con

"Eliminar"	la fecha de la asistencia y una advertencia de confirmación.
7) El actor confirma su selección de eliminación y hace click en el botón "Eliminar".	8) El sistema elimina la información de la asistencia de la base de datos y redirecciona al módulo de la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**
Postcondiciones: Se elimina una asistencia.	

Tabla 22: Caso de uso 14 Eliminar asistencia

5.1.2 Diseño

5.1.2.1 Modelo Entidad Relación

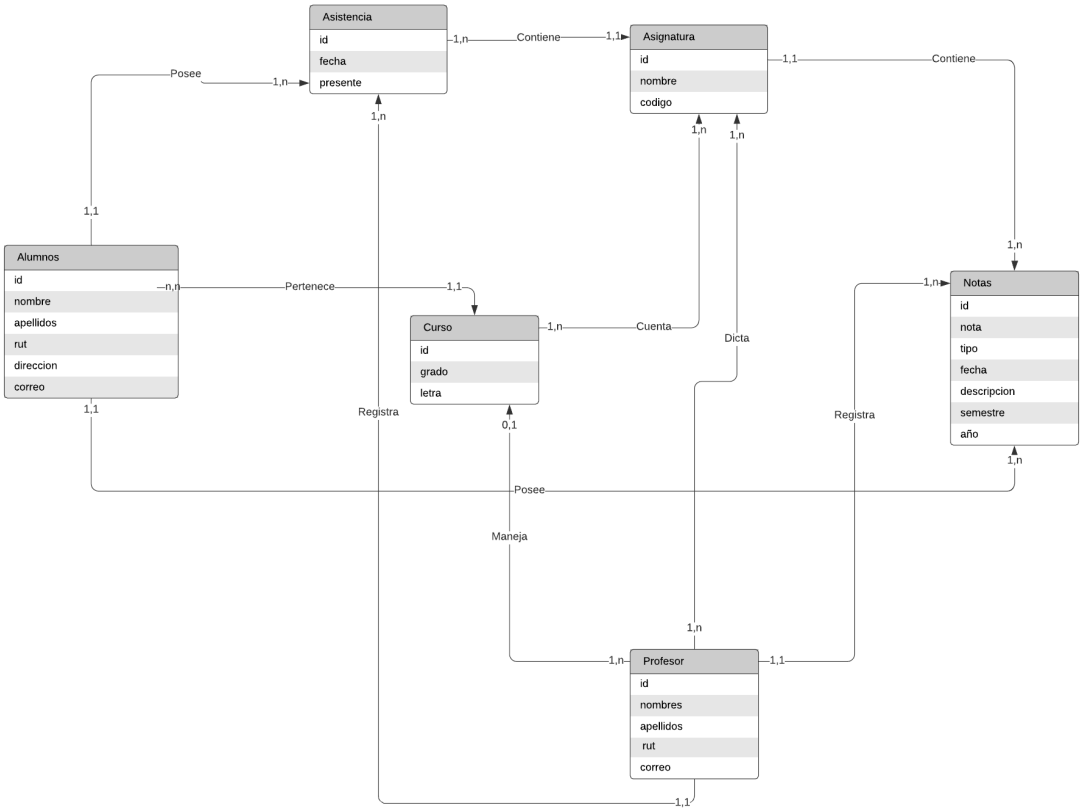

Figura 6: Modelo Entidad relación iteración 1

5.1.2.2 Diseño de la base de datos

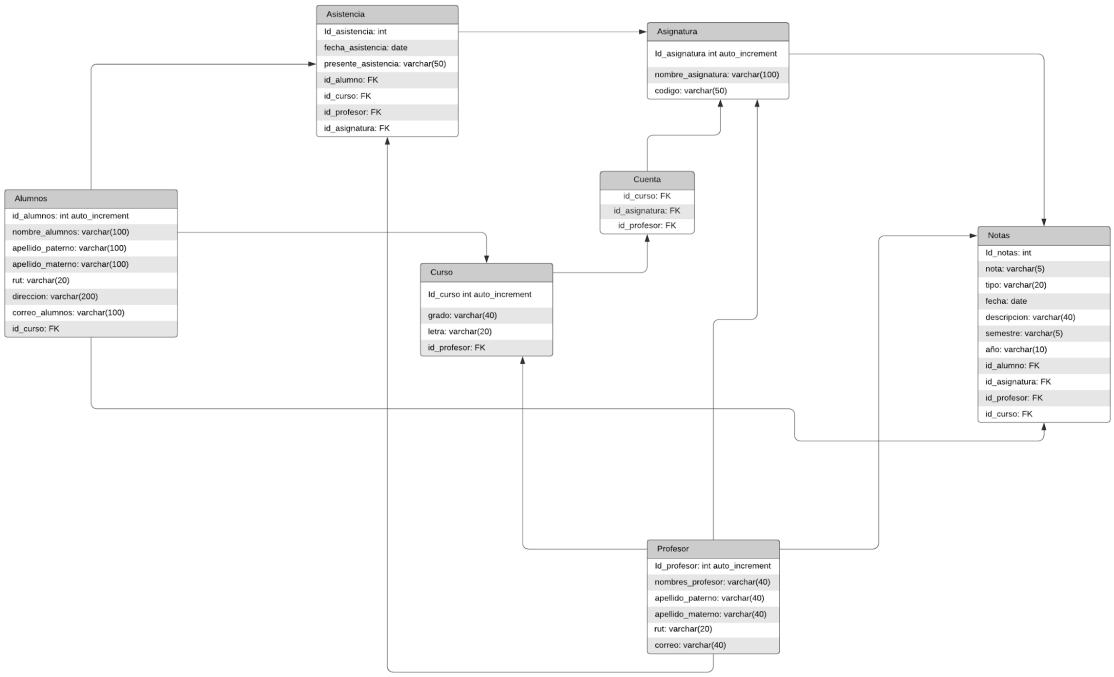

Figura 7: Diseño base de datos iteración 1

5.1.2.3 Diseño interfaz y navegación.

1) Login

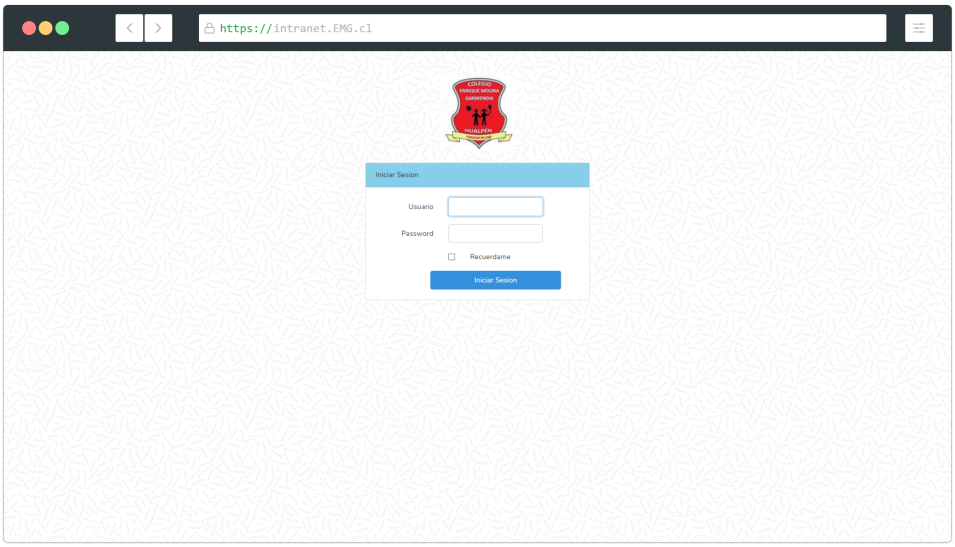

Figura 8: Login

2) Asignaturas profesor

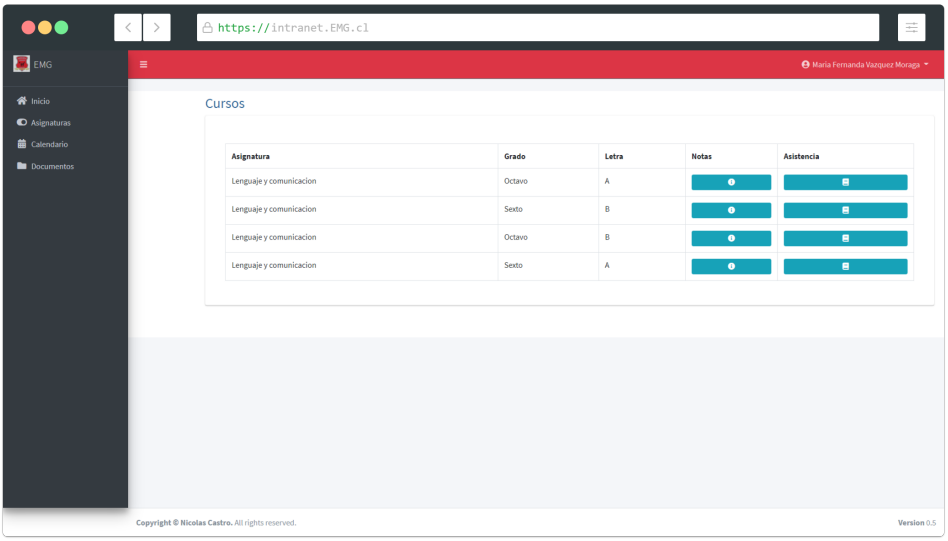

Figura 9: Asignaturas profesor

3) Notas asignaturas profesor

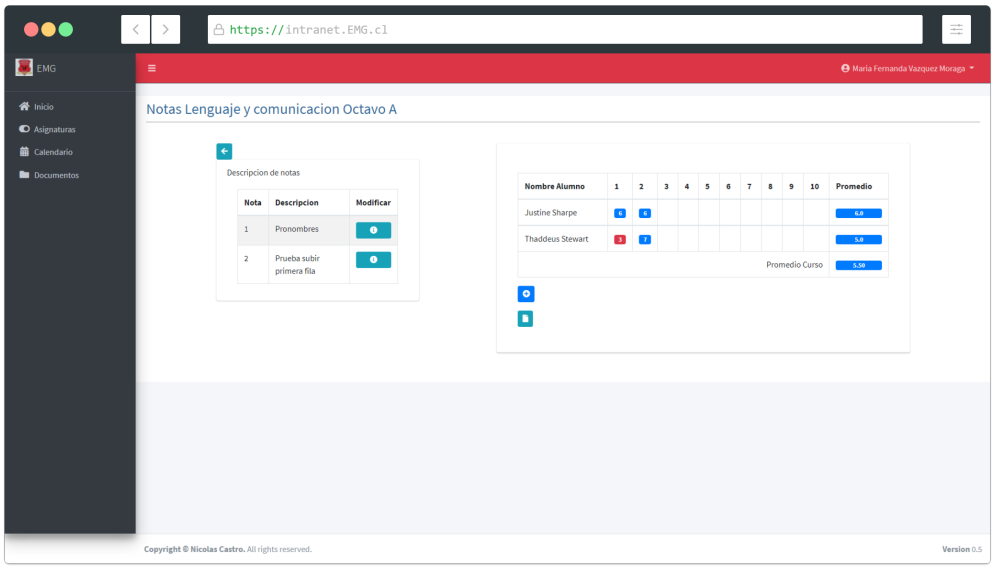

Figura 10: Notas asignaturas profesor

4) Asistencia asignaturas profesor

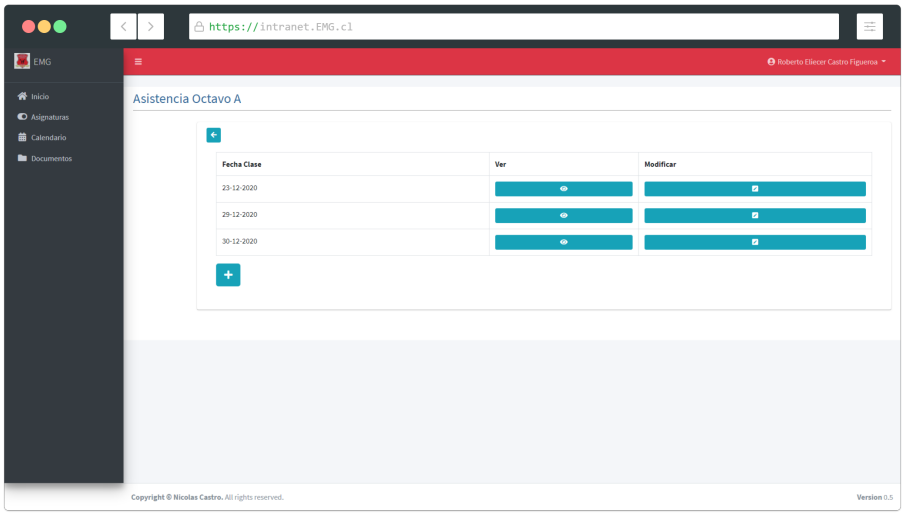

Figura 11: Asistencia asignaturas profesor

5) Asignaturas alumnos

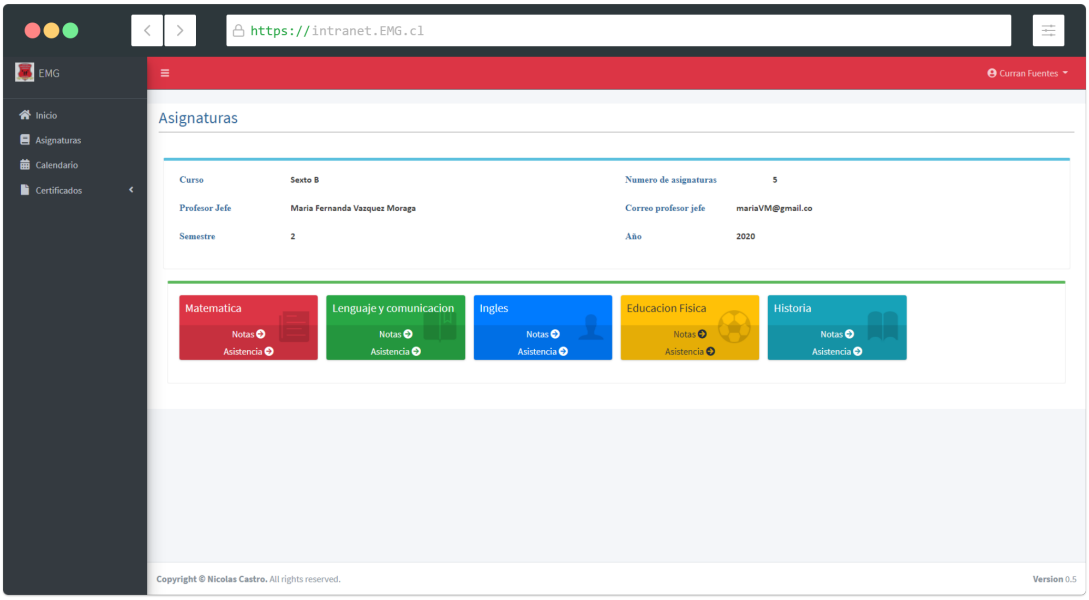

Figura 12: Asignaturas alumnos

5.1.3 Codificación

Conexión a la base de datos

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:1mxwxRSDtDTq/zK5cdXxBx11HW4msl/fM56Lz6xfBbA=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=intranet
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS=null
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_APP_CLUSTER=mt1

MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
```

Figura 13: Conexión a la base de datos.

Ingresar alumno

```
public function create(Request $request){
    $alumno = new Alumnos();
    $alumno->nombre_alumnos = $request->input('nombres');
    $alumno->apellido_paterno = $request->input('apellido_paterno');
    $alumno->apellido_materno = $request->input('apellido_materno');
    $alumno->rut = $request->input('rut');
    $alumno->direccion = $request->input('direccion');
    $alumno->correo_alumnos = $request->input('correo');
    $alumno->id_curso = $request->input('id_curso');
    $alumno->save();

    return Redirect('/alumnos');
}
```

Figura 14: Ingresar alumno

Modificar alumno

```
public function update(Request $request, Alumnos $alumnos){

    $id = $request->input('id');

    $alumnos = Alumnos::find($id);
    $alumnos->nombre_alumnos = $request->input('nombres');
    $alumnos->apellido_paterno = $request->input('apellido_paterno');
    $alumnos->apellido_materno = $request->input('apellido_materno');
    $alumnos->rut = $request->input('rut');
    $alumnos->correo_alumnos = $request->input('correo');
    $alumnos->direccion = $request->input('direccion');
    $alumnos->update();

    return Redirect('/alumnos');

}
```

Figura 15: Modificar alumno

Eliminar alumno

```
public function delete(Request $request, Alumnos $alumnos){
    $id = $request->input('id');

    $alumnos = Alumnos::find($id);
    $alumno->estado = "inactivo";
    $alumnos->update();

    return Redirect('/alumnos');
}
```

Figura 16: Eliminar alumno

Ingresar notas

```
public function create(Request $request){

    $asistencias = DB::table('notas')
                    ->orderBy('id_notas','desc')
                    ->first();

    if($asistencias=""){
        $id_asistencias = "1";
    }else{
        $id_asistencias = ++$asistencias->id_notas;
    }

    echo $id_asistencias;


        $alumnos = DB::table('alumnos')
                    ->where('id_curso','=',$request->input('id_curso'))
                    ->get();

        foreach ($alumnos as $alumno) {
        $nota = new Notas();
        $nota->id_notas = $id_asistencias;
        $nota->nota = $request->input($alumno->id_alumnos);
        $nota->descripcion = $request->input('descripcion');
        if(DATE('m')>='8')
            $nota->semestre = '2';
        else
            $nota->semestre = '1';
        $nota->año = DATE('Y');
        $nota->id_alumno = $alumno->id_alumnos;
        $nota->id_curso = $request->input('id_curso');
        $nota->id_profesor = $request->input('id_profesor');
        $nota->id_asignatura = $request->input('id_asignatura');
        $nota->save();
        }

        return Redirect('/notas/curso');

}
```

Figura 17: Ingresar notas

Modificar notas

```
public function update(Request $request, Notas $nota){
        $id_curso = $request->input('id_curso');
        $id_asignatura = $request->input('id_asignatura');
        $descripcion = $request->input('descripcion');

        $alumnos = DB::table('alumnos')
                    ->get();

        foreach ($alumnos as $alumno) {
            $id_notas = $request->input('id_notas');
            $nota = DB::table('notas')
                    ->where('id_notas','=',$id_notas)
                    ->where('id_alumno','=',$alumno->id_alumnos)
                    ->update(['nota'=> $request->input($alumno->id_alumnos)]);
        }
    return Redirect('/notas/ver');
}
```

Figura 18: Modificar notas.

Eliminar nota

```
public function delete(Request $request){
    $nota = Notas::find($request->input('id'));
    $nota->delete();

    return Redirect('/notas');
}
```

Figura 19: Eliminar nota.

Gestionar asistencia

Ingresar asistencia

```
public function create(Request $request){
    $asistencias = DB::table('asistencia')
                    ->orderBy('id_asistencia','desc')
                    ->first();

    if($asistencias==""){
        $id_asistencias = "1";
    }else{
        $id_asistencias = ++$asistencias->id_asistencia;
    }


    $id_curso = $request->input('id_curso');

    $alumnos = DB::table('alumnos')
                ->where('id_curso','=',$id_curso)
                ->get();
    foreach ($alumnos as $alumno) {
        $asistencia = new Asistencia();
        $asistencia->id_asistencia = $id_asistencias;
        $asistencia->id_alumnos = $alumno->id_alumnos;
        $asistencia->fecha_asistencia = $request->input('fecha');
        if($request->input($alumno->id_alumnos)=='on'){
            $asistencia->presente_asistencia = "Si";
        }else{
            $asistencia->presente_asistencia = "No";
        }

        $asistencia->id_curso = $id_curso;
        $asistencia->id_asignatura = $request->input('id_asignatura');
        $asistencia->id_profesor = $request->input('id_profesor');
        $asistencia->save();
    }

    return Redirect('/asistencia');
}
```

Figura 20: Ingresar asistencia.

Modificar asistencia

```
public function update(Request $request,Asistencia $asistencia){
        $id_asistencia = $request->input('id_asistencia');
        $id_curso = $request->input('id_curso');

        $alumnos = DB::table('alumnos')
                    ->where('id_curso','=',$id_curso)
                    ->get();

        foreach ($alumnos as $alumno) {



            if($request->input($alumno->id_alumnos)=='on'){
            $asistencias = DB::table('asistencia')
                            ->where('id_asistencia','=',$id_asistencia)
                            ->where('id_alumnos','=',$alumno->id_alumnos)
                            ->update(['presente_asistencia' => "Si"]);
            }else{
                $asistencias = DB::table('asistencia')
                            ->where('id_asistencia','=',$id_asistencia)
                            ->where('id_alumnos','=',$alumno->id_alumnos)
                            ->update(['presente_asistencia' => "No"]);

        }
    }
    return Redirect('/asistencia');

}
```

Figura 21: Modificar asistencia

Eliminar asistencia

```
public function delete(Request $request){
    $asistencia = Asistencia::find($request->input('id'));
    $asistencia->delete();

    return Redirect('/asistencia');
}
```

Figura 22: Eliminar asistencia.

5.1.4 Pruebas

5.1.4.1 Elementos de prueba

Los elementos de prueba representan los casos de usos más importantes en esta iteración

Elementos de prueba	**Descripción**
Iniciar sesión	Permite que los usuarios registrados en el sistema puedan acceder a él.
Registrar alumno	Permite al usuario registrar un nuevo alumno en el sistema
Registrar calificación	Permite al usuario registrar una nueva calificación en el sistema
Registrar asistencia	Permite al usuario registrar una asistencia en el sistema

Tabla 23: Elementos de pruebas.

5.1.4.2 Especificación de las pruebas

Características de las pruebas	**Funcionalidad**
Nivel de prueba	Aceptación
Objetivo de la prueba	Verificar que se cumplan los requisitos del sistema, que en casos con entradas erróneas, el sistema se comporte como se debe
Enfoque para la definición de los casos de prueba	Caja negra
Técnicas para la definición de los casos de prueba	Pruebas manuales
Criterio de cumplimiento	Que todos los requerimientos estén 91 cumplidos, y que el sistema se comporte

	según lo esperado.

Tabla 24: Especificación de las pruebas.

5.1.4.3 Calendario de las pruebas

Pruebas	**Rango de fechas**
Iniciar sesión	18/12/2020 - 21/12/2020
Registrar alumno	22/12/2020 - 23/12/2020
Registrar calificación	23/12/2020 - 24/12/2020

Tabla 25: Calendario pruebas.

5.1.4.4 Detalle de las pruebas

- **Elemento de prueba:** Iniciar sesión

	ID	**1**	**2**	**3**
	Característica a probar	**Caso de datos válido**	**Caso de datos invalido**	**Caso de datos obligatorio**
Datos de entrada	Usuario	230530629	23.053.062-9	230530629
	Contraseña	230530629	230530629	
Salida esperada		Sesión iniciada exitosamente, redirecciona a “/home”	Mensaje de error “Usuario no válido”	Mensaje “Contraseña requerida”
Salida obtenida		Sesión iniciada exitosamente, redirecciona a “/home”	Mensaje de error “Usuario no válido”	Mensaje “Contraseña requerida”

Tabla 26: Elemento de prueba iniciar sesión.

- **Elemento de prueba:** Registrar alumno

	ID	**1**	**2**	**3**
	Característica a probar	**Caso de datos válido**	**Caso de datos invalido**	**Caso de datos obligatorio**
Datos de entrada	Nombre	Matias	Matias	Matias
	Apellido paterno	Zapata	Zapata	Zapata
	Apellido materno	Muñoz	Muñoz	Muñoz
	Correo	matiasz@gmail.com	matiasz	matiasz@gmail.com
	Dirección	Dalcahue 320, Diego portales.	Dalcahue 320, Diego portales.	Dalcahue 320, Diego portales.
	Curso	Octavo A	Octavo A	
Salida esperada		Alumno registrado exitosamente	Mensaje de error “Correo incompleto”	Mensaje de error “Ingrese curso”
Salida Obtenida		Alumno registrado exitosamente	Mensaje de error “Correo incompleto”	Mensaje de error “Ingrese curso”

Tabla 27: Elemento de prueba registrar alumno.

- **Elemento de prueba:** Registrar calificación.

	ID	**1**	**2**	**3**
	Característica a probar	**Caso de datos válido**	**Caso de datos invalido**	**Caso de datos obligatorio**
Datos de entrada	Descripción de la nota	Logaritmos	Logaritmos	
	Nota	6,0	62	6,0

Salida esperada	Calificación ingresada exitosamente	Mensaje de error “Formato invalido”	Mensaje de error “Ingrese descripción de la nota”
Salida Obtenida	Calificación ingresada exitosamente	Mensaje de error “Formato invalido”	Mensaje de error “Ingrese descripción de la nota”

Tabla 28: Elemento de prueba registrar calificación.

5.1.4.5 Conclusión de las pruebas

Se realizaron las pruebas con mayor probabilidad a errores por la cantidad de información que se requieren, el sistema cuenta con un validador por el lado del servidor y las pruebas dieron los resultados esperados, por lo que se puede decir que las pruebas han sido un éxito.

5.2 Iteración II

Análisis

En esta iteración se llevarán a cabo los requerimientos funcionales desde el nº 30 al nº 42

5.2.1 Diagrama de flujo de datos

5.2.1.1 Diagrama de contexto

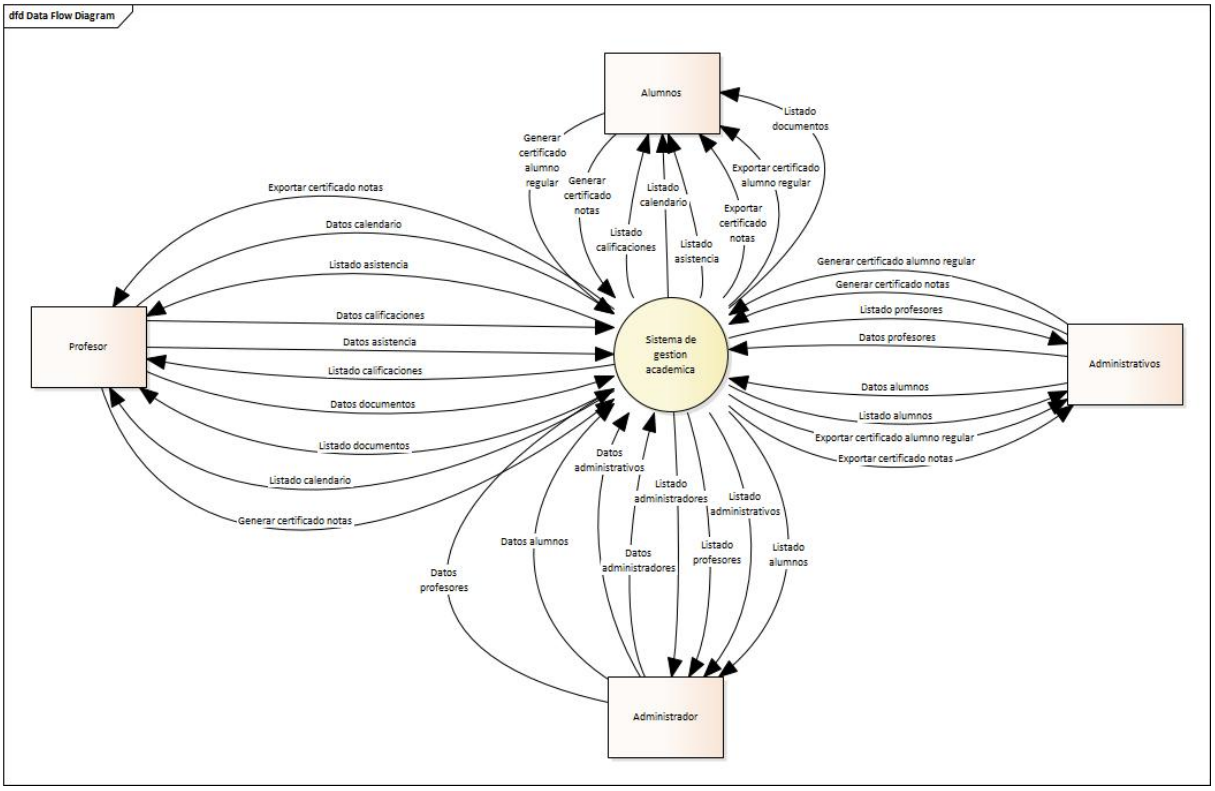

Figura 23: Diagrama de contexto iteración 2.

5.2.1.2 Diagrama de nivel superior

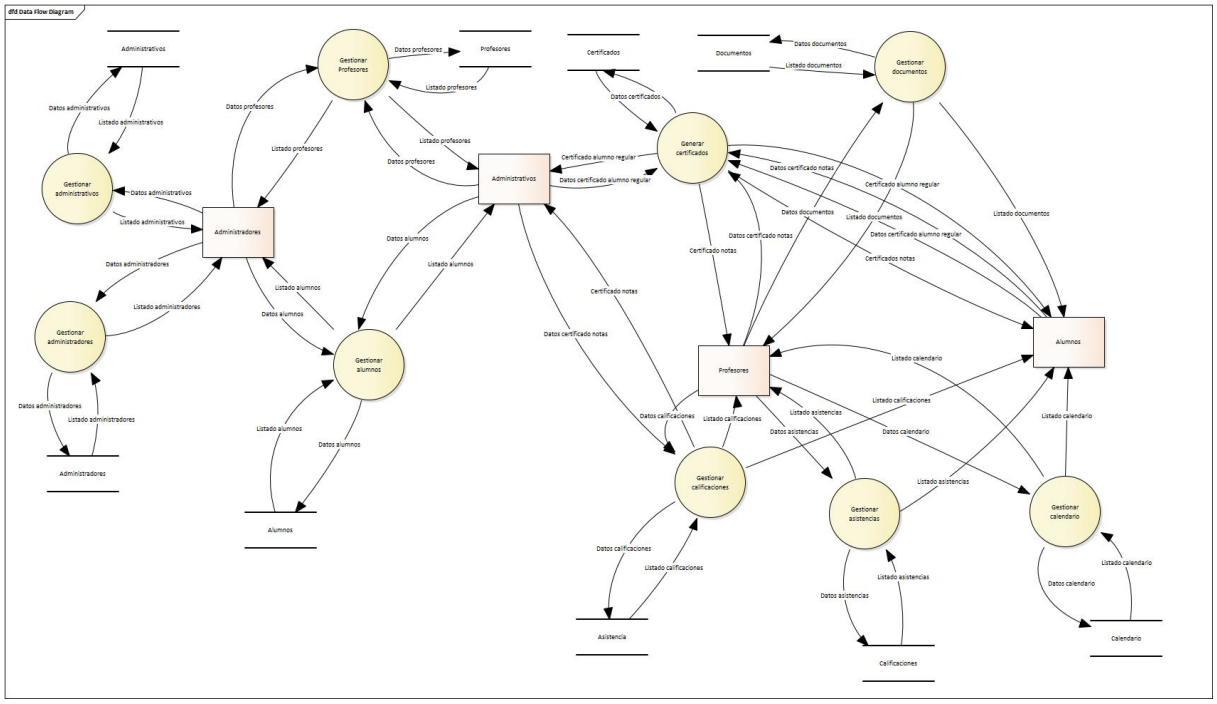

Figura 24: Diagrama de nivel superior iteración 2.

5.2.1.3 Diagrama de casos de uso

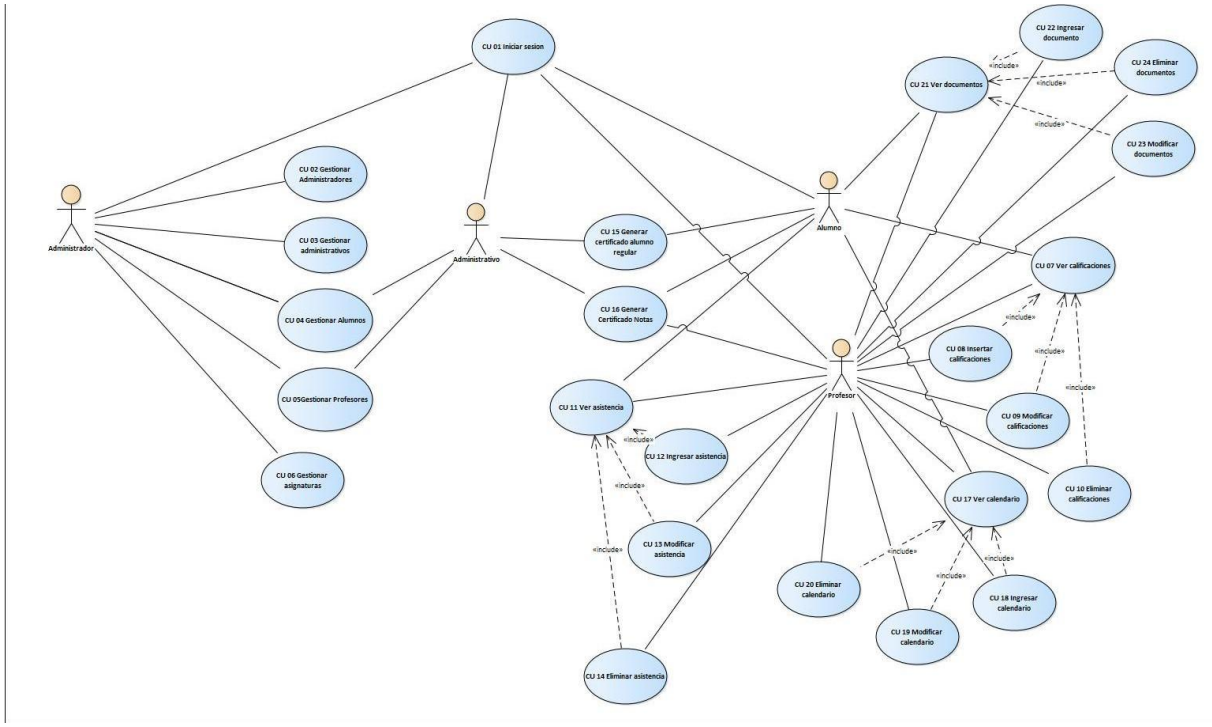

Figura 25: Diagrama de casos de uso iteración 2.

5.2.2 Especificación de casos de uso

CU 15 Generar certificado alumno regular

Casos de uso: Generar certificado alumno regular.	
Actor: Alumno - administrador.	
Descripción: El actor puede generar un certificado de alumno regular.	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de alumno o administrativo.	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de Certificado de alumno regular	2) El sistema redirecciona al módulo de Certificado de alumno regular.
3) El actor hace clic en el botón “Descargar”.	4) El sistema genera el certificado y lo devuelve en formato “PDF” para descargar.
Escenario Alternativo	
Actor	**Sistema**
Postcondiciones: Se exporta un certificado de alumno regular en formato “PDF”.	

Tabla 29: Caso de uso 15 Generar certificado alumno regular.

CU 16 Generar certificado de notas

Casos de uso: Generar certificado de notas.
Actor: Alumno - administrador.
Descripción: El actor puede generar un certificado de notas.
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de alumno.
Escenario principal

Actor	Sistema
1) El actor ingresa al módulo de certificado de notas	2) El sistema redirecciona al módulo de certificado de notas y el devuelve una vista con los datos del alumno.
3) El actor selecciona el semestre del cual desea generar el certificado de notas y luego hace click en "Descargar".	4) El sistema genera el certificado con las notas del semestre seleccionado y exporta el certificado en formato "PDF" para poder descargarlo.
Escenario Alternativo	
Actor	**Sistema**
3a) El actor no selecciona el semestre que desee generar el certificado y hace click en "Descargar"	3b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se exporta un certificado de notas en formato "PDF".	

Tabla 30: Caso de uso 16 Generar certificado notas.

CU 17 Ver calendario

Casos de uso: Ver calendario	
Actor: Profesor - Alumno	
Descripción: El actor puede ver el calendario	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor o alumno	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de calendario	2) En el caso de ser profesor, el sistema devuelve una lista con las asignaturas asignadas, en caso de ser alumno, devuelve el calendario con todos los eventos futuros registrados en el.
3) El actor selecciona la asignatura que	4) El sistema devuelve una lista de los

desee ver.	eventos futuros registrados en la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**
Postcondiciones: Se visualiza el calendario.	

Tabla 31: Caso de uso 17 Ver calendario.

CU 18 Ingresar calendario

Casos de uso: Ingresar calendario.	
Actor: Profesor.	
Descripción: El actor puede ingresar un evento futuro al calendario.	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor.	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de calendario	2) El sistema redirecciona al módulo de calendario y devuelve una vista con una lista de las asignaturas asignadas al profesor.
3) El actor selecciona la asignatura que desee ver.	4) El sistema devuelve una vista con los eventos futuros de la asignatura seleccionada.
5) El actor selecciona el botón + para ingresar un nuevo evento al calendario.	6) El sistema despliega una ventana con los campos a rellenar para el ingreso del evento al calendario.
7) El actor llena los campos con la información pertinente y hace click en el botón “Crear”	8) El sistema ingresa el nuevo evento a la base de datos y devuelve la vista de la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**

7a) El actor deja un campo en blanco y hace clic en “Crear”	7b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se ingresa un evento al calendario.	

Tabla 32: Caso de uso 18 Ingresar calendario.

CU 19 Modificar calendario

Casos de uso: Modificar calendario.	
Actor: Profesor.	
Descripción: El actor puede modificar un evento del calendario.	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor.	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de asistencia	2) El sistema redirecciona al módulo de calendario y devuelve una vista con una lista de las asignaturas asignadas al profesor.
3) El actor selecciona la asignatura que desee ver.	4) El sistema devuelve una vista con los eventos futuros de la asignatura seleccionada.
5) El actor hace clic en el botón de “Modificar”	6) El sistema despliega una ventana con los datos registrador del evento a modificar
7) El actor modificar los campos que desee modificar y hace click en “Guardar”	8) El sistema modifica la información que fue cambiada en los campos de la ventana que se desplegó y redirecciona al módulo de la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**

7a) El actor deja un campo en blanco y hace clic en “Guardar”	7b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se modifica un evento del calendario.	

Tabla 33: Caso de uso 19 Modificar calendario.

CU 20 Eliminar calendario

Actor: Profesor.	
Descripción: El actor puede Eliminar un evento del calendario.	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor.	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de calendario	2) El sistema redirecciona al módulo de calendario y devuelve una vista con una lista de las asignaturas asignadas al profesor.
3) El actor selecciona la asignatura que desee ver.	4) El sistema devuelve una vista con los eventos futuros de la asignatura seleccionada.
5) El actor hace click en el botón de “Eliminar”	6) El sistema despliega una ventana con la descripción del evento y una advertencia de confirmación.
7) El actor confirma su selección de eliminación y hace click en el botón “Eliminar”.	8) El sistema elimina la información del evento de la base de datos y redirecciona al módulo de la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**
Postcondiciones: Se elimina un evento del calendario.	

Tabla 34: Caso de uso 20 Eliminar calendario.

CU 21 Ver documentos

Casos de uso: Ver Documentos	
Actor: Profesor - Alumno	
Descripción: El actor puede ver los documentos de las asignaturas	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor o alumno	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de calendario	2) El sistema devuelve una lista de las asignaturas asignadas del actor
3) El actor selecciona la asignatura que desee ver.	4) El sistema devuelve una lista con los documentos que están ingresados en la asignatura.
Escenario Alternativo	
Actor	**Sistema**
Postcondiciones: Se visualiza los documentos de la asignatura.	

Tabla 35: Caso de uso 21 Ver documentos.

CU 22 Ingresar documentos

Casos de uso: Ingresar documento.
Actor: Profesor.
Descripción: El actor puede ingresar un documento a una asignatura.
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor.
Escenario principal

Actor	Sistema
1) El actor ingresa al módulo de documentos	2) El sistema redirecciona al módulo de calendario y devuelve una vista con una lista de las asignaturas asignadas al profesor.
3) El actor selecciona la asignatura que desee ver.	4) El sistema devuelve una lista con los documentos que están ingresados en la asignatura.
5) El actor selecciona el botón + para ingresar un nuevo documento a la asignatura	6) El sistema despliega una ventana con los campos a rellenar para el ingreso del documento.
7) El actor llena los campos con la información pertinente y hace click en el botón "Crear"	8) El sistema ingresa el nuevo documento y devuelve la vista de la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**
7a) El actor deja un campo en blanco y hace clic en "Crear"	7b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se ingresa un documento a una asignatura.	

Tabla 36: Caso de uso 22 Ingresar documento.

CU 23 Modificar documento

Casos de uso: Modificar documento.	
Actor: Profesor.	
Descripción: El actor puede modificar un documento de una asignatura.	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor.	
Escenario principal	
Actor	**Sistema**

1) El actor ingresa al módulo de documentos	2) El sistema redirecciona al módulo de documentos y devuelve una vista con una lista de las asignaturas asignadas al profesor.
3) El actor selecciona la asignatura que desee ver.	4) El sistema devuelve una lista con los documentos que están ingresados en la asignatura.
5) El actor hace clic en el botón de "Modificar"	6) El sistema despliega una ventana con los datos registrador del documento a modificar
7) El actor modificar los campos que desee modificar y hace click en "Guardar"	8) El sistema modifica la información que fue cambiada en los campos de la ventana que se desplegó y redirecciona al módulo de la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**
7a) El actor deja un campo en blanco y hace clic en "Guardar"	7b) El sistema informa de la información faltante al actor y que es obligatoria.
Postcondiciones: Se modifica un documento de una asignatura.	

Tabla 37: Caso de uso 23 Modificar documento.

CU 24 Eliminar documento

Actor: Profesor.	
Descripción: El actor puede eliminar un documento de una asignatura.	
Precondiciones: El usuario debe estar autenticado en el sistema con el rol de profesor.	
Escenario principal	
Actor	**Sistema**
1) El actor ingresa al módulo de	2) El sistema redirecciona al módulo de

documentos	documentos y devuelve una vista con una lista de las asignaturas asignadas al profesor.
3) El actor selecciona la asignatura que desee ver.	4) El sistema devuelve una lista con los documentos que están ingresados en la asignatura.
5) El actor hace click en el botón de “Eliminar”	6) El sistema despliega una ventana con el nombre del documento y una advertencia de confirmación.
7) El actor confirma su selección de eliminación y hace click en el botón “Eliminar”.	8) El sistema elimina el documento de la base de datos y redirecciona al módulo de la asignatura seleccionada.
Escenario Alternativo	
Actor	**Sistema**
Postcondiciones: Se elimina un documento de una asignatura.	

Tabla 38: Caso de uso 24 Eliminar documento.

5.2.3 Diseño

5.2.3.1 Modelo entidad relación

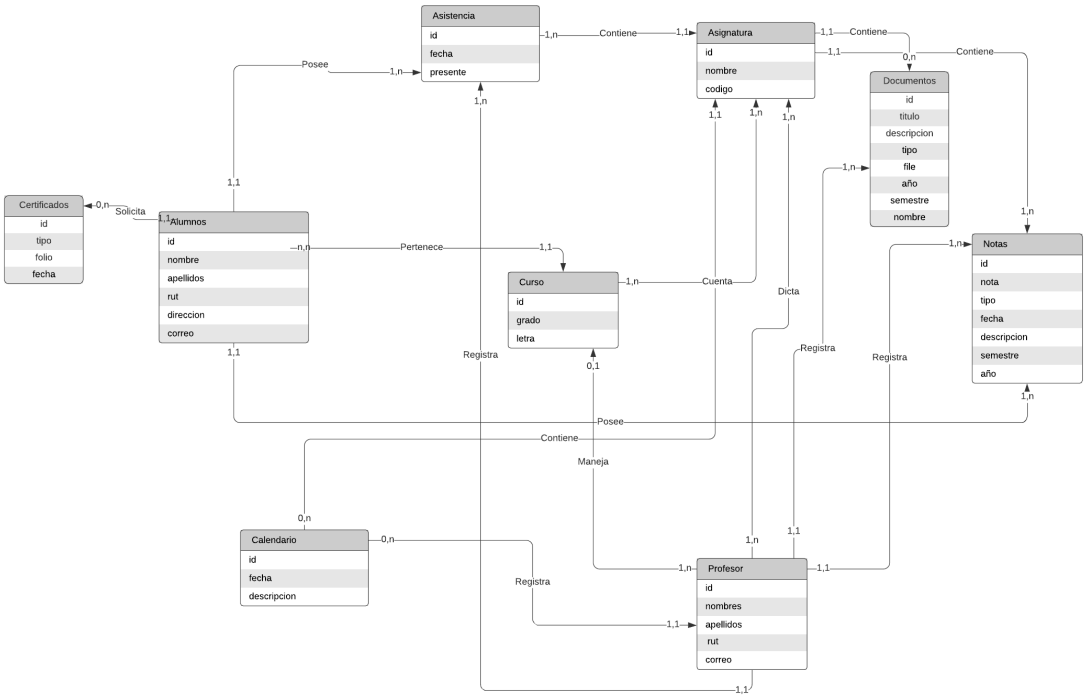

Figura 26: Modelo entidad relación iteración 2.

5.2.3.2 Diseño de la base de datos

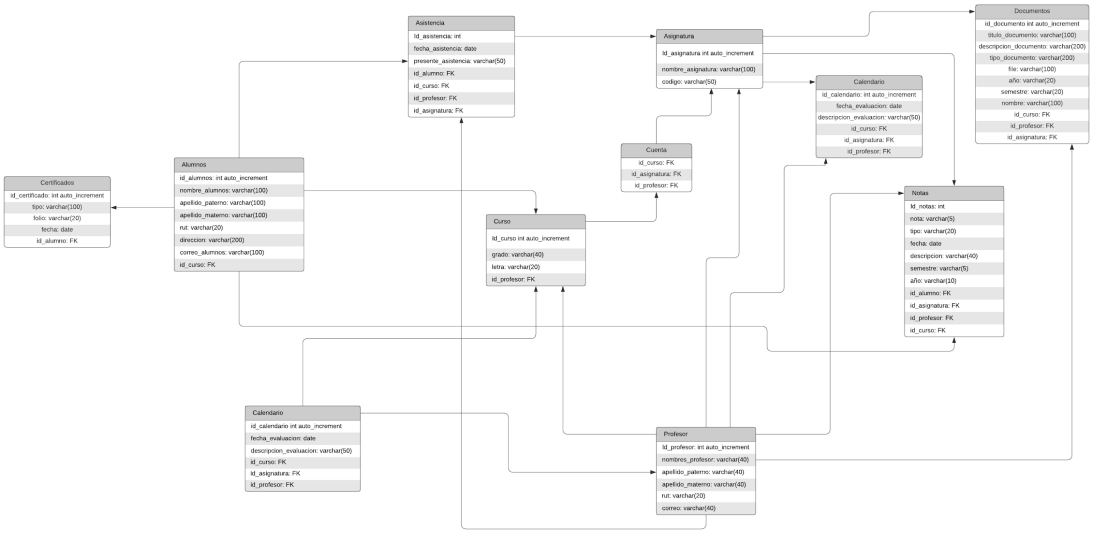

Figura 27: Diseño de la base de datos iteración 2.

5.2.3.3 Diseño de interfaz y navegación

1) Calendario

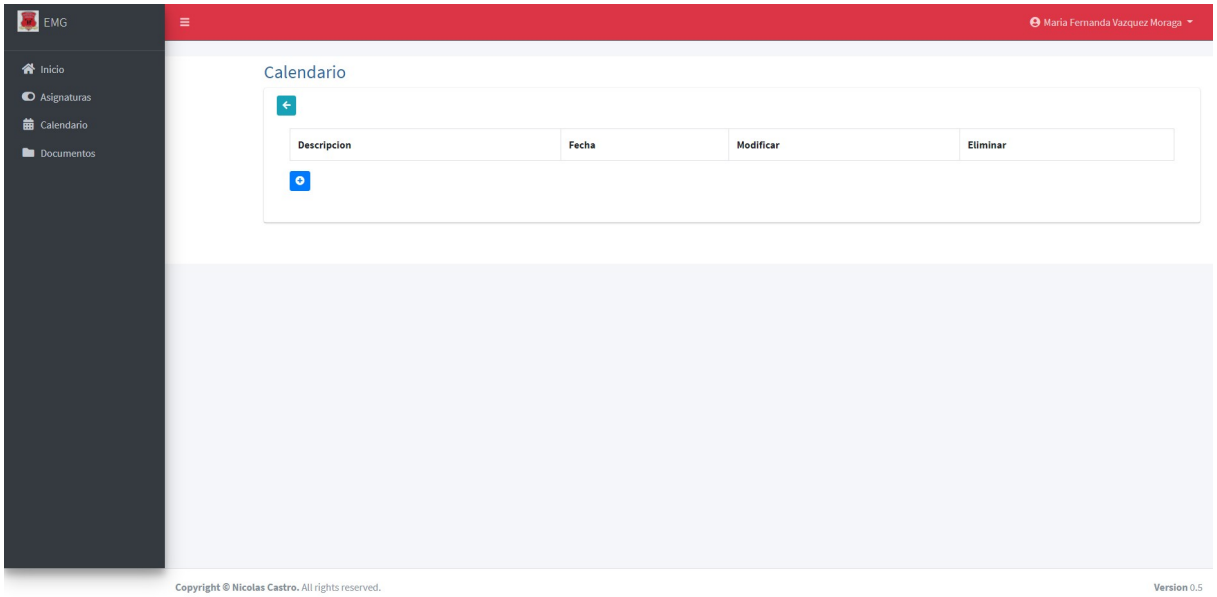

Figura 28: Calendario.

2) Documentos

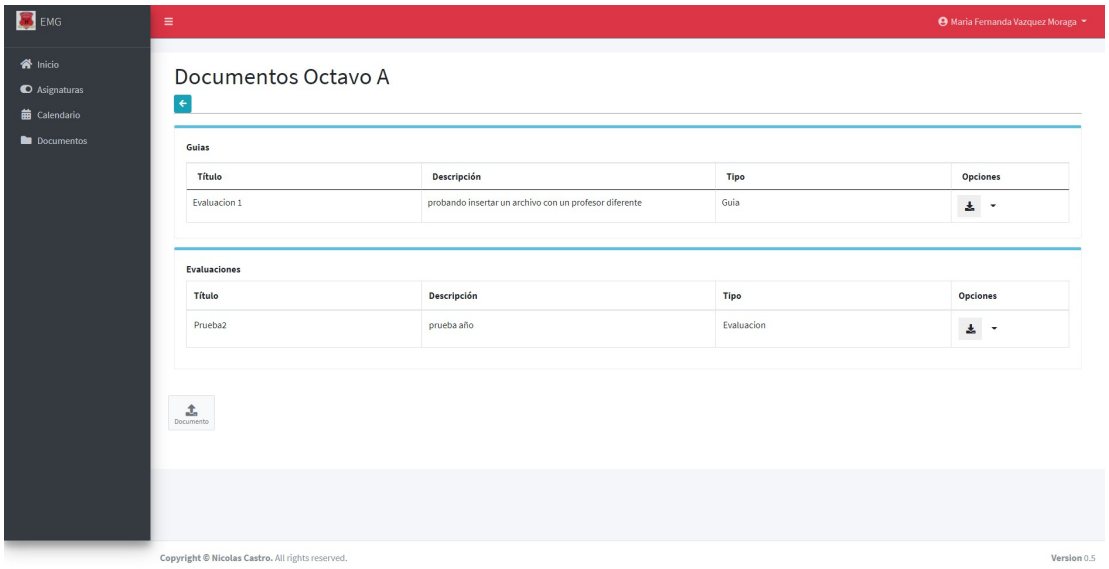

Figura 29: Calendario.

3) Certificado alumno regular

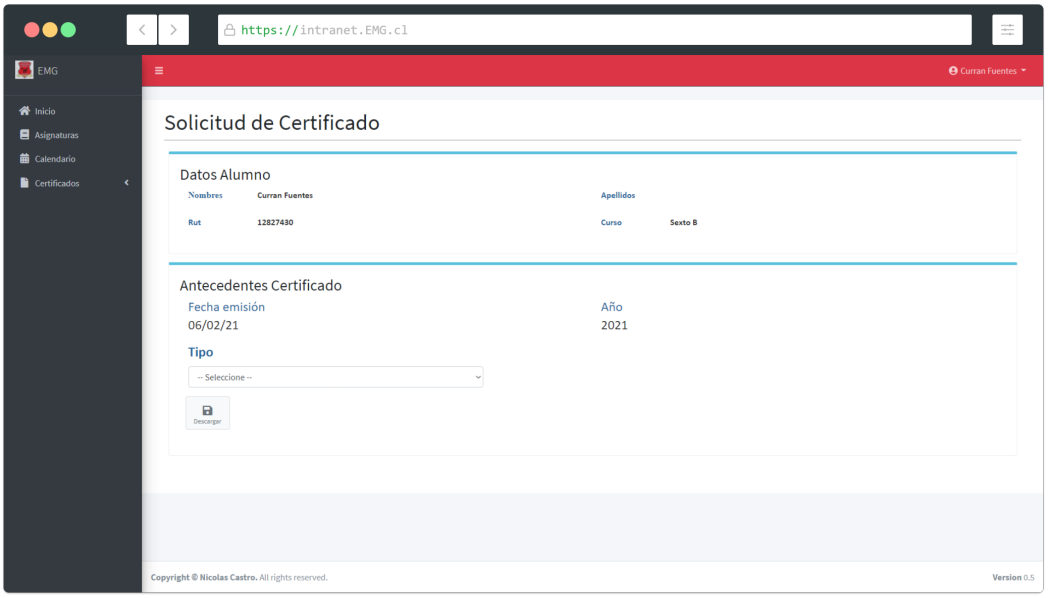

Figura 30: Certificado alumno regular.

4) Certificado notas alumno

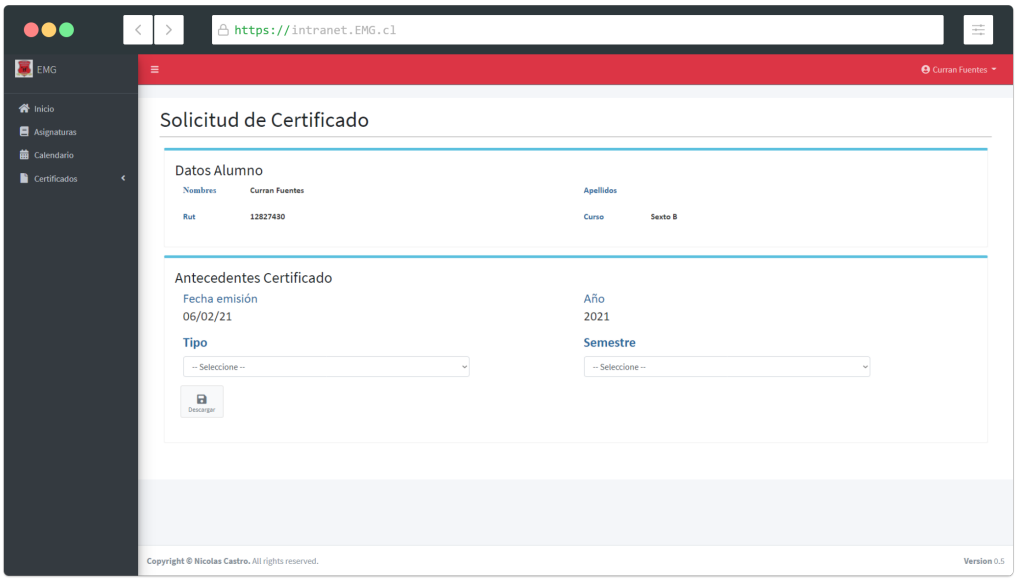

Figura 31: Certificado notas alumno.

5.2.4 Codificación

Ingresar calendario

```
public function create(Request $request){

    $calendario = new Calendario();
    $calendario->fecha_evaluacion = $request->input('fecha');
    $calendario->descripcion_evaluacion = $request->input('descripcion_evaluacion');;
    $calendario->id_curso = $request->input('id_curso');
    $calendario->id_asignatura = $request->input('id_asignatura');
    $calendario->id_profesor = $request->input('id_profesor');
    $calendario->save();

    return Redirect('/calendario');

}
```

Figura 32: Ingresar calendario.

Modificar calendario

```
public function update(Request $request, Calendario $calendario){

    $calendario = Calendario::find($request->input('id_calendario'));
    $calendario->fecha_evaluacion = $request->input('fecha_evaluacion');
    $calendario->descripcion_evaluacion = $request->input('descripcion_evaluacion');
    $calendario->update();

    return Redirect('/calendario');
}
```

Figura 33: Modificar calendario.

Eliminar calendario

```
public function delete(Request $request, Calendario $calendario){
    $calendario = Calendario::find($request->input('id_calendario'));
    $calendario->delete();

    return Redirect('/calendario');
}
```

Figura 34: Eliminar calendario.

Ingresar documento

```
public function store(Request $request){

    $documento = new Documentos;

    if ($request->file('file')) {
        $ruta = auth()->user()->rut;
        $ruta = str_replace('-','', $ruta);
        $ruta = str_replace('.','', $ruta);
        $file = $request->file('file');
        $nombre = $file->getClientOriginalName();
        $filename = time().'.'.$file->getClientOriginalName();
        $request->file->move(public_path().'/documentos/'.$ruta, $filename);
        $documento->nombre = $nombre;
        $documento->file = $filename;
    }

    $documento->titulo_documento = $request->input('titulo');
    $documento->descripcion_documento = $request->input('descripcion');

    $documento->id_curso = $request->input('id_curso');
    $documento->id_profesor = $request->input('id_profesor');
    $documento->id_asignatura = $request->input('id_asignatura');
    $documento->tipo_documento = $request->input('tipo');
    if(date('m')<3){
        $documento->año = date('Y')-1;
        $documento->semestre = 2;
    }else{
        if(date('m')<=8){
            $documento->año = date('Y');
            $documento->semestre = 1;
        }else{
            $documento->año = date('Y');
            $documento->semestre = 2;
        }
    }

    $documento->save();

    return Redirect('/material');
}
```

Figura 35: Ingresar documento.

Modificar documento

```
public function update(Request $request){

    $documento = Documentos::find($request->input('id'));

    if ($request->file('file')) {
        $ruta = auth()->user()->rut;
        $ruta = str_replace('-','', $ruta);
        $ruta = str_replace('.','', $ruta);
        $file = $request->file('file');
        $nombre = $file->getClientOriginalName();
        $filename = time().'.'.$file->getClientOriginalName();
        $request->file->move(public_path().'/documentos/'.$ruta, $filename);
        $documento->nombre = $nombre;
        $documento->file = $filename;
    }

    $documento->titulo_documento = $request->input('titulo');
    $documento->descripcion_documento = $request->input('descripcion');

    $documento->id_curso = $request->input('id_curso');
    $documento->id_profesor = $request->input('id_profesor');
    $documento->id_asignatura = $request->input('id_asignatura');
    $documento->tipo_documento = $request->input('tipo');
    if(date('m')<3){
        $documento->año = date('Y')-1;
        $documento->semestre = 2;
    }else{
        if(date('m')<=8){
            $documento->año = date('Y');
            $documento->semestre = 1;
        }else{
            $documento->año = date('Y');
            $documento->semestre = 2;
        }
    }

    $documento->update();

    return Redirect('/material');
}
```

Figura 36: Modificar documento.

Eliminar documento

```
public function delete(Request $request){
        $documento = Documentos::find($request->input('id'));

        $documento->delete();

        return Redirect('/material');
}
```

Figura 37: Eliminar documento.

Certificado alumno regular

```
public function index(){
    $alumno = Alumnos::all()->where('rut',auth()->user()->rut);
    return view('certificados.sacar_certificadoalumnoregular',compact('alumno'));
}
```

```
public function alumnoregular(Request $request){
    $alumno = Alumnos::all()->where('rut',auth()->user()->rut);
    foreach ($alumno as $a ) {
    $curso = Curso::all()->where('id_curso',$a->id_curso);
    }
    $pdf = PDF::loadview('certificados.certificado_alumnoregular',compact('alumno','curso'));
    return $pdf->stream('regular.pdf');
}
```

Figura 38: Certificado alumno regular.

Certificado notas

```
public function indexnotasa(){
    $alumno = Alumnos::all()->where('rut',auth()->user()->rut);
    return view('certificados.sacar_certificadonotas_alumno',compact('alumno'));
}
```

```
public function notasalumno(Request $request){
    $semestre=$request->input('semestre');
    $alumno = Alumnos::where('rut',auth()->user()->rut)->first();
    $curso = Curso::where('id_curso',$alumno->id_curso)->first();
    $asignatura = DB::table('cuenta')
                ->where('id_curso','=',$alumno->id_curso)
                ->get();



    $pdf = PDF::loadview('certificados.certificado_notaalumno',compact('alumno','curso','asignatura','semestre'));
    return $pdf->stream();

}
```

Figura 39: Certificado notas.

5.2.5 Pruebas

5.2.5.1 Elementos de prueba

Los elementos de prueba representan los casos de usos más importantes en esta iteración

Elementos de prueba	**Descripción**
Ingresar calendario	Permite al usuario ingresar un evento al calendario
Ingresar documento	Permite al usuario ingresar un documento al sistema
Generar certificado alumno regular	Permite al usuario generar un certificado de alumno regular
Generar certificado notas	Permite al usuario generar un certificado de notas parciales

Tabla 39: Elementos de prueba iteración 2.

5.2.5.2 Especificación de las pruebas

Características de las pruebas	**Funcionalidad**
Nivel de prueba	Aceptación
Objetivo de la prueba	Verificar que se cumplan los requisitos del sistema, que en casos con entradas erróneas, el sistema se comporte como se debe
Enfoque para la definición de los casos de prueba	Caja negra
Técnicas para la definición de los casos de prueba	Pruebas manuales
Criterio de cumplimiento	Que todos los requerimientos estén 91

	cumplidos, y que el sistema se comporte según lo esperado.

Tabla 40: Especificación de las pruebas iteración 2.

5.2.5.3 Calendario de las pruebas

Pruebas	**Rango de fechas**
Ingresar calendario	01/02/2021 - 01/02/2021
Ingresar documento	02/02/2021 - 02/02/2021
Generar certificado alumno regular	03/02/2021 - 03/02/2021
Generar certificado notas	04/02/2021 - 04/02/2021

Tabla 41: Calendario de pruebas iteración 2.

5.2.5.4 Detalle de las pruebas

- **Elemento de prueba:** Ingresar calendario

	ID	**1**	**2**	**3**
	Característica a probar	**Caso de datos válido**	**Caso de datos invalido**	**Caso de datos obligatorio**
Datos de entrada	Fecha	20/07/2020	20/07/2020	
	Descripción	Reunión de apoderados	123456789	Reunión de apoderados
Salida esperada		Evento ingresado exitosamente	Mensaje de error “Ingrese una descripción válida”	Mensaje “Fecha requerida”
Salida obtenida		Evento ingresado exitosamente	Mensaje de error “Ingrese una descripción válida”	Mensaje “Fecha requerida”

Tabla 42: Elemento de prueba Ingresar calendario.

- **Elemento de prueba:** Ingresar documento

	ID	**1**	**2**	**3**
	Característica a probar	**Caso de datos válido**	**Caso de datos invalido**	**Caso de datos obligatorio**
Datos de entrada	Título	Algebra	Alge-bra	Algebra
	Descripción	Guia de problemas algebraicos	Guia de problemas algebraicos	Guia de problemas algebraicos
	Tipo documento	Guia	Guia	Guia
	Archivo	guia_a.pdf	guia_a.pdf	
Salida esperada		Documento ingresado exitosamente	Mensaje de error “Ingrese un título válido ”	Mensaje de error “Ingrese un archivo”
Salida Obtenida		Documento ingresado exitosamente	Mensaje de error “Ingrese un título válido”	Mensaje de error “Ingrese un archivo”

Tabla 43: Elemento de prueba Ingresar documento.

- **Elemento de prueba:** Generar certificado alumno regular.

	ID	**1**	**2**
	Característica a probar	**Caso de datos válido**	**Caso de datos obligatorio**
Datos de entrada	Tipo	Alumno regular	
Salida esperada		Certificado generado exitosamente	Mensaje de error “Seleccione tipo de certificado”
Salida Obtenida		Certificado generado exitosamente	Mensaje de error “Seleccione tipo de

		certificado”

Tabla 44: Elemento de prueba Generar certificado alumno regular.

- **Elemento de prueba:** Generar certificado notas

	ID	**1**	**2**
	Característica a probar	**Caso de datos válido**	**Caso de datos obligatorio**
Datos de entrada	Tipo	Concentración de notas	Concentración de notas
	Semestre	Primer	
Salida esperada		Certificado generado exitosamente	Mensaje de error “Seleccione semestre”
Salida Obtenida		Certificado generado exitosamente	Mensaje de error “Seleccione semestre”

Tabla 45: Elemento de prueba Certificado notas.

5.2.5.5 Conclusión de las pruebas

Se realizaron las pruebas con mayor probabilidad a errores por la cantidad de información que se requieren, el sistema cuenta con un validador por el lado del servidor y las pruebas dieron los resultados esperados, por lo que se puede decir que las pruebas han sido un éxito.

Con estos resultados se puede concluir que el sistema se encuentra apto para su utilización por parte de los alumnos, profesores, administrativos y administradores del establecimiento.

CAPÍTULO VI: PLAN DE CAPACITACIÓN Y PUESTA EN MARCHA

1) Plan de capacitación y entrenamiento.

La capacitación tiene como objetivo enseñar de forma práctica cómo se debe utilizar de forma correcta el sistema. Se asume que no todos los usuarios están familiarizados con las nuevas tecnologías por lo que se tomó en cuenta al momento de diseñar de la manera más intuitiva posible el sistema.

Los usuarios a capacitar son los siguientes:

- Administradores: Son aquellos que tienen el permiso de poder gestionar a los alumnos, profesores, administrativos y administradores.
- Administrativos: Son aquellos que tienen el permiso de poder gestionar a los alumno y profesores, además de ser capaces de generar certificados de alumno regular y de notas de los alumnos.
- Profesores: Son aquellos que tienen el permiso de poder gestionar las calificaciones y asistencias de los alumnos, además puede gestionar los documentos que se suben a la plataforma y el calendario, para que los alumnos puedan visualizarlos.
- Alumnos: Son aquellos que pueden ver sus calificaciones, asistencia a clases, ver el material pedagógico que los profesores suben a la plataforma, ver el calendario y también poder generar certificados de alumno regular y de calificaciones.

Las capacitaciones se realizarán de manera telepresencial debido a la pandemia que el país está atravesando en estos momentos. Las capacitaciones se realizarán en 3 jornadas con su respectivo tiempo estimado:

- Jornada 1: Administradores y administrativos 1-3 días.
- Jornada 2: Profesores 1-5 días.
- Jornada 3: Alumnos. 1-2 días.

2) Plan de implementación y puesta en marcha.

Para la puesta en marcha, primeramente, se debe realizar la instalación del sistema en el hosting contratado por el establecimiento, en el cual se instalará la plataforma web y la base de datos, la cual al momento de estar localizada en el hosting, se completará con información verídica del establecimiento, quiere decir, información de los alumnos, profesores, administradores, administrativos, asignaturas, etc.

Ya instalado el sistema, se realizará una puesta en marcha en paralelo, lo que tendrá una duración de 1 semestre máximo, esto para que los usuarios se puedan familiarizar con la plataforma y el cambio abrupto de un sistema a otro no genere errores que puedan ser perjudiciales para el establecimiento, como por ejemplo, la desaparición de las notas de los alumnos.

Se realizaran respaldos de la información ingresada a la base de datos cada 15 días para mayor seguridad.

Se realizaran mantenciones preventivas cada trimestre en el primer año de funcionamiento del sistema, pasado ese tiempo se realizaran mantenciones cada semestre, en caso de presentarse fallas a nivel de programación, se brindara soporte de manera gratuita.

En caso de alguna falla grave de la plataforma, el desarrollador brindará la asistencia necesaria.

3) Resumen esfuerzo requerido.

8.1 Calcular UAW (Unadjusted actor weights)

Tipo de actor	Descripción	Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación de aplicaciones (API).	1
Medio	Otro sistema interactuando a través de un protocolo (ej. TCP/IP) o una persona interactuando a través de una interfaz en modo texto.	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica (GUI).	3

Tabla 46: Calcular UAW.

Actor	Tipo de actor	Peso
Administrador	Complejo	3
Administrativo	Complejo	3
Profesor	Complejo	3
Alumnos	Complejo	3
UAW = 3 + 3 + 3 + 3		**UAW** = 12

Tabla 47: Cálculo UAW.

8.2 Calcular UUCW (Unadjusted use case weights)

Tipo de caso de uso	Descripción	Peso
Simple	Menos de 5 clases	5
Medio	5 a 10 clases	10
Complejo	Más de 10 clases	15

Tabla 48: Calcular UUCW.

Caso de uso	Tipo de caso de uso	Peso
Iniciar sesión	Simple	5
Gestionar administradores	Medio	10
Gestionar administrativos	Medio	10
Gestionar alumnos	Medio	10
Gestionar profesores	Medio	10
Gestionar asignaturas	Medio	10
Ver calificaciones	Simple	5
Insertar calificaciones	Simple	5
Modificar calificaciones	Simple	5
Eliminar calificaciones	Simple	5
Ver Asistencia	Simple	5
Insertar asistencia	Simple	5
Modificar asistencia	Simple	5
Eliminar asistencia	Simple	5
Generar Certificado alumno regular	Simple	5

Generar Certificado notas	Simple	5
Ver calendario	Simple	5
Ingresar calendario	Simple	5
Modificar calendario	Simple	5
Eliminar calendario	Simple	5
Ver documentos	Simple	5
Ingresar documento	Simple	5
Modificar documento	Simple	5
Eliminar documento	Simple	5
UUCW = 5 + 10 + 10 + 10 + 10 + 10 + 5 + 5 + 5 + 5 + 5+ 5 + 5 + 5 + 5 + 5 + 5 + 5 + 5 + 5+ 5 + 5 + 5 + 5 + 5		**UUCW** = 145

Tabla 49: Cálculo UUCW.

8.3 Calcular TCF (Technical Complexity Factors)

Descripción	**Peso**	**Factor (Complejidad percibida)**	**Total**
Sistema distribuido	2	1	2
Rendimiento	1	3	3
Eficiencia usuario final	1	3	3
Procesamiento interno complejo	1	3	3
El código debe ser reutilizable	1	5	5
Fácil de instalar	0.5	5	2.5
Fácil de usar	0.5	5	2.5

Portable	2	4	8
Fácil de cambiar	1	5	5
Concurrencia	1	4	4
Incluye objetivos especiales de seguridad	1	5	5
Provee acceso directo a terceras partes	1	0	0
Se requiere facilidades especiales de entrenamiento a usuario	1	5	5
TCF = 2 + 3 + 3 + 3 + 5 + 2.5 + 2.5 + 8 + 5 + 4 + 5 + 0 + 5			**TCF** = 48

Tabla 50: Cálculo TCF..

8.4 Calcular ECF (Environmental complexity factor)

Descripción	**Peso**	**Factor (Complejidad percibida)**	**Total**
Familiaridad con el modelo de proyecto utilizado	1.5	5	7.5
Personal part-time	-1	0	0
Capacidad del analista líder	0.5	3	1.5
Experiencia en la aplicación	0.5	5	2.5
Experiencia en orientación a objetos	1	0	0

Motivación	1	5	5
Dificultad del lenguaje de programación	-1	3	-3
Estabilidad de los requerimientos	2	5	10
ECF = 7.5 + 0 + 1.5 + 2.5 + 5 + -3 + 10			23,5

Tabla 51: Cálculo ECF.

8.5 Calcular UCP (use case points)

8.5.1 Calcular UUCP (Unadjusted use case points)

UUCP = UAW + UUCW
UUCP = 12 + 145
UUCP = 167

8.5.2 Calcular TCF (Technical complexity factor)

TCF = 0,6 * (0,01 *48)
TCF = 0,6 + (0,48)
TCF = 1,08

8.5.3 Calcular ECF (Environmental complexity factor)

EFC = 1,4 + (-0,03 * ECF)
EFC = 1,4 + (-0.03 * 23,5)
EFC = 1,4 +(- 0,705)
EFC = 0,695

8.5.4 Calcular UCP (Use case points)

UCP = UUCP + TCF + EFC
UCP = 167 + 1,08 + 0,695
UCP = 168,775

8.6 Estimación final

UCP = 168,775
LOE = 20

Esfuerzo total = UCP * LOE
Esfuerzo total = 3.375,5

CAPÍTULO VII: CONCLUSIONES

Debido a esta pandemia, que afronta nuestro país, se ha dejado ver que algunos establecimientos educacionales necesitan una actualización en sus procesos, ya que algunos, al dictar clases de manera presencial, se vieron afectados por las condiciones que cumple cada localidad del país, ya sea temporalmente o que sus tiempos de respuestas han aumentado en documentos o información académica del alumnado.

Uno de los problemas que generó esta pandemia, fue la abrupta separación de los alumnos y profesores, ya que pasaron de tener clases presenciales a tener clases ”online”, lo que ha dificultado, entre otras cosas, la entrega oportuna de los conocimientos necesarios para seguir su formación académica, la entrega de material pedagógico y la obtención de calificaciones. Debido a esto, surgió la necesidad de un sistema que permita la entrega de este material, como así también un sistema que permita optimizar la gestión académica de los estudiantes.

Para llevar a cabo este proyecto se necesitó conocer los procesos que fueron afectados por esta situación que se vive a nivel país. De estos procesos se priorizaron los de mayor impacto a nivel educacional.

La plataforma fue realizada de manera que pudiese ser adaptada hacia otras áreas en un futuro para que el establecimiento cuente con un sistema más automatizado. El uso de la metodología iterativa-incremental permitió poder entregar pequeñas iteraciones del sistema y así poder gestionar los cambios correspondientes, como también los nuevos requerimientos que pudiesen salir en el desarrollo de este proyecto.

Además de poder resolver las necesidades del establecimiento, este proyecto fue una gran experiencia para la formación profesional del desarrollador, como también poder comprender y analizar las funciones internas de un colegio en torno a la gestión académica de los alumnos.

CAPÍTULO VIII: BIBLIOGRAFÍA

Free Bootstrap admin template. (s/f). Adminlte.io. Recuperado el 16 de marzo de 2021, de https://adminlte.io/

Taylor Otwell. Laravel - The PHP Framework For Web Artisans. (s. f.-a) Recuperado 16 de marzo de 2021, de https://laravel.com/docs/8.x

Otto, M., Thornton, J., & Bootstrap contributors. (s/f). Bootstrap. Getbootstrap.com. Recuperado el 16 de marzo de 2021, de https://getbootstrap.com/

Pressman, R. (2010). Ingeniería del Software, un enfoque práctico. Editorial McGraw Hill. 7ma edición.

Puntos de casos de uso (S/f). Www.uv.mx. Recuperado el 16 de marzo de 2021, de https://www.uv.mx/personal/asumano/files/2010/07/PUNTOS-DE-CASOS-DE-USO-2011.pdf

Ramírez, P. (2017, febrero 1). Modelo Vista Controlador MVC: ¿Qué es y para qué sirve? Com.co. https://itsoftware.com.co/content/modelo-vista-controlador-mvc-sirve/

Sommerville, I. (2011) Ingeniería de software. Editorial Pearson Educación, 8va edición.

ANEXOS

ANEXO 1: CARTA GANTT

1	**Análisis e inicio del proyecto**		
1.1	Reunion y toma de requerimientos	19/10/20	21/10/20
1.2	Analisis de requerimientos	22/10/20	26/10/20
1.3	Entrega de propuesta	28/10/20	28/10/20
2	**Primera iteracion**		
2.1	Entrevistas a docentes	02/11/20	04/11/20
2.2	Modelamiento de base de datos	05/11/20	11/11/20
2.3	Desarrollo iteracion 1	12/11/20	17/12/20
2.4	Pruebas	18/12/20	24/12/20
2.5	Primera entrega	25/12/20	28/12/20
3	**Segunda iteracion**		
3.1	Verificacion y correccion de errores	29/12/20	30/12/20
3.2	Desarrollo iteracion 2	31/12/20	29/01/21
3.3	Pruebas unitarias	01/02/21	04/02/21
3.4	Pruebas de integracion	05/02/21	10/02/21
3.5	Segunda entrega	11/02/21	12/02/21
4	**Etapa Final**		
4.1	Correciones finales	15/02/21	18/02/21
4.2	Entrega final del proyecto	19/02/21	24/02/21
4.3	Entrega informe final	26/02/21	26/02/21

Figura 40: Anexo Carta Gantt.