

**UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE QUITO**

**CARRERA:**

**INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:**

**Ingenieros de Sistemas**

**TEMA:**

**DESARROLLO DE UNA APLICACIÓN MÓVIL Y UNA APLICACIÓN WEB  
PARA LA UNIDAD EDUCATIVA AUTOGESTIONARIA “SOLIDARIDAD”  
PARA LA GESTIÓN DE LA INFORMACIÓN Y EL SEGUIMIENTO  
ACADÉMICO DE LOS ESTUDIANTES POR PARTE DE LOS PADRES DE  
FAMILIA.**

**AUTORES:**

**VERÓNICA ELIZABETH RODRÍGUEZ CALVOPIÑA  
DAMIÁN ALEXANDER CRUZ GORDILLO**

**TUTOR:**

**GAIBOR NARANJO WALTER FERNANDO**

**Quito, agosto 2020**

## CESIÓN DE DERECHOS DE AUTOR

Nosotros, Verónica Elizabeth Rodríguez Calvopiña con documento de identificación N° 1719416545 y Damián Alexander Cruz Gordillo con documento de identificación N° 1716625296, cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: “DESARROLLO DE UNA APLICACIÓN MÓVIL Y UNA APLICACIÓN WEB PARA LA UNIDAD EDUCATIVA AUTOGESTIONARIA “SOLIDARIDAD” PARA LA GESTIÓN DE LA INFORMACIÓN Y EL SEGUIMIENTO ACADÉMICO DE LOS ESTUDIANTES POR PARTE DE LOS PADRES DE FAMILIA”, mismo que ha sido desarrollado para la obtención del título de: INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



-----  
VERÓNICA ELIZABETH  
RODRÍGUEZ CALVOPIÑA

C.I.: 1719416545



-----  
DAMIÁN ALEXANDER  
CRUZ GORDILLO

C.I.: 1716625296

Quito, agosto de 2020.

## DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “DESARROLLO DE UNA APLICACIÓN MÓVIL Y UNA APLICACIÓN WEB PARA LA UNIDAD EDUCATIVA AUTOGESTIONARIA “SOLIDARIDAD” PARA LA GESTIÓN DE LA INFORMACIÓN Y EL SEGUIMIENTO ACADÉMICO DE LOS ESTUDIANTES POR PARTE DE LOS PADRES DE FAMILIA”, realizado por los estudiantes Verónica Elizabeth Rodríguez Calvopiña y Damián Alexander Cruz Gordillo, obteniendo un proyecto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.



-----  
GAIBOR NARANJO WALTER FERNANDO

C.I.: 1713106647

Quito, agosto de 2020

## **DEDICATORIA**

A mis padres que siempre me motivan a ser una buena persona, a mis hijos Luciana y Demian que son mi corazón y a Damián por ser parte de este logro importante en nuestras vidas, por su constancia y esfuerzo para alcanzarlo.

Verónica Elizabeth Rodríguez Calvopiña

A Luciana y Demian, porque son mi motivación principal y mi inspiración para seguir adelante y a Verónica porque siempre me acompaña en ese camino y búsqueda de superación.

Damián Alexander Cruz Gordillo

## **AGRADECIMIENTO**

Agradecemos a la Universidad Politécnica Salesiana por su contribución al formarnos académica y profesionalmente y a nuestro tutor de tesis, Ing. Walter Gaibor por su dedicación y apoyo constante.

Agradecemos a nuestros padres, nuestros suegros, cuñados y hermanos que siempre nos brindan apoyo, aliento y la certeza de que podemos contar con ellos.

Verónica Elizabeth Rodríguez Calvopiña

Damián Alexander Cruz Gordillo

## ÍNDICE

CESIÓN DE DERECHOS DE AUTOR.....	2
DECLARATORIA DE COAUTORÍA DEL TUTOR .....	3
INTRODUCCIÓN .....	1
Antecedentes .....	1
Descripción del Problema .....	1
Justificación.....	3
Objetivo Principal.....	4
Objetivos Específicos.....	4
CAPÍTULO 1 .....	5
Marco Teórico .....	5
1.1    La Escuela Unidad Educativa Autogestionaria Solidaridad .....	5
1.2    Aplicaciones de Gestión Educativa .....	6
1.3    Aplicaciones Web.....	7
1.4    Aplicaciones Móviles .....	9
1.4.1    Aplicaciones Móviles Nativas.....	9
1.4.2    Aplicaciones Móviles Web .....	10
1.4.3    Aplicaciones Híbridas.....	10
1.4.4    Aplicaciones Progresivas Web.....	11
1.5    Sistemas Operativos Móviles .....	11
1.5.1    Sistema Operativo Android.....	12
1.6    Herramientas de Desarrollo de Software .....	13
1.6.1    Front End.....	13
1.6.2    Back End .....	14
1.6.3    HTML 5 .....	14
1.6.4    CSS3 .....	15

1.6.5	JavaScript.....	16
1.6.6	AJAX.....	16
1.6.7	jQuery.....	17
1.6.8	Bootstrap.....	18
1.6.9	PHP.....	18
1.6.10	Codeigniter.....	20
1.6.11	MySQL.....	20
1.6.12	Servicios Web Rest.....	21
1.6.13	Node JS.....	22
1.6.14	React Native.....	22
1.6.15	Arquitectura Modelo Vista Controlador.....	23
1.7	Marco Metodológico.....	24
1.7.1	Miembros del Equipo Scrum.....	25
1.7.2	Herramientas Scrum.....	25
1.7.3	Ciclo de trabajo.....	26
CAPÍTULO 2.....		28
Análisis y Modelado de Requerimientos.....		28
2.1	Introducción a los requerimientos.....	28
2.2	Proceso manual de registro de calificaciones.....	28
2.3	Historias de Usuario.....	30
2.4	Definiciones del Product Owner.....	32
2.5	Diagrama de Contexto.....	34
2.6	Diagramas de Casos de uso.....	34
2.6.1	Caso de uso del Administrador del Sistema.....	35
2.6.2	Caso de uso de Maestros.....	36
2.6.3	Caso de uso de Representantes.....	37
2.7	Diagramas de Secuencia.....	37

2.7.1	Diagrama de Secuencia del Administrador.....	38
2.7.2	Diagrama de Secuencia del Maestro.....	39
2.7.3	Diagrama de Secuencia del Representante .....	39
2.7.4	Diagrama de Actividades.....	40
2.8	Proceso de registro de calificaciones.....	41
2.9	Diagrama de Datos .....	42
2.10	Diagrama de Clases .....	44
CAPÍTULO 3 .....		45
Diseño del Sistema .....		45
3.1	Diseño de la Arquitectura del Sistema.....	45
3.2	Diagramas de Paquetes .....	46
3.3	Diagrama de Componentes .....	47
3.4	Diseño de Interfaces .....	49
3.4.1	Diseño de la Pantalla de Login.....	49
3.4.2	Diseño de la Pantalla Principal.....	50
3.4.3	Diseño de Pantalla Interna .....	51
3.4.4	Diseño de Pantallas de la Aplicación Móvil .....	52
3.5	Elementos de diseño .....	52
CAPÍTULO 4 .....		53
Desarrollo del Sistema .....		53
4.1	Configuración del IDE.....	53
4.2	Plantilla Bootstrap .....	54
4.3	Clases del Sistema .....	54
4.4	Métodos del Sistema.....	55
4.5	Servicios API Rest del Sistema.....	58
4.6	Codificación .....	59
4.7	Perspectiva Visual de Aplicación Web.....	63

4.8	Perspectiva Visual de Aplicación Móvil.....	67
	CAPÍTULO 5 .....	69
	Pruebas y Correcciones Realizadas .....	69
5.1	Pruebas del sistema.....	69
5.2	Pruebas con técnica de caja negra .....	69
5.2.1	Pruebas funcionales (casos de uso / historias de usuario).....	69
5.2.2	Partición de equivalencias.....	71
5.3	Pruebas de Usabilidad.....	72
5.3.1	Resultados de las Pruebas .....	75
5.3.2	Síntesis de los Resultados Obtenidos en las Pruebas de Usabilidad .....	78
5.4	Pruebas de rendimiento utilizando Jmeter .....	79
	CONCLUSIONES .....	82
	RECOMENDACIONES .....	84
	LISTA DE REFERENCIAS .....	85

## ÍNDICE DE TABLAS

Tabla 1. Estructura de Calificaciones .....	29
Tabla 2. Historias de Usuario .....	31
Tabla 3. Lista de las clases del sistema.....	55
Tabla 4. Lista de los métodos del sistema.....	56
Tabla 5. Lista de los servicios Api Rest del sistema.....	58
Tabla 6. Esquema de las pruebas funcionales para los casos de uso.....	70
Tabla 7. Clasificación de entradas .....	71
Tabla 8. Casos de prueba .....	72
Tabla 9. Listado de preguntas propuestas .....	73
Tabla 10. Respuestas posibles para la encuesta.....	73

## ÍNDICE DE FIGURAS

Figura 1. Organización administrativa y pedagógica de la UEAS.....	6
Figura 2. Arquitectura de tres capas .....	8
Figura 3. Separación de las capas gracias al patrón Vista Modelo Controlador .....	24
Figura 4. Descripción gráfica del ciclo de trabajo en la Metodología Scrum.....	27
Figura 5. Diagrama de modelo de contexto del sistema y su entorno externo.....	34
Figura 6. Diagrama de caso de uso del actor administrador del sistema .....	35
Figura 7. Diagrama de caso de uso del actor Maestro .....	36
Figura 8. Diagrama de caso de uso del actor Maestro .....	37
Figura 9. Diagrama de secuencia en el proceso del Administrador .....	38
Figura 10. Diagrama de secuencia para el usuario maestro .....	39
Figura 11. Diagrama de secuencia del usuario representante .....	40
Figura 12. Diagrama del flujo de actividades para el perfil Maestro .....	41
Figura 13. Diagrama que muestra el proceso el registro de calificaciones.....	42
Figura 14. Diagrama del Modelo Entidad-Relación de la Base de Datos. ....	43
Figura 15. Diagrama de clases y objetos del sistema. ....	44
Figura 16. Modelo de requerimientos hacia el diseño del software. ....	45
Figura 17. Diagrama de Paquetes organizados según el patrón MVC. ....	47
Figura 18. Diagrama de Componentes del sistema y su interrelación.....	48
Figura 19. Diseño de la página para la autenticación de los usuarios. ....	49
Figura 20. Diseño de la página principal o página del panel general.....	50
Figura 21. Diseño de la página interna o página de las opciones internas.....	51
Figura 22. Diseño de las pantallas para la aplicación móvil.....	52
Figura 23. Pantalla Codeigniter y la separación de las capas – Sistema en web.....	53

Figura 24. Código de inicio de la clase Student. ....	59
Figura 25. Código de la función student. ....	60
Figura 26. Código fuente de la programación del servicio students.....	61
Figura 27. Código fuente de la pantalla para Login .....	62
Figura 28. Captura de la pantalla de inicio de sesión .....	63
Figura 29. Captura de pantalla de una opción del perfil administrador.....	64
Figura 30. Captura de pantalla para el ingreso de estudiantes .....	65
Figura 31. Captura de pantalla para el registro de calificaciones.....	66
Figura 32. Captura de pantalla para la consulta de calificaciones.....	67
Figura 33. Capturas de pantalla para la consulta de calificaciones en app .....	68
Figura 34. Atributos de la Usabilidad según la Norma ISO 25010.....	72
Figura 35. Captura de pantalla de la encuesta en Google Forms .....	74
Figura 36. Gráfico de los resultados obtenidos a pregunta de Operatividad .....	75
Figura 37. Gráfico de los resultados obtenidos a pregunta de Operatividad .....	75
Figura 38. Gráfico de los resultados - pregunta capacidad para ser usado. ....	76
Figura 39. Gráfico de los resultados obtenidos a pregunta de entendimiento .....	76
Figura 40. Gráfico de los resultados obtenidos a pregunta de aprendizaje.....	77
Figura 41. Gráfico de los resultados obtenidos a pregunta sobre visual.....	77
Figura 42. Gráfico de los resultados obtenidos a pregunta sobre visual.....	78
Figura 43. Porcentajes de los tipos de respuestas obtenidos en la encuesta .....	79
Figura 44. Plan de pruebas para Jmeter .....	80
Figura 45. Resultados obtenidos en las pruebas con Jmeter .....	81
Figura 46. Resultados con error obtenidos en las pruebas con Jmeter .....	81

## **RESUMEN**

Este proyecto busca la implementación de un sistema para la gestión de información académica de la Unidad Educativa Autogestionaria Solidaridad. El sistema está compuesto por una aplicación Web, para el uso del personal de la institución y una aplicación móvil, para que los representantes de los estudiantes puedan consultar la evolución académica de los alumnos en la institución.

El presente documento contiene el marco teórico en el cual se detallan las herramientas tecnológicas usadas para la construcción del sistema y la metodología adoptada para el desarrollo del proyecto. En los capítulos posteriores constan: el análisis de información relevante y de los requerimientos funcionales de los usuarios, el diagrama de los procesos involucrados, los diagramas de casos de uso, los diagramas de secuencia y el diseño de la arquitectura bajo la cual se construyó el sistema.

En el documento también se encuentran; la descripción del sistema con tablas que contienen las clases y los métodos, capturas de partes del código fuente e imágenes de las pantallas del sistema en funcionamiento.

Finalmente se detallan las pruebas a las que el sistema es sometido y los resultados obtenidos, las últimas páginas contienen la exposición de las conclusiones y recomendaciones.

## **ABSTRACT**

This project seeks the implementation of a system for the management of academic information for the school “Unidad Educativa Autogestionaria Solidaridad”. The system has two parts: first is a Web application, for the use of the institution's staff and second is a mobile application, where parents can see the academic evolution of the students.

This document contains the theoretical framework in which we detail the technological tools used for system construction and the methodology adopted for the development of the project. The subsequent chapters include: the analysis of relevant information and the functional requirements of the users, the diagrams of processes involved, the diagrams of use cases, the diagrams of sequence and the design of the architecture under which the system was constructed.

The document also contains; the system description with tables that contain the classes and methods, source code captures and screens with the system in operation.

Finally, document has the tests which the system was subjected and the results obtained, the last pages contain for the exposition of the conclusions and recommendations.

# INTRODUCCIÓN

## **Antecedentes**

En la actualidad se puede recurrir a los sistemas informáticos y a la tecnología para casi todas las actividades del ser humano, el ámbito de la educación no es ajeno a esta premisa y en éste la comunicación efectiva entre maestros y padres de familia se convierte en una necesidad imperiosa para el desarrollo académico y bienestar de los estudiantes. Si se considera que el uso de teléfonos inteligentes en Ecuador se ha vuelto común entre la mayor parte de los adultos, se puede explotar este acceso a la tecnología para implementar soluciones que les permitan acceder a información relevante, sobre todo cuando se trata de información sobre el desenvolvimiento académico de sus hijos.

## **Descripción del Problema**

La Unidad Educativa Autogestionaria Solidaridad (UEAS), es una institución que brinda servicios educativos y que por el momento no cuenta con un sistema de gestión de información de los estudiantes y en consecuencia tampoco cuenta con un sistema de gestión académica. Los escasos recursos económicos de la institución no le han permitido adquirir un sistema tecnológico que satisfaga las necesidades de autoridades, maestros y padres de familia para la gestión, administración y consulta de información relevante.

En la institución, el manejo de la información sobre las calificaciones de los estudiantes se realiza con hojas de Excel y mediante el uso de un formato pre-establecido por el Ministerio de Educación, la información se almacena de forma local en computadoras de escritorio, solo se puede acceder a toda esta información mediante archivos que se comparten en correos electrónicos, pen drives o en los formatos impresos. Para el informe a los padres de familia de las calificaciones obtenidas por los estudiantes, el proceso incluye dos reuniones, cada una al final del quimestre, en las cuales los maestros entregan un reporte de las materias que cursan los estudiantes y la calificación obtenida en cada una de esas materias en el período correspondiente. Aunque el ausentismo de los padres de familia a este tipo de reuniones es bajo, preocupa el poco acceso a la información que tienen los padres que no logran asistir a las reuniones. Por otro lado, cabe la posibilidad de que el acceso a esta información sea tardío ya que, al ser reuniones de entregas de notas, éstas ya no se pueden modificar.

Sin un sistema o plataforma que permita la consulta de calificaciones en línea y bajo demanda y sin la notificación oportuna a los padres de familia de las novedades que pudiesen existir en torno al desempeño académico de los estudiantes de la institución, se está dejando a un lado la oportunidad de optimizar los procesos de comunicación entre los padres de familia y maestros, por otro lado, se limita a las autoridades del establecimiento a información valiosa que organizada puede permitir realizar procesos de análisis de datos y estadísticos, pero sobre todo no permite realizar un seguimiento de la evolución académica de los estudiantes de manera individual ni grupal.

## **Justificación**

El proyecto tecnológico que se propone en el presente documento busca solventar la necesidad de los actores del sistema educativo; maestros, estudiantes y padres de familia de la Unidad Educativa Autogestionaria Solidaridad, de contar con una aplicación que les permita acercarse entre ellos, comunicarse de forma más efectiva y establecer parámetros de control eficaces sobre la evolución de aprendizaje de los estudiantes.

La importancia del proyecto radica por un lado, en dotar de una aplicación tecnológica a los maestros para que la publicación de las calificaciones e indicaciones generales puedan realizarse de forma ágil y oportuna y sin la demanda de grandes recursos tanto humanos como financieros y por otra parte la importancia se halla en que los padres de familia puedan aprovechar las capacidades de sus dispositivos móviles o celulares para que puedan informarse a tiempo de las actividades y novedades de la institución educativa en donde estudian sus hijos y sobre todo de las calificaciones que van obteniendo en las tareas, pruebas, actividades y lecciones que entregan cada semana.

El acceso temprano a los resultados académicos de los estudiantes puede permitir a los padres de familia tomar acciones inmediatas cuando identifiquen una baja calificación, beneficiando finalmente al rendimiento de los estudiantes. Estas tareas facilitan el trabajo de los docentes, quienes pueden manejar de forma más eficiente la comunicación con padres de familia y establecer parámetros de control más efectivos sobre los estudiantes que así lo requieran.

## **Objetivo Principal**

Desarrollar una aplicación móvil y una aplicación web para la Unidad Educativa Autogestionaria “Solidaridad” para la gestión de la información y el seguimiento académico de los estudiantes por parte de los padres de familia.

## **Objetivos Específicos**

Desarrollar el módulo de administración que permita crear, eliminar y editar usuarios (roles y perfiles), materias y grados.

Desarrollar el módulo de estudiantes que permita ingresar, editar la información de estudiantes y la asignación de los respectivos grados.

Desarrollar el módulo de maestros que permita administrar la información académica de los estudiantes como el ingreso de calificaciones de aprovechamiento como de comportamiento, además el envío de notificaciones a padres de familia.

Desarrollar los reportes solicitados por el Ministerio de Educación a la Institución y que luego se puedan obtener mediante una consulta en el módulo de administración.

Desarrollar un módulo para los representantes o padres de familia que permita la visualización de las calificaciones en un entorno Web y en una aplicación móvil.

## **CAPÍTULO 1**

### **Marco Teórico**

#### **1.1 La Escuela Unidad Educativa Autogestionaria Solidaridad**

La Unidad Educativa Autogestionaria Solidaridad está ubicada en el barrio Ciudadela Plywood, Parroquia Quitumbe en el Sur del Distrito Metropolitano de Quito. La escuela fue fundada en el año 1.997 por los dirigentes de la Cooperativa Simón Rodríguez que vieron la necesidad de crear una escuela particular autosuficiente financieramente, pero sin fines de lucro. Para el proyecto de creación de la escuela consiguieron un terreno por parte del Ministerio de Bienestar Social cedido en comodato. Construyeron las aulas con formas hexagonales, cuidaron mantener los amplios espacios verdes y la mayor cantidad de árboles.

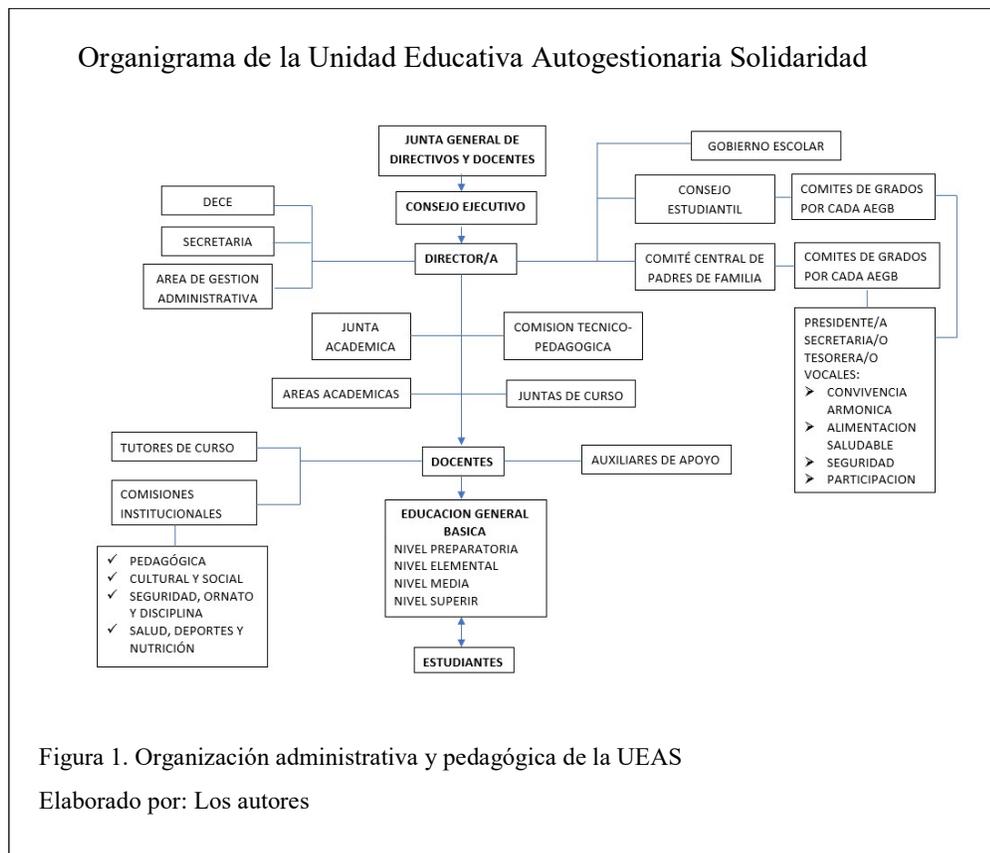
La escuela cuenta con: piscina, 3 canchas de fútbol con césped natural, cancha de básquet, vivero, huerto, sendero con árboles, salón de actos, biblioteca y área de juegos infantiles y 2 laboratorios de computación e informática.

La institución tiene 160 estudiantes repartidos en 11 niveles de educación; uno de pre-básica y 10 niveles de educación general básica. Existe un solo paralelo por nivel, son 10 los maestros encargados de dictar las diferentes materias y 3 profesionales encargados del DECE.

La administración de la institución se divide en dos áreas: La primera es el área pedagógica a cargo de la Directora, encargada de la parte educativa y de la relación con el Ministerio de Educación. La otra parte es el área financiera, a cargo de los directivos de la Coop. Simón Rodríguez que está registrada en la Superintendencia

de Economía Popular y Solidaria. El organigrama de la institución se puede ver en la figura 1.

Actualmente la institución está inmersa en el proceso para ser una escuela fiscomisional y de esa forma contar con apoyo económico por parte del Estado.



## 1.2 Aplicaciones de Gestión Educativa

En Ecuador existen empresas que se dedican al desarrollo e implementación de sistemas de gestión educativa para escuelas y colegios que pueden costear este tipo de plataformas tecnológicas, un ejemplo de ello es Idukay, empresa cuyo software ofrece la automatización de procesos académicos y la mejora de la comunicación dentro de la institución educativa. (Idukay – Optimiza El Aprendizaje, s.f.). Así como esta

empresa existen otras que inclusive ya han hecho propuestas comerciales a los directivos de la UEAS, sin embargo, como lo menciona la Directora en una de las entrevistas; este tipo de sistemas son bastante costosos y están lejos del alcance del presupuesto de la institución, estos sistemas los pueden pagar escuelas y colegios cuyas pensiones son hasta 20 veces más altas de lo que cobra la escuela Solidaridad por concepto de pensiones.

Las principales bondades de estos sistemas es que digitalizan procesos que antes eran totalmente manuales, como el registro de estudiantes, el registro de notas y la entrega de las mismas a los padres de familia.

Brindar acceso a los padres de familia a la información académica de los estudiantes mediante aplicaciones Web o aplicaciones móviles genera una mayor y oportuna atención a temas relevantes tanto de la institución como del desenvolvimiento de los estudiantes. Una comunicación eficiente beneficia a la relación entre estudiantes, maestros y padres de familia y disponer de canales de comunicación accesibles hace efectiva esa comunicación. Un canal que está al alcance de la mayoría de personas son los celulares, su uso para la comunicación genera mayor intervención por parte de los padres de familia. (Can, 2016)

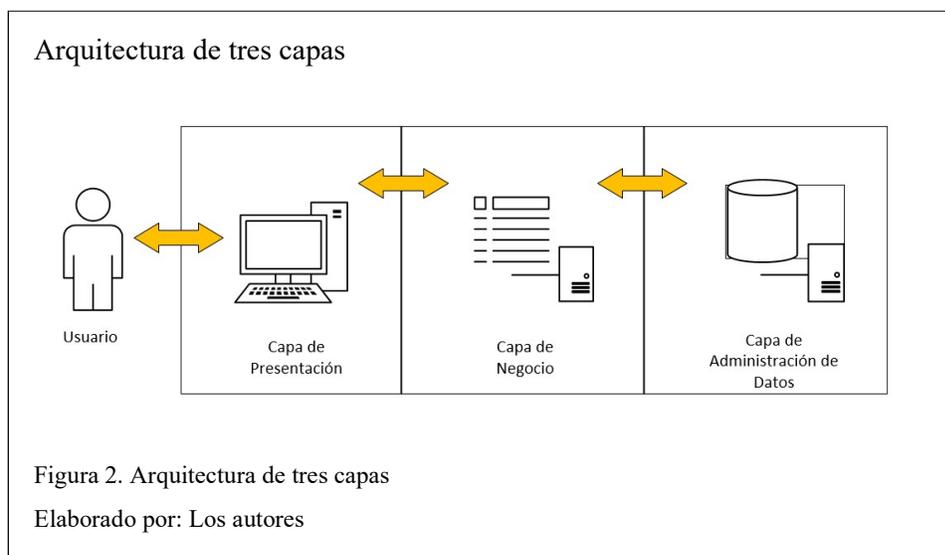
### **1.3 Aplicaciones Web**

Las aplicaciones Web son sistemas de tipo cliente/servidor y que básicamente contienen un conjunto de páginas a las que accede un usuario para la consulta de información que le sea útil. Las aplicaciones acceden a un servidor Web a través de Internet o de una Intranet mediante un navegador o browser. En otras palabras, las aplicaciones Web son programas codificados en lenguajes que los navegadores web

pueden interpretar para la ejecución correspondiente y la presentación en una pantalla al usuario final.

Las aplicaciones Web están organizadas en tres niveles o capas, las mismas que agrupan las distintas funcionalidades o reglas propias de la estructura, los niveles son:

- El nivel superior, encargado de la interacción con el usuario, también se lo conoce como Capa de Presentación y es la capa que ejecuta el navegador.
- El nivel intermedio, encargado de procesar los datos según las definiciones o lógica del negocio. Contiene las reglas, los eventos, las condiciones y las actividades necesarias para atender las peticiones del usuario, también es conocido como capa de negocio.
- El nivel inferior, encargado de proporcionar la información. Aquí se encuentra la gestión de la Base de Datos; almacenamiento, entrega y preservación de los datos. Esta capa también es conocida como Capa de Administración de Datos. (Luján Mora, 2002)



## **1.4 Aplicaciones Móviles**

Se denominan aplicaciones móviles al conjunto de funciones que tienen un fin específico en un celular. Es común pensar que las aplicaciones móviles aparecen junto con las tiendas de aplicaciones, pero en realidad éstas forman parte del entorno tecnológico desde mucho antes, las aplicaciones aparecen junto con los Smartphones o teléfonos inteligentes. Cada función de un teléfono inteligente es una aplicación, por ejemplo, la aplicación para hacer llamadas de voz, la aplicación para guardar contactos, etc. Con el lanzamiento del primer iPhone se populariza el término “App” y además se genera un cambio en la percepción de la utilidad de las aplicaciones pues se popularizan y se generan modelos de negocio que hacen rentables el desarrollo de las mismas. Desde el punto de vista tecnológico existen varios tipos de aplicaciones móviles:

### **1.4.1 Aplicaciones Móviles Nativas**

Son aplicaciones desarrolladas para un sistema operativo específico; Android, iOS y anteriormente también para Windows Phone, sin embargo, éste último ha dejado de ser usado y son muy pocas marcas celulares que aún lo utilizan, mientras que desde marzo de 2020 empieza a aparecer en el mercado una nueva plataforma para aplicaciones que la empresa Huawei intenta posicionar en su afán de restar su dependencia de los servicios de Google. En el desarrollo de aplicaciones nativas es común que se tenga que adaptar el lenguaje de programación según el sistema operativo para el que se desarrolla y que se use el SDK (Software Development Kit), de cada plataforma. Este tipo de aplicaciones tiene como ventaja que se explotan las

características de los dispositivos, se logra el máximo rendimiento y el diseño es el óptimo considerando el sistema operativo del celular, mientras que su principal desventaja es que son costosos pues se multiplica el esfuerzo de desarrollo.

Dado que el componente del sistema propuesto es una aplicación móvil que debe constar dentro del grupo de aplicaciones instaladas por el representante en su dispositivo celular, debe ser una aplicación de nativa y se considera esta naturaleza para la selección de las herramientas para su desarrollo.

#### **1.4.2 Aplicaciones Móviles Web**

Este tipo de aplicaciones se desarrollan con HTML, CSS y JavaScript, aunque además se puede recurrir al uso de varios componentes y librerías que han surgido para potenciar este tipo de desarrollos. La ventaja es que el sistema se adapta a cualquier sistema operativo, básicamente se construye un sitio web diseñado para pantallas pequeñas y los usuarios lo consumen directamente desde los navegadores de sus celulares, es decir no se instalan y por lo tanto los usuarios deben ingresar por una URL. Como desventaja se encuentra que este tipo de aplicaciones no funcionan sin conexión a internet y que se limita el acceso e interacción con otras aplicaciones nativas de los dispositivos, pero sobre todo es mucho más complicado la adopción por parte de los usuarios pues los pasos para “abrir” la aplicación es más extenso.

#### **1.4.3 Aplicaciones Híbridas**

Como su nombre lo indica son una mezcla de las dos anteriores. Su desarrollo conjuga el uso de HTML, CSS y JavaScript, con el uso de un contenedor nativo, como

Cordova, por ejemplo, que a la vez permite acceder a las capacidades de los dispositivos celulares, independientemente del sistema operativo gracias a la interacción con las APIs de éstos. (Apache Cordova - Architectural Overview of Cordova Platform, s.f.)

#### **1.4.4 Aplicaciones Progresivas Web**

Este tipo de aplicaciones están siendo apenas desarrolladas y probadas, no existe abundante documentación pues las primeras implementaciones con esta tecnología se encuentran desde 2019, a pesar de ello, se conoce de algunas ventajas en la implementación de éstas. Son pequeños sistemas que tienen un diseño y la funcionalidad de una aplicación, aunque pueden ejecutarse desde el navegador pueden acceder a funciones y características del dispositivo como si fuera aplicaciones nativas como el localizador, la cámara, el micrófono o la brújula. También se pueden generar archivos compilados que las tiendas de Android y iOS aceptan para su comercialización y descarga. Sus principales ventajas se pueden resumir en tres atributos: Capaces, Confiables e Instalables. (Web-Dev - What Are Progressive Web Apps?, s.f.)

#### **1.5 Sistemas Operativos Móviles**

Los Sistemas Operativos Móviles son plataformas de software que se almacenan en los dispositivos celulares para permitir la ejecución de programas y aplicaciones.

Hoy en día existen dos sistemas operativos predominantes en el mercado; iOS y Android y aunque es posible y común generar aplicaciones que puedan instalarse en ambos sistemas operativos, en ocasiones es necesario limitar el desarrollo únicamente a uno de los dos. A la hora de seleccionar el sistema operativo para el que se va a desarrollar una aplicación es necesario identificar el grupo objetivo ya que generalmente existen segmentos de población con una mayor tendencia a usar ciertos dispositivos con su correspondiente sistema operativo. Según la International Data Corporation, la cuota de mercado de Android a nivel mundial en 2021 será del 86.5% mientras que para iOS será del 13.5%. (IDC - Smartphone Market Share - OS, s.f.)

En ciertos segmentos de la población la diferencia de la cuota de mercado es aún mayor, por ejemplo, en Banco Solidario que atiende a segmentos de población de bajos recursos con productos de micro-finanzas, el porcentaje de descargas de su aplicación móvil desde dispositivos Android es del 93.8% mientras que desde iOS es del 6.2%. Es precisamente este el segmento económico al que pertenece la mayoría los padres de familia de la UEAS, por lo cual la aplicación del proyecto se desarrolla únicamente para Android.

### **1.5.1 Sistema Operativo Android**

Android es un sistema operativo de código abierto creado y lanzado por Google bajo la licencia Apache. El sistema está basado en Linux y corre bajo un framework que contiene librerías compatibles con Java. Android fue desarrollado para ser compatible con pantallas táctiles, la primera versión aparece en octubre de 2008 y desde allí se han generado 16 versiones adicionales, la última versión fue lanzada en septiembre de 2019 y se denomina Android 10, es la única versión que no lleva el

nombre de un dulce o postre famoso como lo sí lo fueron sus antecesores. (Ahmad et al., 2013)

## **1.6 Herramientas de Desarrollo de Software**

Para la construcción del sistema para la gestión de información educativa de la UEAS se usan varios recursos tecnológicos. A continuación, se describen con detalle estos recursos tecnológicos entre los que se encuentran; lenguajes de programación, lenguajes de marcado y estilos, frameworks de desarrollo o librerías de apoyo, motores de bases de datos. Además, se comparten conceptos sobre la arquitectura o modelos de desarrollo, así como patrones de agrupación de funcionalidades.

### **1.6.1 Front End**

El Front-end también conocido como “Client-Side” o “Lado del Cliente” en español, es la parte del desarrollo correspondiente a la parte frontal de un sitio web, en otras palabras, es el área en la que se diseña y construye las páginas con las cuales interactúa el usuario final. En esta área del desarrollo se gestiona la estructura visual del sitio, los estilos y formatos de letras, tablas, vínculos, botones, menús, etc., también en esta área se revisa y se da formato al contenido que se presenta como información al usuario final. El desarrollo del área de Front End se basa principalmente en el uso de lenguaje de marcado HTML, el manejo de estilos mediante CSS.

## **1.6.2 Back End**

Es la parte o estructura de un sitio Web que tiene toda la funcionalidad por detrás para que el mismo funcione correctamente. Esta área de un sistema toma los datos, los procesa, los ordena y los presenta al usuario cuando este lo requiere. Generalmente aquí se encuentran la configuración y las consultas o peticiones a la Base de Datos, la conexión con el servidor, las reglas para el manejo de la información, la lógica para la integración entre distintos servicios web, etc. Para el desarrollo de sistemas de entorno Web se cuenta con una gama de lenguajes y herramientas que le permiten cumplir con su objetivo. Los principales lenguajes de programación y los principales motores de bases de datos que actualmente están siendo usados para la creación de sitios dinámicos son: PHP y MySQL, respectivamente.

## **1.6.3 HTML 5**

HTML5 es la última versión del lenguaje de marcado HTML (HyperText Markup Language). La versión presenta importantes avances frente a su antecesor HTML4. Las mejoras están orientadas principalmente a facilitar el desarrollo de páginas más dinámicas y de mayor alcance mediante el uso de más recursos tecnológicos y su implementación de forma más sencilla y transparente para el desarrollador. Los principales recursos, que además están clasificados según su función son los siguientes: (HTML5 - HTML | MDN, s.f.)

- **Semántica:** Los diferentes componentes de una página Web pueden ser descritos con mayor precisión y de forma más clara para la comprensión del desarrollador.
- **Conectividad:** Existen nuevas formas de comunicarse con el servidor.

- Sin conexión: Las páginas web pueden almacenar información del lado cliente y lograr varias funcionalidades sin conexión y de forma más eficiente.
- Rendimiento: Mejora los tiempos de respuesta y velocidad en la carga de las páginas gracias al uso adecuado del hardware.
- Acceso a los dispositivos: Proporciona APIs para conectar y usar, en ciertos casos, componentes internos de los dispositivos tanto de entrada como de salida.

#### 1.6.4 CSS3

Es el último estándar de CSS (Cascading Style Sheets), es un lenguaje de estilos que define la presentación de la información en documentos HTML. Mediante el uso de este lenguaje de estilos se definen las características visuales de una página Web, es decir se administran parámetros relativos a: colores, márgenes, espacios, bordes, tipos y tamaños de letra, imágenes de fondo, altura y anchura de todos los elementos, posiciones, etc.

Anteriormente los formatos de los elementos de una página Web estaban definidos en el mismo código HTML, sin embargo, actualmente lo más recomendable es el uso de CSS para el manejo de estilos ya que de esta forma se separa la estructura y la forma del documento, la estructura dada por HTML y la forma por CSS. La administración del diseño mediante CSS facilita los cambios que pueda requerir un sitio Web en cuanto a diseño pues al tener todos los elementos comunes referenciados a un tipo de formato, al modificar éste, se actualizan automáticamente todos los elementos que tienen ese estilo.

### **1.6.5 JavaScript**

Es un lenguaje interpretado en el que las instrucciones (scripts) se introducen en el código HTML. El código se envía para que sea interpretado por el cliente al cargar una página Web. Cualquier instrucción u orden de JavaScript requiere de HTML, es decir, JavaScript no puede usarse para crear aplicaciones independientes que no estén contextualizadas en un entorno Web o que requieran un navegador. Las principales características de JavaScript son: (Cobo et al., 2009)

- Al ser un lenguaje interpretado, no necesita compilación y es el propio navegador del cliente el que traduce el código.
- Es un lenguaje multiplataforma, puede correr en cualquier sistema operativo.
- Tiene la capacidad de gestionar la mayoría de eventos que una página web requiera para su funcionamiento.

### **1.6.6 AJAX**

Es el acrónimo de Asynchronous JavaScript And XML. No es propiamente una tecnología, pero sí agrupa a algunas tecnologías que pueden trabajar en conjunto o integrarse entre ellas.

La funcionalidad de AJAX se resume en la combinación que hace de las siguientes tecnologías para la creación de aplicaciones interactivas:

- HTML y Hojas de Estilo en Cascada (CSS) para presentar la información de forma ordenada, estructurada y formateada al gusto del desarrollador.

- Document Object Model (DOM) y JavaScript, para publicar dinámicamente datos o información y que la misma se presente de acuerdo al comportamiento del usuario durante la navegación.
- La principal característica de AJAX es la ventaja que brinda para presentar la información de forma dinámica y de partes de la web, es decir permite mostrar el contenido originado en una consulta realizada por HTTP sin la necesidad de cargar toda la página nuevamente, sino que mientras muestra el contenido ya consultado permiten pequeñas interacciones sin afectar al resto de la página. (Zakas et al., 2006)

### 1.6.7 jQuery

Es un conjunto de librerías y códigos para el uso de Javascript. Este framework agrupa un conjunto de clases que permiten programar aplicaciones web sin que importe el navegador en el cual el usuario final decida ejecutar la aplicación.

jQuery es un producto confiable y de gran popularidad entre los desarrolladores de aplicaciones web pues detrás de su evolución se encuentra un equipo enorme de desarrolladores encargados de la mejora continua y actualización del framework. Es gratuito, se lo puede descargar de internet sin ningún costo, y su bajo peso (91KB aprox.) permite que su uso no implique consumos exagerados de ancho de banda de Internet, como tampoco espacios grandes de almacenamiento en disco. Una de las características más destacables de jQuery es la facilidad que brinda para implementar aplicaciones que usen AJAX.

### **1.6.8 Bootstrap**

Bootstrap es una plataforma de código abierto lanzada en 2011 por empleados de Twitter y que ha sido adoptado como un framework de desarrollo de páginas Web con el uso de HTML, CSS y JavaScript y su librería de imágenes. Resulta muy útil para crear interfaces de usuario limpias y que se adapten fácilmente a todo tipo de dispositivos y pantallas, sin importar su tamaño, es decir facilita la implementación de aplicaciones que puedan resultar responsivas o adaptativas para su uso tanto en pantallas grandes como monitores, tabletas y dispositivos celulares. Es de fácil comprensión y adopción. Su técnica para la separación de una página en bloques haciendo divisiones por columnas permite definir ágilmente la estructura de una página y a la vez de los bloques que la componen. Además, Bootstrap dispone de las herramientas que permiten crear cualquier tipo de sitio web usando los estilos y elementos de sus librerías que incluyen entre otros; íconos, efectos para hacer dinámica la navegación, estilos de bordes, etc. (Winer & Spurlock, 2010)

Actualmente existen muchísimas plantillas de libre descarga que pueden usarse para evitar generar un sitio web desde cero y que el desarrollo se concentre en la funcionalidad y no necesariamente en el estilo del mismo.

### **1.6.9 PHP**

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor y es un lenguaje de programación interpretado, éste fue diseñado originalmente para la creación de Páginas web dinámicas pues permite la interpretación del lado del servidor sin la necesidad de compilaciones tal como sucede con lenguajes como JAVA. Fue creado por Rasmus Lerdorf en el año 1994; y desde allí se han generado múltiples

cambios, el lenguaje ha sido sujeto de una constante evolución que ha dado origen a las distintas versiones. (PHP - Manual, s.f.)

Las principales características de PHP son:

- Es un lenguaje de programación que puede correr en distintas plataformas o sistemas operativos, por ello se considera un lenguaje multiplataforma, no existen restricciones para que se pueda implementar en múltiples plataformas.
- Está orientado al desarrollo de aplicaciones web dinámicas que se sirvan de información consultada a una Base de Datos.
- Dado que el servidor se encarga de la ejecución del código y únicamente envía el resultado HTML al navegador, el código fuente escrito en PHP es invisible para el navegador y para el cliente, convirtiéndolo en un lenguaje de seguro.
- PHP puede conectarse con casi todos los motores de Base de Datos que se usan en la actualidad, siendo las integraciones más frecuentes las realizadas con MySQL.
- Es un lenguaje de programación versátil, puede aumentar su potencial mediante el uso de innumerables módulos también conocidos como “extensiones” que se encuentran disponibles en Internet y que reciben soporte, mantenimiento y constante actualización de los internautas.
- La página oficial de PHP (<https://www.php.net/>) contiene una muy amplia documentación, en la misma constan todas las funciones que posee el lenguaje con una gran cantidad de ejemplos.
- Finalmente, al desarrollar usando PHP no se está obligado a seguir una determinada metodología de programación ya que puede aplicar cualquier técnica de desarrollo para la generación de código ordenado, estructurado y manejable.

(PHP - EcuRed, s.f.)

### **1.6.10 Codeigniter**

Es un framework de código abierto utilizada para la construcción de aplicaciones dinámicas usando PHP. Codeigniter tiene como principal objetivo permitir la implementación de proyectos de manera más ágil y rápida pues evita la construcción desde cero de toda la estructura de un sitio web.

La principal característica de Codeigniter es que está basado en un patrón de desarrollo llamado Modelo Vista Controlador (MVC por sus siglas en inglés). Se trata de un enfoque o arquitectura de software que busca separar la lógica de la aplicación de la presentación al usuario, haciendo que el código sea estructurado y claro. En otras palabras, este patrón permite la separación de las capas del sistema tal como se presenta anteriormente en el punto 1.3 del presente documento.

### **1.6.11 MySQL**

Es un sistema gestor de bases de datos relacionales (RDBMS por sus siglas en inglés). MYSQL es un sistema muy conocido y de popularmente usado pues es simple de usar y tiene un muy buen rendimiento. Está disponible en Internet bajo licencia GPL (General Public License), es un sistema de código abierto y está basado en el lenguaje de consultas estructurado (SQL). Puede ser usado en cualquier tipo de aplicación, pero se asocia más con aplicaciones de entorno Web y de generación de información en línea. Sus principales características son: (Carrera, 2014)

- Funciona y se integra con múltiples plataformas.
- Se pueden usar procedimientos almacenados (Store Procedures) que facilitan, mejoran y automatizan la programación de interacciones con la base de datos.

- También soporta el uso de Activadores (Triggers) que ayudan a definir reglas de negocio.
- Es escalable, soporta gran cantidad de tablas y registros.
- Existe extensa documentación sobre sus APIs para C, C++, Java, Perl, PHP y otros.

### **1.6.12 Servicios Web Rest**

La World Wide Web Consortium define a los servicios web como sistemas de software diseñado para la interacción máquina a máquina a través de una red. Se pueden implementar servicios web usando diversos protocolos, entre ellos; SOAP (Simple Object Access Protocol) o REST (REpresentacional State Transfer). (Web Services Glossary, s.f.)

SOAP permite la comunicación mediante el intercambio de información codificada en XML, tiene gran potencial, pero también implica una considerable complejidad para la implementación. REST por su parte es una solución más simple para la manipulación de datos y es muy recomendado para tecnologías Web pues trabaja con estándares XML y HTTP y no tiene la necesidad de contar con una capa de transporte como lo hace SOAP. (Castro & Sánchez, 2013)

Para del desarrollo del sistema propuesto se recurre al uso de servicios web basados en tecnología REST, pues permite la creación de vistas con la información específica que se desea mostrar ya sea dentro del aplicativo web o móvil, gracias al uso de los métodos que el protocolo proporciona; POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).

### **1.6.13 Node JS**

Este es otro lenguaje de programación que termina siendo por demás interesante. Se pueden encontrar varias definiciones sobre esta tecnología, pero en la mayoría se resume a Node JS como un entorno de tiempo de ejecución, multiplataforma, de código abierto y que a pesar de que está basado en JavaScript corre del lado del servidor, su principal virtud es que está basado en eventos asíncronos, lo cual difiere de los tradicionales sistemas basados en hilos que limitan el número de conexiones recurrentes, puesto que con Node se cambia la forma en que el sistema se conecta al servidor ya que cada conexión ejecuta un evento en el motor sin la posibilidad de quedarse en un punto muerto pues no se bloquean las llamadas de entrada o salida. (IBM Developer - ¿Simplemente Qué Es Node.Js?, s.f.)

Este es el lenguaje de programación con el cual se construyen los servicios REST que consume la aplicación móvil para las consultas a la Base de Datos del sistema, tanto para el proceso de autenticación del representante como para la obtención de calificaciones, pero además se utiliza para la programación de la aplicación móvil debido a que junto a otras herramientas como React Native permite la explotación de recursos para la fabricación de aplicaciones.

### **1.6.14 React Native**

Este framework usado principalmente para la construcción de aplicaciones móviles también es conocido como React.js y es un conjunto de librerías JavaScript que ha sido exitosamente adoptado por desarrolladores de empresas como Uber y Netflix debido a las virtudes que presenta para la generación de aplicaciones. React es

de código abierto y puede ser usado tanto del lado del cliente como del lado del servidor y aunque produce aplicaciones en JavaScript, las mismas se compilan en código nativo tanto para iOS y para Android, es decir permite la producción de aplicaciones nativas para los dos principales Sistemas Operativos.

En la parte técnica, el objetivo principal de React es la generación de funciones que se ejecutan según se actualice o manipule el estado de la página o pantalla, estas funciones logran una nueva representación virtual de la página para su posterior traducción en cambios del DOM. Esto se hace de forma muy ágil pues se comparan los cambios entre la página actual y la nueva y únicamente se lanzan las actualizaciones de los cambios identificados, logrando una importante optimización en el uso de recursos y en el tiempo de carga. Para programar en React se utiliza JavaScript, HTML y CSS, pero el código se ejecuta en WebView.

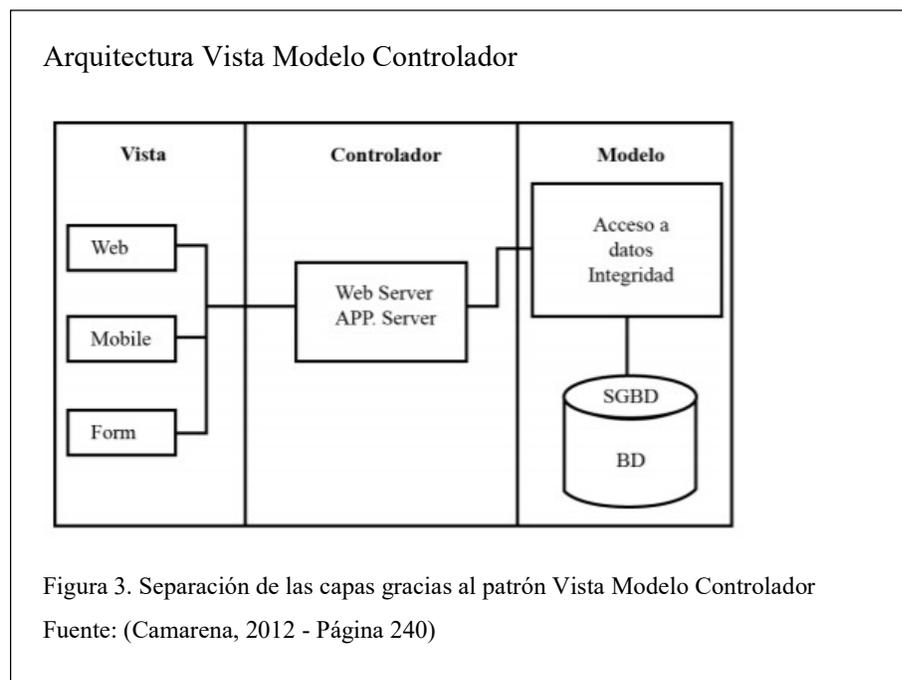
Entre las principales ventajas de usar React están que la curva de aprendizaje es corta, la experiencia de desarrollo se asemeja a programar para Web y que no se requiere compilar por cada cambio sino únicamente almacenar los cambios. (Lobato Tapia, 2019)

### **1.6.15 Arquitectura Modelo Vista Controlador**

La arquitectura Modelo Vista Controlador es un patrón de diseño que sirve para el desarrollo de aplicaciones que pretenden separar la lógica de negocios de la interfaz de usuario. Este patrón es muy usado por desarrolladores en la implementación de aplicaciones que incluyen interfaces de usuario puesto que la arquitectura separa los datos o el modelo de los datos, de la vista o presentación de los datos mismos al usuario, mientras que el controlador es el responsable de relacionar a ambos. Esa

“independencia” entre las capas convierte a esta arquitectura en una de las más recomendadas para la construcción de proyectos grandes. (Camarena, 2012)

Utilizar MVC supone algunas ventajas, una de las principales es la fácil implementación de interfaces de usuario o vistas al mismo tiempo y para distintos canales, por ello para el presente proyecto se usa la arquitectura MVC, la misma que está representada en el esquema de la figura 3.



## 1.7 Marco Metodológico

La metodología usada para la gestión del proyecto es Scrum, que brinda agilidad, flexibilidad y está orientada a trabajar de forma colaborativa entre todos los miembros del equipo; teniendo como premisa el mantener siempre buena comunicación en cada una de las fases del proyecto.

Por otro lado, es necesario mencionar que la metodología permite trabajar bajo un modelo incremental e iterativo lo que implica entregas parciales de los productos en desarrollo a los stakeholders al final de cada iteración. Esto genera un marco de comunicación dinámico y que se puedan identificar errores, mejoras, cambios pequeños de forma temprana y oportuna. En cada iteración se repite el ciclo de análisis, diseño, implementación y prueba.

### **1.7.1 Miembros del Equipo Scrum**

Los miembros de un equipo que implementa un proyecto bajo la metodología Scrum son:

- **Product Owner:** una persona con conocimientos del negocio y encargada de dar la visión del valor en cada requerimiento del cliente; en el proyecto se considera a la Lic. Margoth Tito por ser el nexo entre las autoridades de la Institución y el proyecto.
- **Scrum Master:** es un líder orientado a la ejecución del proyecto que permite que las actividades fluyan, eliminando los obstáculos que puedan existir.
- **Equipo de Desarrollo:** son todas las personas que realizan las actividades para desarrollar el proyecto.

### **1.7.2 Herramientas Scrum**

En la metodología Scrum se recurre al uso de algunas herramientas que permiten la gestión adecuada del proyecto, estas herramientas son:

- Product Backlog: aquí se listan todos los requerimientos funcionales que se exige del producto, esta es una herramienta que está en constante cambio a medida que avanza el proyecto.
- Historias de Usuario: son las funcionalidades que requiere el usuario para un uso claro, son redactadas de forma que se entienda quién, <necesita> qué <necesita> y para qué necesita.
- Backlog del Sprint: una vez que se tiene el product backlog se van priorizando en cada sprint las tareas a realizarse y esas tareas se convierten en el backlog del Sprint. (Scrum Guide, s.f.)

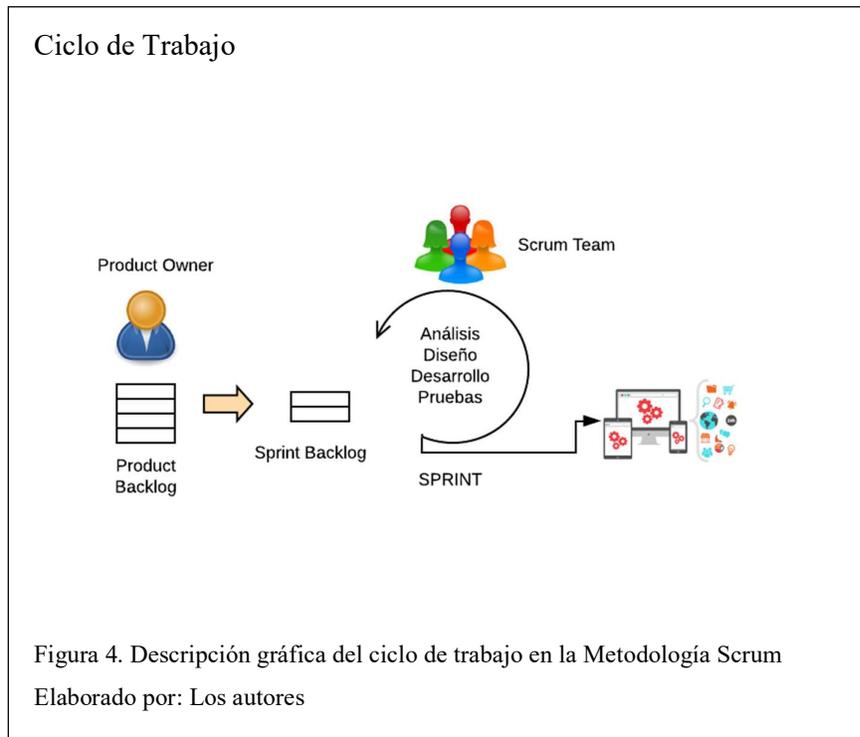
### 1.7.3 Ciclo de trabajo

En la metodología Scrum se siguen algunos pasos, denominados ciclos de trabajo, los mismos que generalmente obedecen al siguiente orden:

- El product owner genera el listado de requerimientos que constituye el product backlog, al iniciar con el Sprint se priorizan las tareas de este listado.
- Un Sprint da inicio cuando todos los involucrados se reúnen para planificarlo, con la priorización realizada por el product owner. El sprint tiene una duración de mínimo 1 semana o máximo 4 semanas y depende de la rapidez del equipo. En el caso del proyecto se define que los sprints tengan una duración de 1 semana.
- El equipo se reúne diariamente para revisar si todas las tareas están siendo desarrolladas sin problemas, de no ser así el Scrum Master como facilitador destraba los problemas.
- Al terminar cada iteración o Sprint el resultado es presentado al cliente para recibir retroalimentación, en ocasiones se generan solicitudes de cambios funcionales para

en el producto. La metodología permite flexibilizar los cambios importantes para ser incluidos en el próximo Sprint. Este proceso se repite hasta el fin del proyecto. La imagen de la figura 4 muestra este ciclo de trabajo en la metodología Scrum.

(Rad & Turley, 2013)



## **CAPÍTULO 2**

### **Análisis y Modelado de Requerimientos**

#### **2.1 Introducción a los requerimientos**

Para el análisis de la solución es necesario el levantamiento de información del proceso que la Unidad Educativa tiene actualmente, además las historias o requerimientos del usuario que permitan entender lo que la institución espera del sistema. Por otro lado, es necesario la definición del proceso que se implementa en el sistema, para lo cual se recurre a un diagrama del proceso. También se requiere el modelamiento del comportamiento del sistema y se lo realiza mediante el uso de diagramas de contexto, diagramas de casos de uso, diagramas de secuencia y actividades, y finalmente los diagramas de datos y de las clases del sistema.

#### **2.2 Proceso manual de registro de calificaciones**

En las entrevistas realizadas en las reuniones con el personal administrativo de la UEAS se recopila la información del proceso de registro de calificaciones que tienen actualmente.

Durante el proceso de matriculación, el personal administrativo a cargo actualiza formularios en Excel donde registra la información de cada alumno, esa información luego se consolida para agruparla digitalmente según el año y el paralelo.

Estas hojas de Excel también son distribuidas a los diferentes profesores para que puedan ir actualizando las calificaciones a medida que van ocurriendo los parciales del año lectivo.

El año lectivo está dividido en 2 quimestres y cada uno tiene 3 parciales. Las actividades en cada parcial son:

- Tareas en casa: la calificación final se obtiene del promedio de todas las tareas
- Actividades individuales: la calificación final se obtiene del promedio de todas las actividades individuales realizadas.
- Actividad grupal: la calificación final se obtiene del promedio de todas las actividades grupales realizadas.
- Lecciones: la calificación final se obtiene del promedio de todas las lecciones tomadas.
- Prueba final
  - La calificación final del parcial es el promedio de todas las calificaciones.
  - La calificación final es el 80% del promedio de las notas parciales y un examen final que aporta el 20% restante.

**Tabla 1.** Estructura de Calificaciones

PARCIALES					PROMEDIO QUIMESTRAL					
EVALUACIÓN FORMATIVA				SUMATIVA	PROMEDIO - P1-P2-P3	PROMEDIO PARCIALES	EQUIVALENCIA 80%	EXAMEN QUIMESTRAL	EQUIVALENCIA 20%	PROMEDIO FINAL 100%
TAREAS (TAI)	ACT. INDIVIDUAL	ACT. GRUPAL	LECCIONES	PRUEBA ESCRITA						

**Nota:** La tabla contiene los componentes para obtener las notas parciales.

La dirección de la escuela, a través de los maestros, realiza reuniones de padres de familia en el final de cada parcial para entregar un reporte físico de las calificaciones obtenidas por los estudiantes.

Los archivos Excel con las calificaciones de cada estudiante y el consolidado de las calificaciones de los estudiantes por cada grado son enviados a la persona encargada de consolidar y generar los reportes finales en otros documentos Excel. Finalmente, la persona encargada organiza la información, la consolida y genera el reporte que solicita el Ministerio de Educación.

### **2.3 Historias de Usuario**

Luego de la revisión del proceso actual se realiza el levantamiento de los requerimientos funcionales y de usuario, gracias a las entrevistas y reuniones con los interesados se determina la necesidad de que el sistema cuenta con los siguientes módulos funcionales, lo cual no implica que sean módulos de acceso según el perfil:

- Módulo de Administración
- Módulo de Maestros
- Módulo de Estudiantes
- Módulo de Representantes

Con este levantamiento se crea el listado de requerimientos llamado Product Backlog y se prioriza las historias de usuario en conjunto con la persona designada por la institución que figura como Product Owner.

Siguiendo las recomendaciones de la metodología Scrum, se revisa el esfuerzo que toma cada historia de usuario y se acuerda que los Sprints tengan una duración de 1 semana con 36 horas de trabajo para cada semana. La tabla 2, muestra las historias de usuario levantadas y la definición del esfuerzo en términos de horas de trabajo para cada actividad.

**Tabla 2.** Historias de Usuario

Requerimiento	Historias Usuario	Esfuerzo (horas)	Iteración (Sprint)	Criterios de Aceptación
Creación del módulo Administrador	Diseño de la base de datos	8	1	Debe tener la funcionalidad de recuperar la clave El administrador ingresa, edita grados, paralelos y materias
	Desarrollar Login y página principal (Index)	10		
	Desarrollar pantalla funcionalidad grados, paralelos, mat.	18		
	Desarrollar pantalla y funcionalidad de materias	13	2	El administrador ingresa y edita materias
	Desarrollar pantalla y funcionalidad de maestros	5		El administrador ingresa y edita maestros
	Desarrollar funcionalidad asignación maestros vs materias	5		El administrador asigna un maestro por cada materia creada
Creación del módulo Estudiante	Desarrollar pantallas y funcionalidad de representantes	5	2	El administrador ingresa y edita representantes
	Desarrollar pantalla y funcionalidad de ingreso estudiantes uno a uno	8		El administrador ingresar y edita estudiantes uno a uno Se asigna a cada estudiante un representante creado
	Desarrollar pantallas y funcionalidad de ingreso estudiantes en lote	5	3	El administrador ingresa estudiantes mediante una carga masiva en un formato predefinido
	Desarrollar funcionalidad de estudiantes a paralelos	5		El administrador asigna el paralelo a cada estudiante.
Creación del módulo de Calificaciones	Desarrollar pantallas y funcionalidad de calificaciones	26	4	Maestros ingresan calificaciones
	Desarrollar pantallas y funcionalidad de calificaciones 1ro	14		Maestros ingresan calificaciones cualitativas
	Desarrollar los reportes académicos para el Ministerio de Educación	16		El administrador genera los reportes a enviar al Ministerio de acuerdo al formato solicitado por la entidad
	Desarrollar reportes de rendimiento por estudiante	6		El administrador genera reportes académicos por estudiante
	Desarrollar reportes de rendimiento por paralelo	11		El administrador genera reportes académicos por paralelo
Mensajería	Desarrollar opciones de mensajería vía correo electrónico y mensajes internos	20	5	El administrador envía correos electrónicos de forma masiva a todos los representantes El maestro envía un mensaje interno a un representante
Creación de aplicación móvil	Desarrollo de la aplicación móvil para representantes	40	6	El representante ingresa a la aplicación y revisa las calificaciones del estudiante

**Nota:** La tabla muestra los requerimientos de los usuarios, levantados como historias de usuarios con el detalle del esfuerzo que implican.

## 2.4 Definiciones del Product Owner

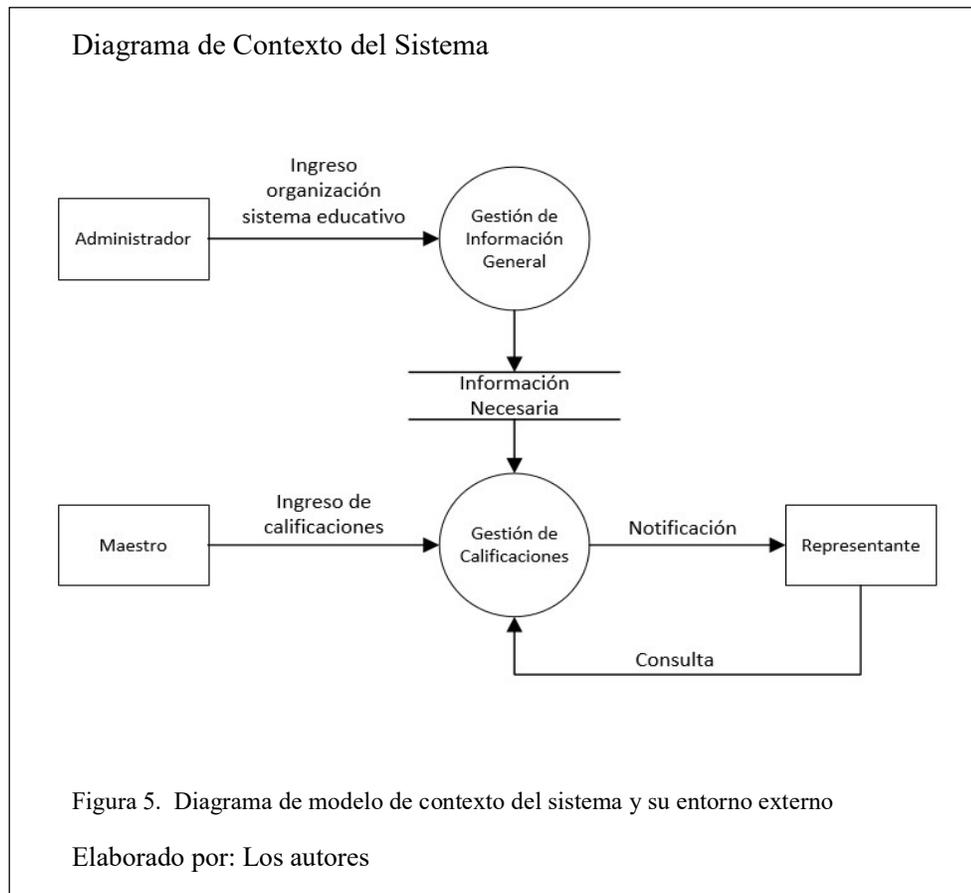
La escuela mediante la participación del Product Owner del proyecto establece algunas consideraciones o reglas particulares que deben ser tomadas en cuenta sobre las características de ingreso de datos o validaciones, las misas se detallan a continuación:

- Cuando un representante sea registrado en el sistema, se debe generar su usuario y contraseña, los mismos que deben ser iguales a su número de cédula y no debe existir la posibilidad de que los representantes modifiquen estos parámetros, así como no debe estar permitido que un representante actualice ningún campo de su información de perfil, aunque dicha información debe estar disponible en el sistema para su consulta. De esta manera se evita que los representantes puedan olvidar sus credenciales de acceso y/o actualizar información personal que es relevante para la institución. Esta práctica es habitual en sistemas de consulta no transaccionales.
- Los maestros sí pueden cambiar su contraseña de acceso a la plataforma, la contraseña debe ser igual a su número de cédula en el momento de la creación o registro de un nuevo maestro.
- Solo existe un usuario de tipo administrador, es decir no se contempla una opción para la creación de este tipo de perfil.
- El sistema debe permitir el envío de comunicaciones desde el administrador o desde el maestro, sin embargo, los representantes no deben tener la opción de enviar mensajes o comunicaciones a los maestros o al administrador.

- Cuando en el sistema se genere un nuevo grado, se debe crear por defecto el paralelo “A”, debido a que solo existe un paralelo por grado, sin excluir que posteriormente se puedan añadir paralelos.
- El tratamiento de las calificaciones en el caso de estudiantes que no alcancen la nota mínima al final del año lectivo para aprobar el mismo, se realiza mediante el mismo módulo calificaciones puesto que se contemplan actividades de recuperación que pueden actualizar las calificaciones ya registradas.
- La actualización y registro de calificaciones no debe tener una fecha máxima de aplicación, tampoco se debe restringir la actualización de calificaciones anteriores. Sí debe existir una restricción para las pruebas de los parciales; la opción para el registro de calificación de la prueba del parcial no debe habilitarse si no se ha ingresado al menos una nota de la actividad de un parcial.
- Los estudiantes y sus representantes no deben eliminarse de las tablas, inclusive si el año escolar ha concluido, en su lugar se debe manejar un registro de estado que permita la re-utilización de los registros para posteriores matrículas. Además, la información es de utilidad para el registro histórico de los estudiantes y para la consulta en eventuales necesidades de información pasada.
- A pesar de que los reportes contienen la información registrada en el sistema, los mismos deben descargarse en un formato editable, por tal motivo los reportes se deben generar como archivos de Excel.

## 2.5 Diagrama de Contexto

El diagrama de contexto permite describir de forma gráfica la forma en que se delimitan los márgenes de acción entre el sistema y el mundo exterior mediante la interfaz y cómo ésta les permite relacionarse.



## 2.6 Diagramas de Casos de uso

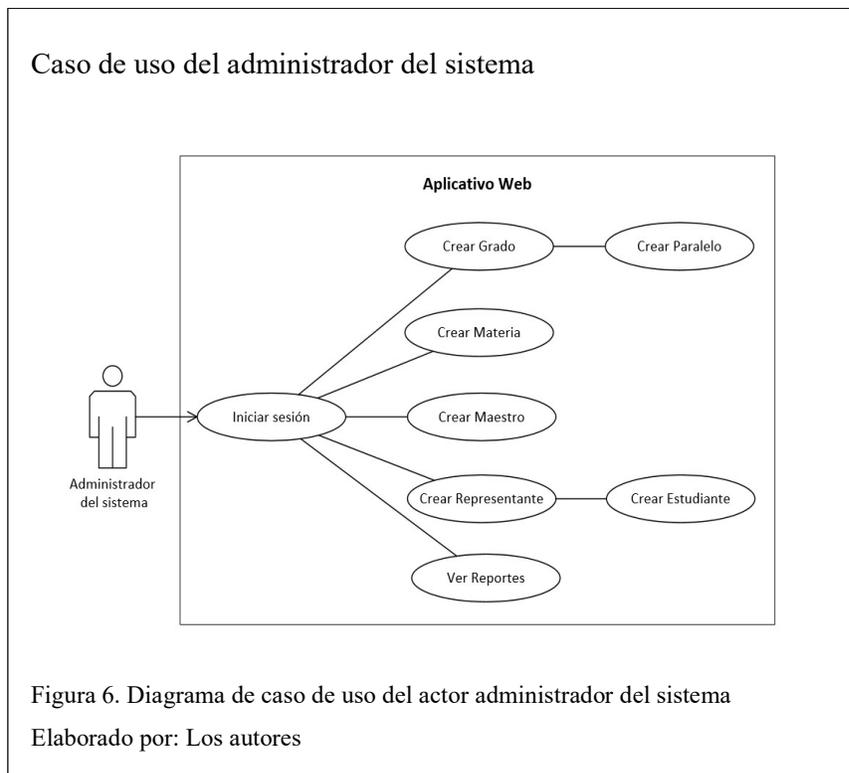
Para identificar los usuarios o actores, roles, acciones que se pueden ejecutar, todo enmarcado en la funcionalidad del sistema se usa la técnica de modelado de casos de uso basado en UML (Unified Modeling Language). De esta forma se documenta

las principales partes del sistema desde el punto de vista de los perfiles que lo usan. Gracias al levantamiento de las historias de usuario y la definición del proceso que el sistema permite ejecutar, son fácilmente comprensibles los roles de los actores, los mismos que se visualizan a continuación.

### 2.6.1 Caso de uso del Administrador del Sistema

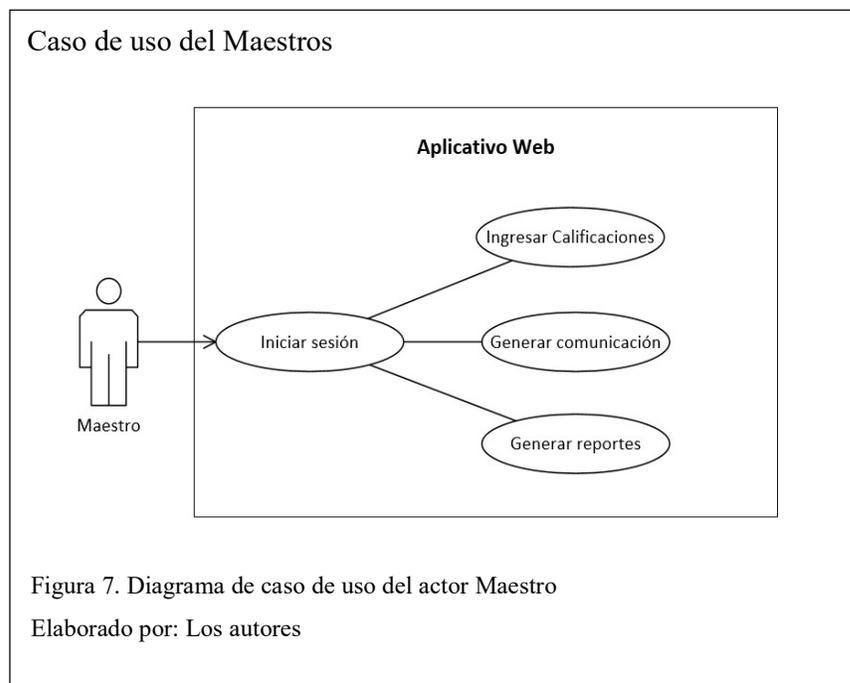
El usuario con perfil Administrador inicia sesión para poder realizar varias acciones orientadas a principalmente a parametrizar el sistema y brindar la funcionalidad que el resto de usuarios del sistema requieren.

Las posibles acciones de este actor se pueden graficar mediante el diagrama de los casos de uso para el usuario con perfil administrador, tal como se puede observar en la figura 6.



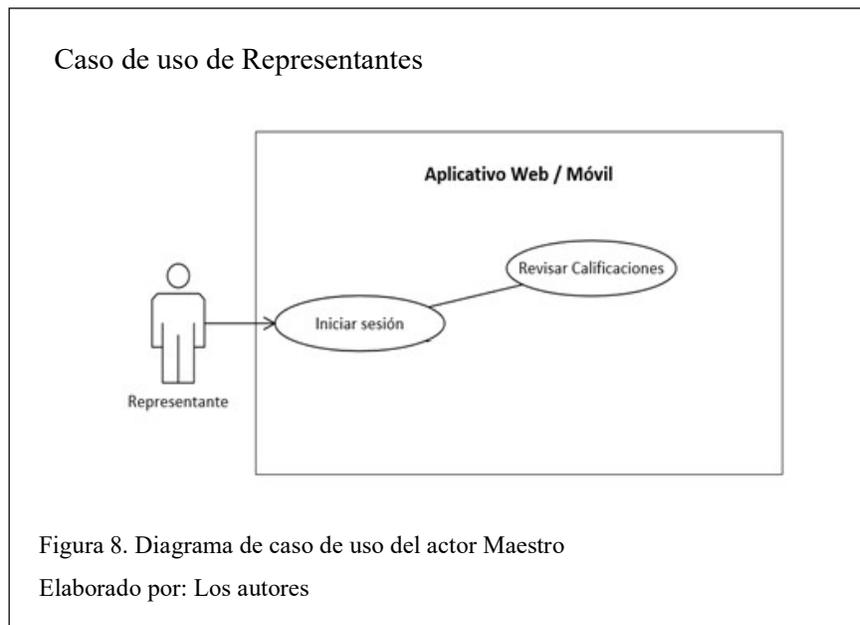
## 2.6.2 Caso de uso de Maestros

El usuario con perfil tipo maestro debe iniciar sesión y tras ello puede ingresar y consultar las calificaciones de los estudiantes según la materia, grado y paralelo. Además, puede generar reportes de las calificaciones ingresadas y entre las acciones también se encuentra la creación de mensajes para establecer comunicación con los representantes de los estudiantes, o contestar mensajes que los representantes hayan generado previamente, todo esto se puede observar de manera gráfica en el diagrama de casos de uso de la figura 7.



### 2.6.3 Caso de uso de Representantes

El usuario con perfil tipo Representante debe iniciar sesión y tras ello, tanto en la aplicación web como en la aplicación móvil para celulares puede revisar las calificaciones de sus representados. Estas acciones se representan gráficamente en el diagrama de caso de uso para el perfil de tipo representante de la figura 8.

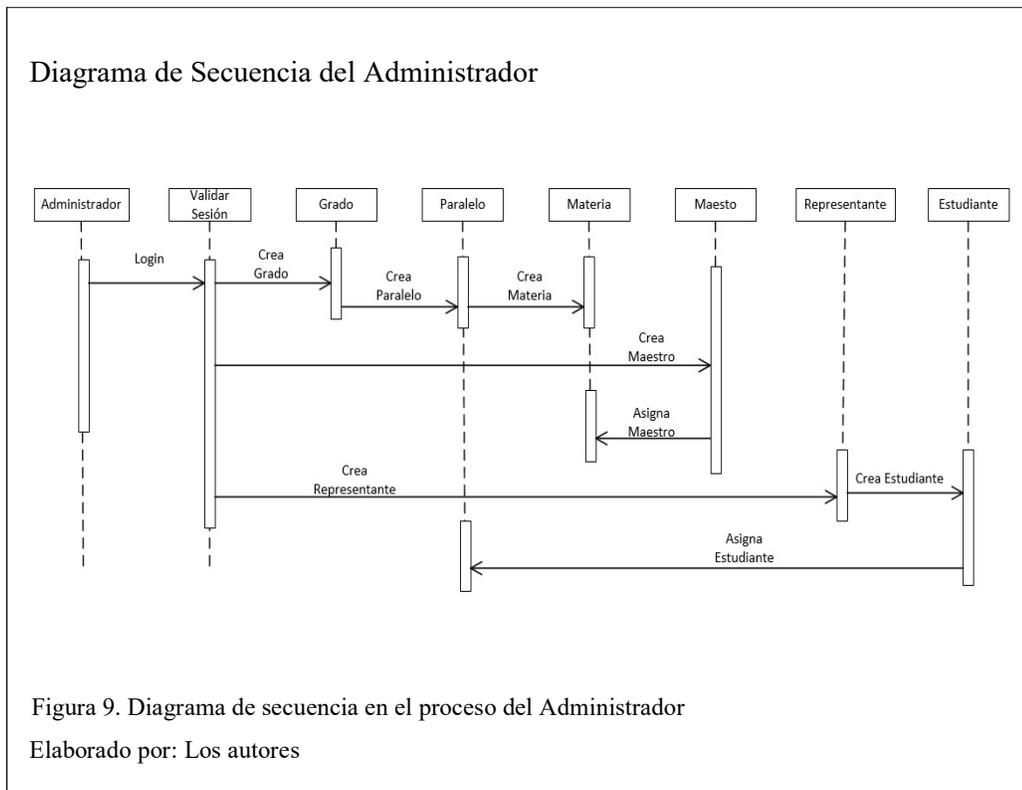


### 2.7 Diagramas de Secuencia

Los diagramas de secuencia permiten ver la forma en que los eventos provocan las transiciones entre los objetos del sistema durante su periodo de vida o mientras dura el proceso. Se muestra un diagrama de secuencia por cada caso de uso presentado anteriormente, es decir para cada actor del sistema; Administrador, Representante y Maestro.

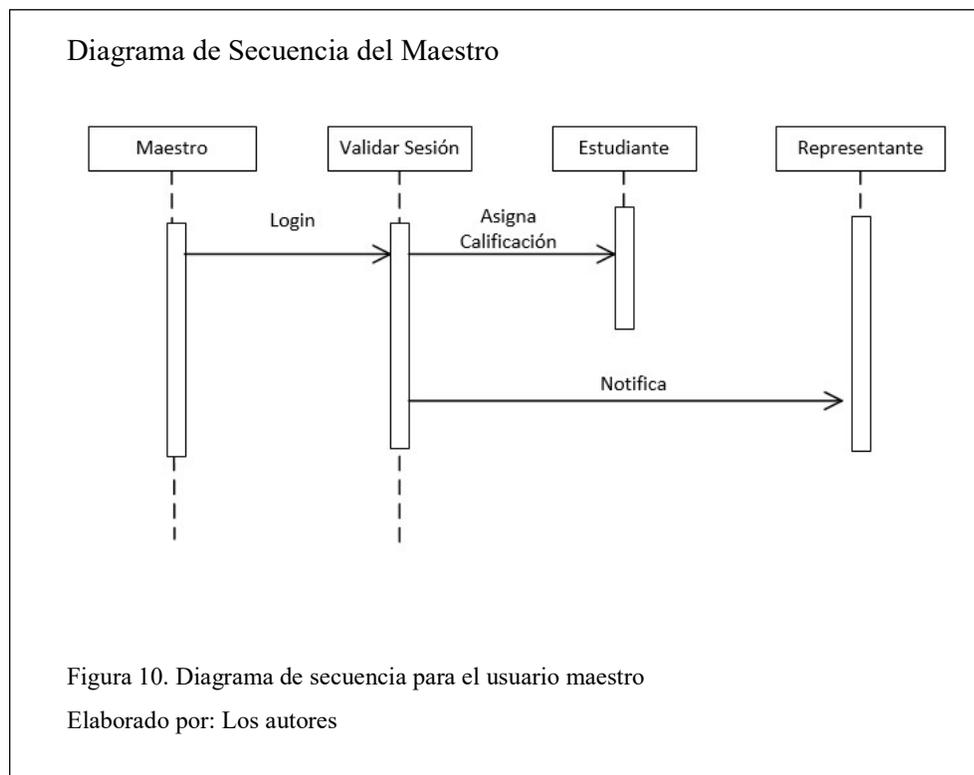
### 2.7.1 Diagrama de Secuencia del Administrador

El diagrama de la figura 9 muestra cómo se crean los distintos parámetros que están a cargo del usuario con perfil administrador y que permiten la funcionalidad del sistema y la dependencia que existe entre ellos. Las líneas horizontales muestran el orden o secuencia para la configuración de la información que es imprescindible. El gráfico indica que el Administrador del sistema debe generar los grados, para luego generar los paralelos para cada grado, después es necesario crear las materias y asignarlas a cada grado, posteriormente se ingresan a los maestros y se les asigna materias. Finalmente, como parte del proceso de matriculación de estudiantes, el administrador debe ingresar la información de representantes y de estudiantes y relacionarlos según corresponda.



### 2.7.2 Diagrama de Secuencia del Maestro

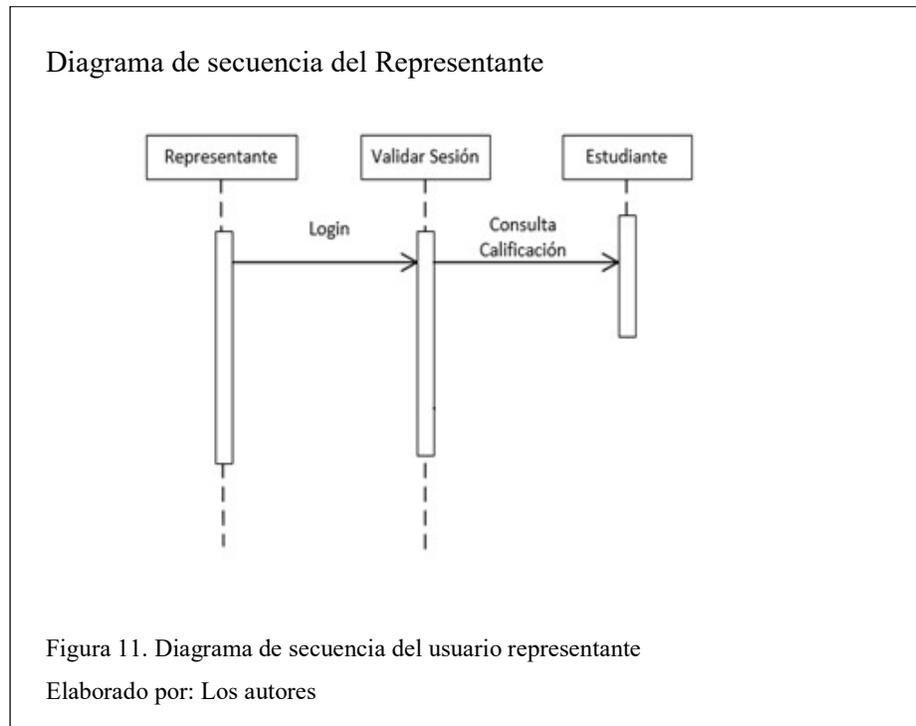
El diagrama de la figura 10 muestra cómo se comportan los objetos en el ámbito del usuario del tipo maestro, cuya principal tarea es el registro de calificaciones de los estudiantes que tiene a cargo según las materias que dicta y la eventual comunicación con uno o con varios representantes.



### 2.7.3 Diagrama de Secuencia del Representante

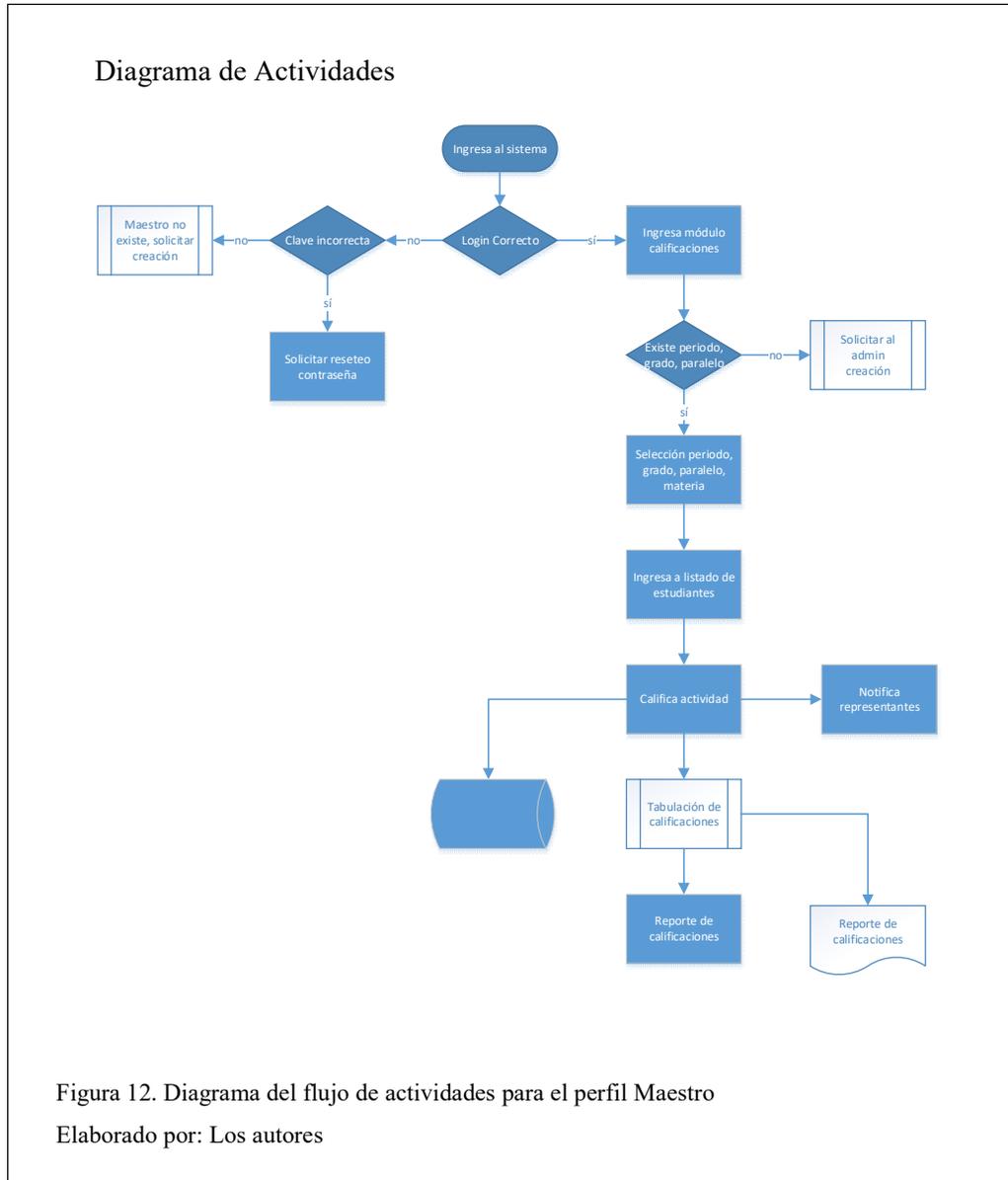
El diagrama de la figura 11 muestra cómo se comportan los objetos en el ámbito del usuario del tipo representante y en el módulo para representantes. Como se puede ver el margen de acción del representante es corta y se reduce al inicio de sesión o paso

de autenticación, la selección del estudiante y la consulta de las calificaciones obtenidas por su(s) representado(s).



#### 2.7.4 Diagrama de Actividades

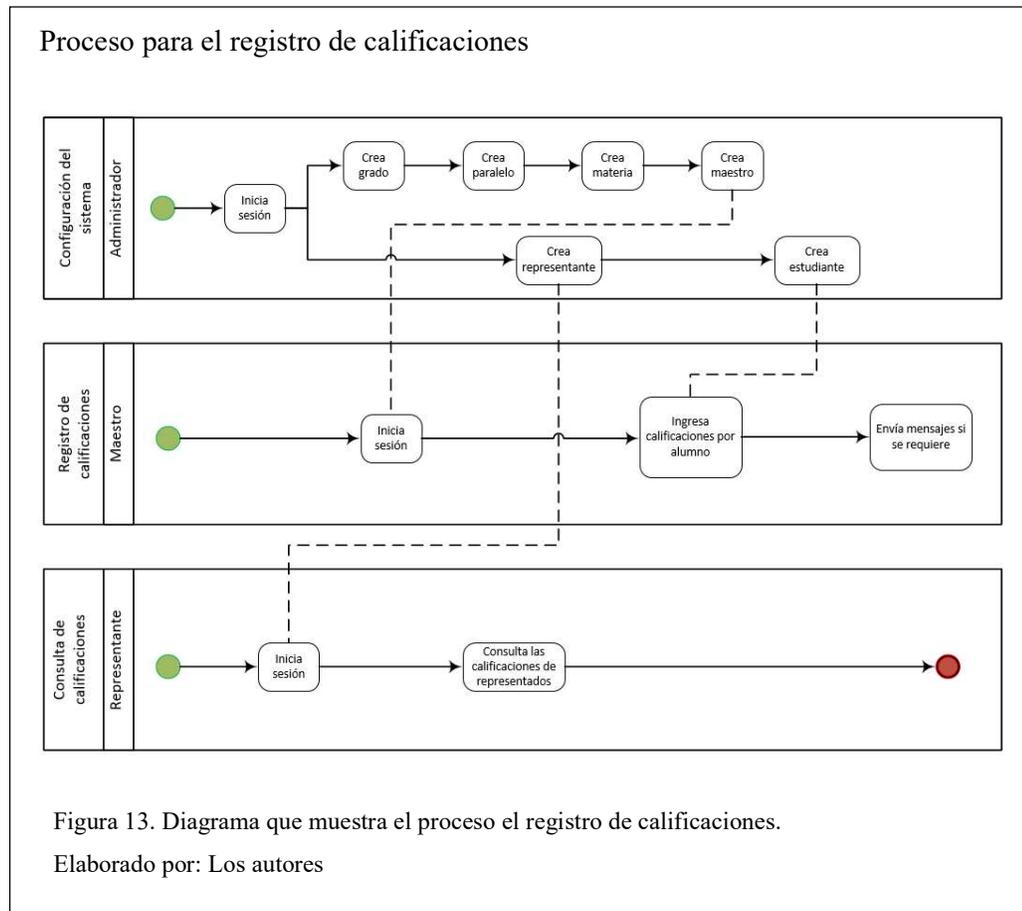
El diagrama de la figura 12 permite visualizar las principales acciones del sistema modelado en el proceso que sigue un maestro para el registro de calificaciones de los estudiantes. Se considera las actividades de autenticación o login, la validación de la existencia de los parámetros que permiten especificar a qué estudiante o grupo de estudiantes se va a calificar y luego se mapea la acción de tabulación de las notas, y el almacenamiento de toda la información ordenada y estructurada para finalmente registrar la actividad que genera una eventual comunicación del maestro con los representantes.



## 2.8 Proceso de registro de calificaciones

Mientras que las figuras 7 y 10 muestran de forma gráfica el proceso que se pretende implementar para la funcionalidad principal del sistema y de su objetivo que es el registro de calificaciones por parte de los profesores, la figura 11 representa la consulta de las calificaciones por parte de los representantes. El proceso unificado de

las acciones de estos dos usuarios se muestra como un solo proceso “TO-BE” y se describe gráficamente en la figura 13.



## 2.9 Diagrama de Datos

Para el diagrama de datos se usa un modelo entidad-relación que muestra la estructura de la base de datos mediante la representación de las entidades, los atributos de las mismas y cómo se relacionan entre ellas, se aprecian en la figura 14. En el esquema se pueden observar las tablas componentes de la base de datos del sistema que luego son usadas tanto para la aplicación web y móvil.

## Diagrama de Datos. Modelo Entidad - Relación

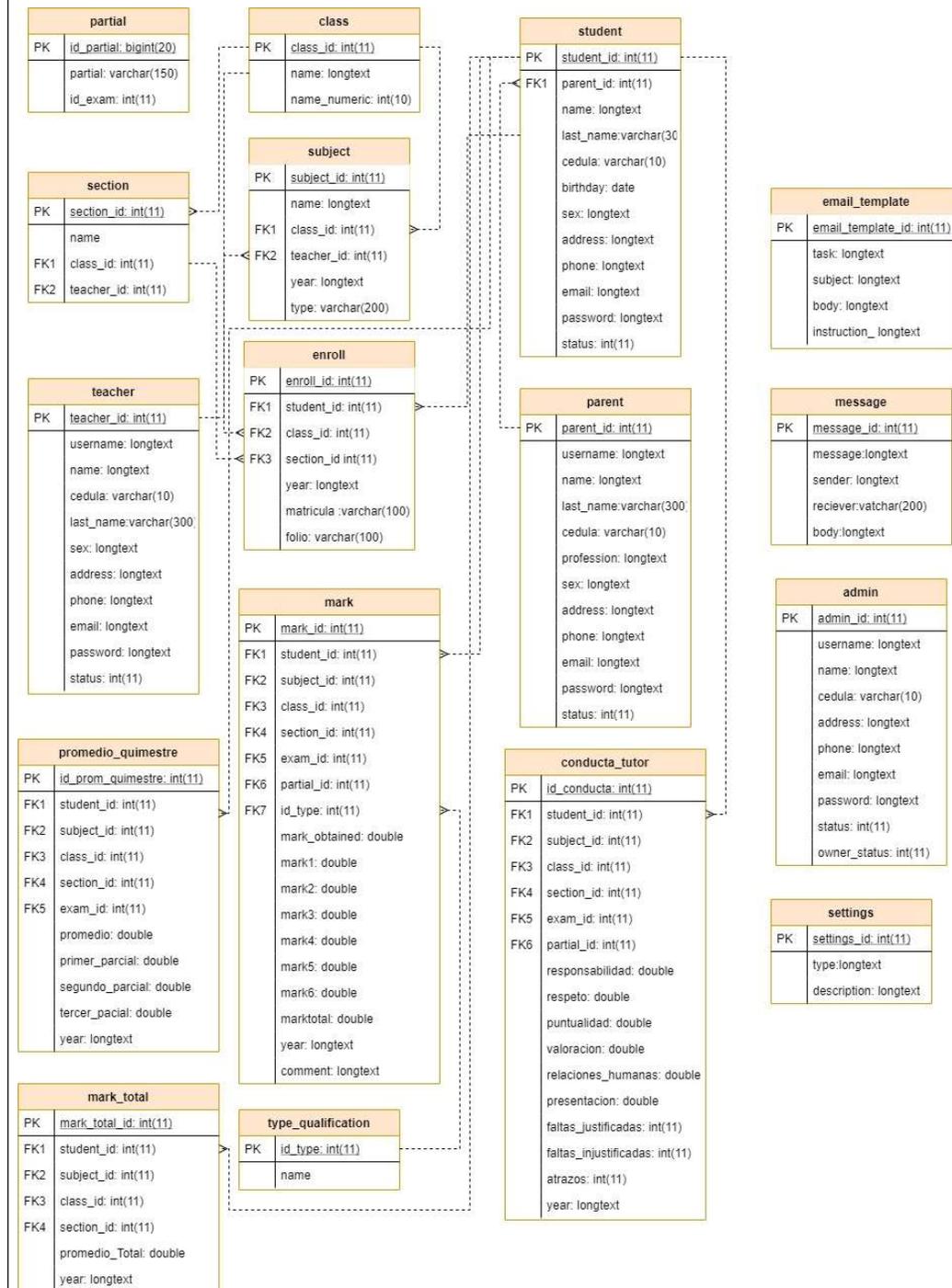
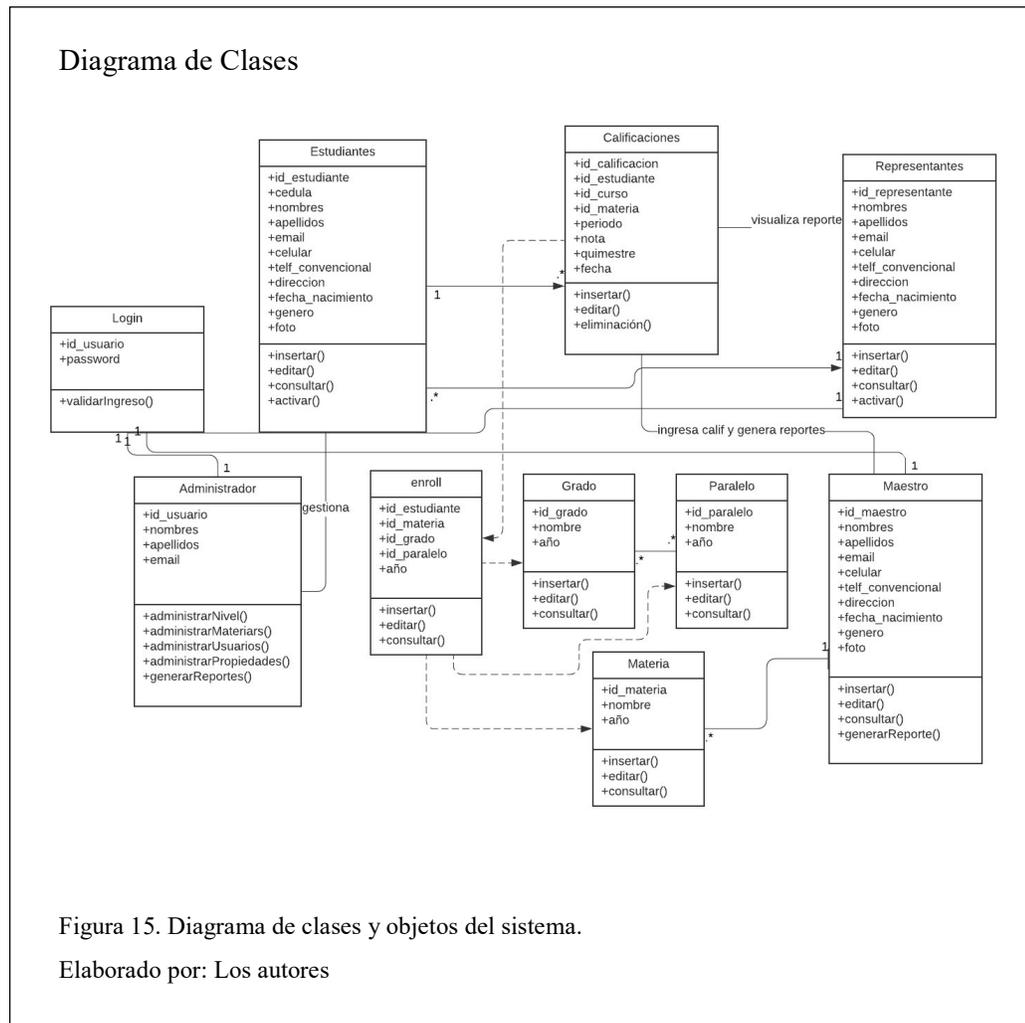


Figura 14. Diagrama del Modelo Entidad-Relación de la Base de Datos.

Elaborado por: Los autores

## 2.10 Diagrama de Clases

El diagrama de clases de la figura 15 representa la estructura y el comportamiento de los objetos del sistema y la relación con los otros objetos. La clase es la agrupación del conjunto de objetos con los mismos atributos. Como se puede apreciar éstos son principalmente los actores del sistema y los parámetros que permiten la funcionalidad del mismo.



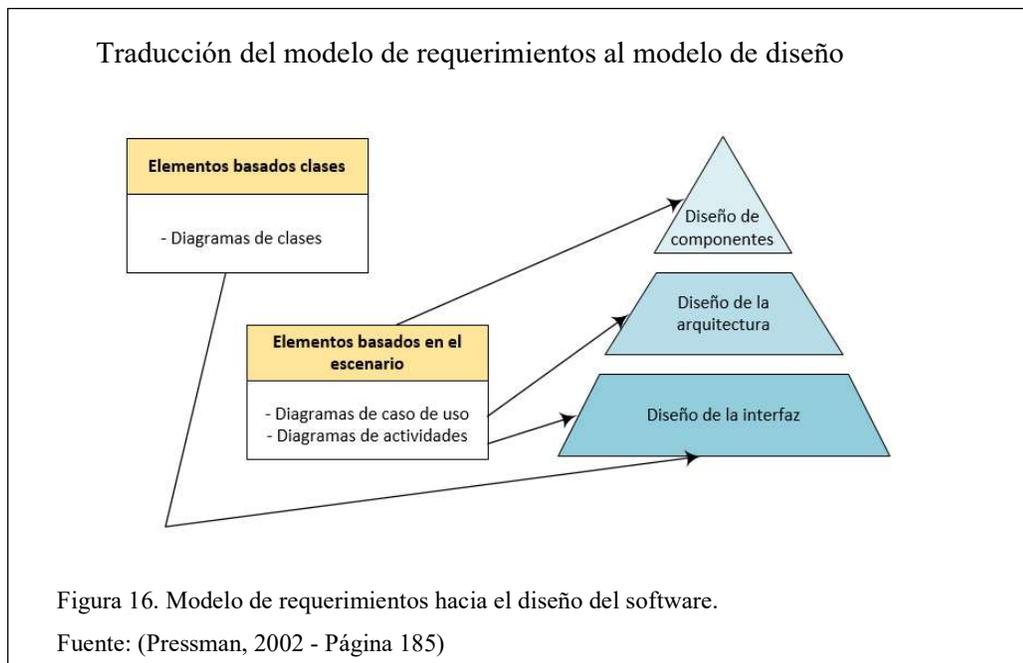
## CAPÍTULO 3

### Diseño del Sistema

#### 3.1 Diseño de la Arquitectura del Sistema

Desde el punto de vista de la ingeniería, el diseño explícito es una representación del modelo que tiene el software y del resto de elementos que le permiten su funcionamiento, el diseño además proporciona detalles sobre la arquitectura del sistema, la relación entre los componentes, la estructura y organización de los datos y de las interfaces o propiedades externas visibles para la interacción con el usuario.

La información obtenida en todo el capítulo anterior, permite trasladar el modelado de requerimientos a un modelo de diseño del software, la figura 16 ilustra cómo se transfieren datos importantes; desde el modelo de requerimientos se genera información necesaria que alimenta el trabajo de diseño.



El diseño del software suele variar dependiendo de la metodología que se usa en el desarrollo, en este caso se usan diseños con modelos estáticos, con diagramas de paquetes y componentes, que a su vez generan como entregables; representaciones arquitectónicas, interfaces en el nivel de componente e interfaces en el nivel de despliegue.

### **3.2 Diagramas de Paquetes**

Como se ha señalado anteriormente la arquitectura del sistema está basada en el patrón Modelo Vista Controlador, por lo tanto, es necesario identificar gráficamente la distribución del sistema en un diagrama de paquetes. En la figura 17 se puede apreciar la separación de los 3 principales paquetes, que también podrían ser consideradas las capas en el patrón MVC.

Es el usuario quien inicia el proceso mediante una solicitud HTTP, es decir al ingresar a la URL del sistema, la llamada la recibe el Controlador que realiza validaciones iniciales y llama a la Vista correspondiente, cuando el usuario realiza las diferentes tareas, éstas se validan y se identifican en el Controlador que realiza peticiones al Modelo si se requiere o devuelve la información a la vista para mostrar la información en pantalla.

Diagrama de Paquetes con el patrón MVC

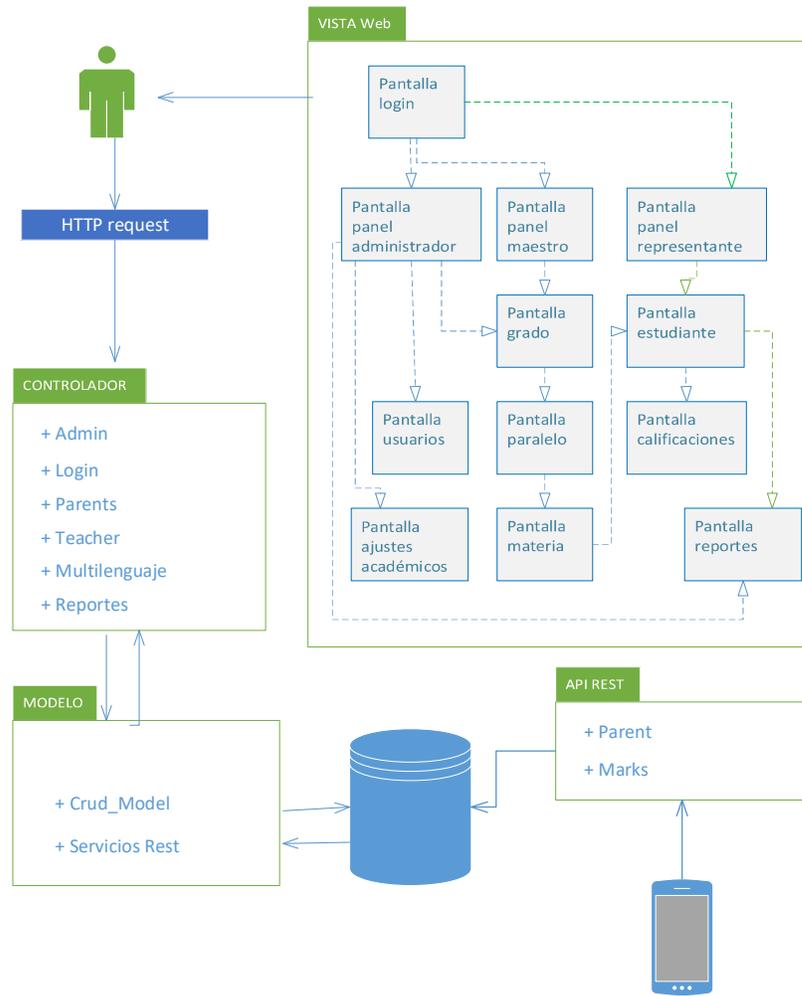


Figura 17. Diagrama de Paquetes organizados según el patrón MVC.

Elaborado por: Los autores

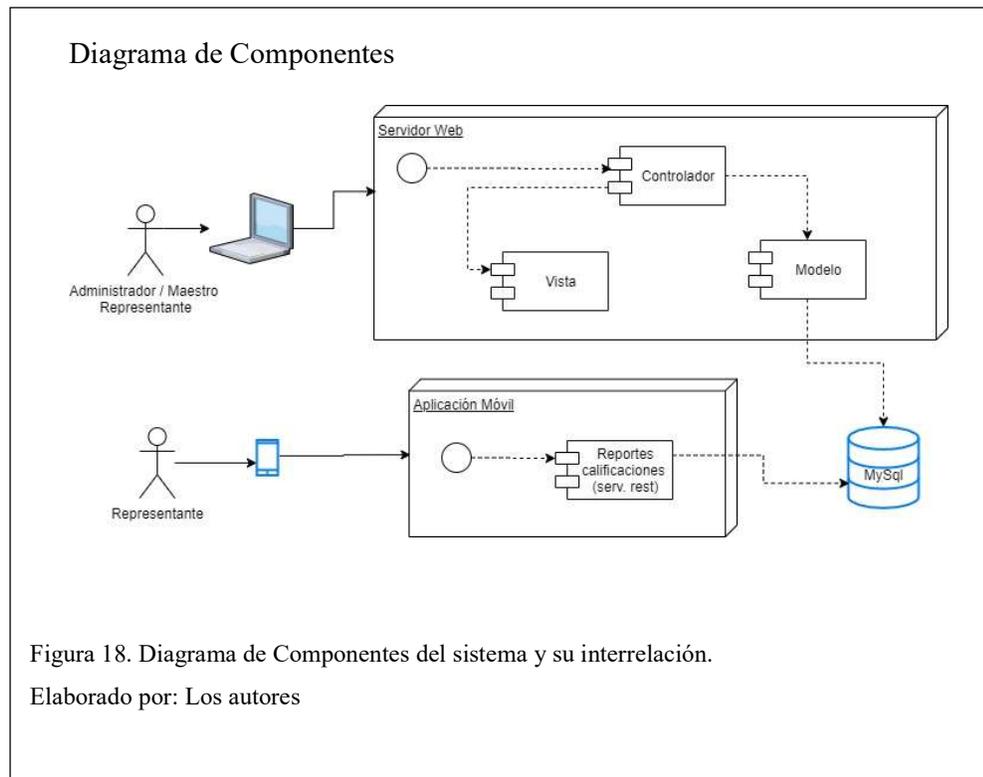
### 3.3 Diagrama de Componentes

El siguiente diagrama describe la manera en cómo se encuentran organizados los distintos componentes del sistema. El sistema web y la base de datos están alojados

en un host que almacena la información publicada gracias a un dominio previamente adquirido y coordinado con la UEAS. Los usuarios (Administrador, Maestro y Representante) ingresan al sistema a través de esta URL por medio de un navegador.

Internamente se tienen los componentes de autenticación, administración y gestión de calificaciones que son utilizados tanto por el componente sistema web como por el componente aplicación web. Finalmente se cuenta con el componente de acceso al motor de base de datos y la gestión de la información en la base de datos.

En la figura 18 se puede apreciar el diagrama con los componentes señalados y cómo se relacionan entre ellos, es importante señalar que la aplicación móvil es parte de otro componente, así como los servicios que utiliza para obtener la información de los otros componentes del sistema.

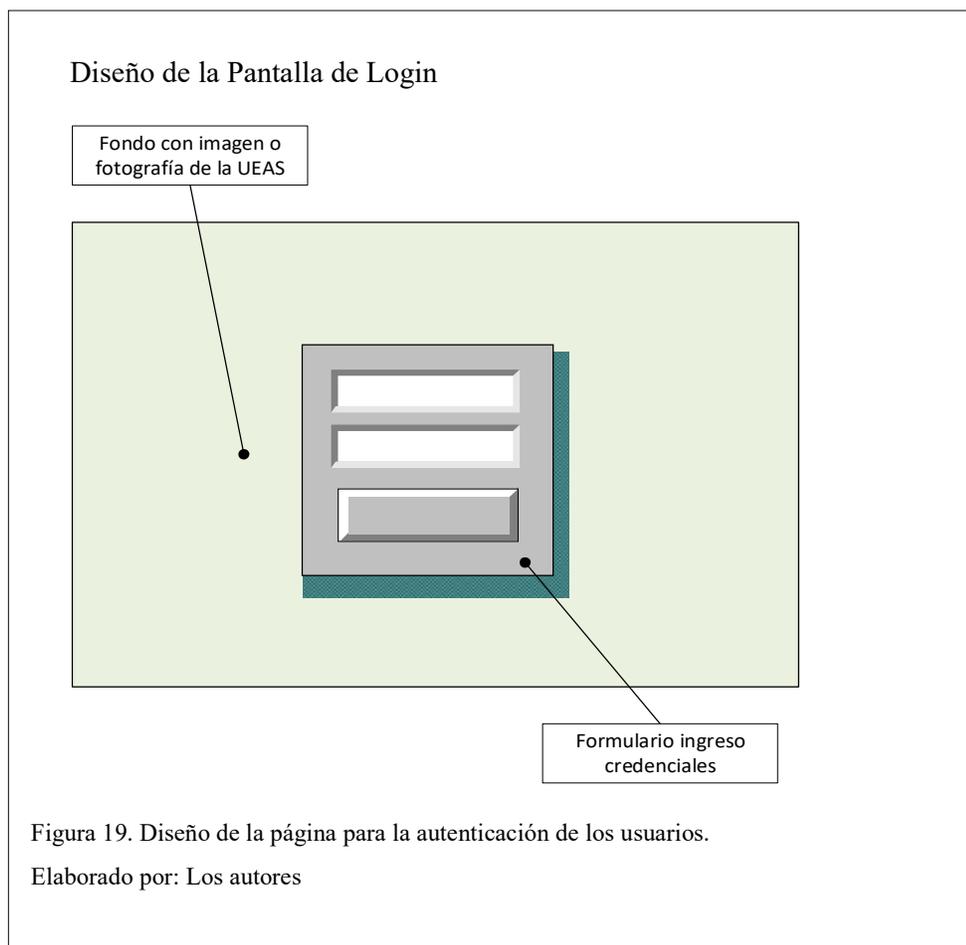


### 3.4 Diseño de Interfaces

El diseño de las interfaces muestra la estructura general de las pantallas tanto para la Web como para la aplicación móvil, la distribución de la información en las páginas y las principales características de diseño que tiene el sistema.

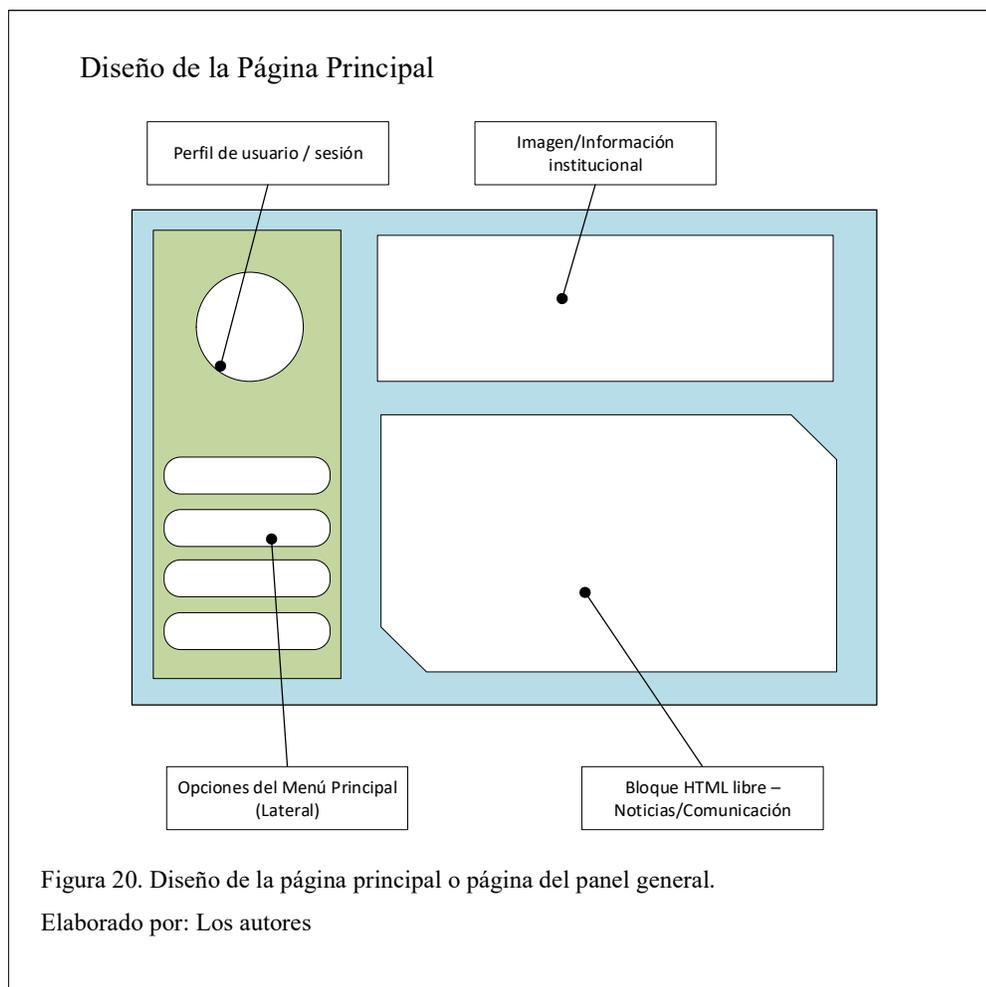
#### 3.4.1 Diseño de la Pantalla de Login

Es la primera pantalla del sistema y únicamente contiene el formulario de acceso, por ello se establece como diseño una imagen de fondo y la caja para el ingreso de datos.



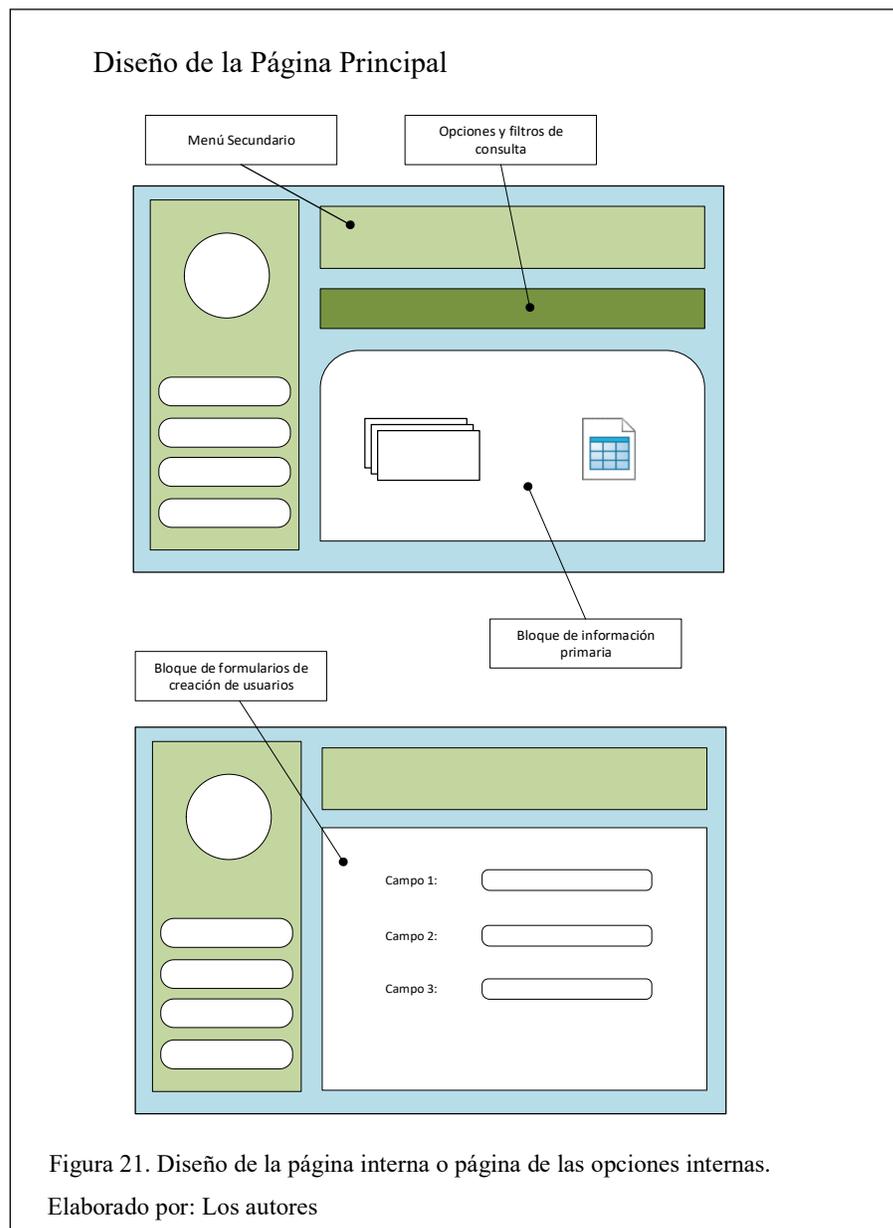
### 3.4.2 Diseño de la Pantalla Principal

Esta pantalla es común para todos los usuarios, acceden a ella después del proceso de autenticación y les permite navegar hacia cada una de las opciones habilitadas mediante el menú principal en el costado izquierdo, en esta caja se muestra el nombre del usuario y la opción para salir del sistema. Esta página en su lado derecho contiene dos bloques, uno que corresponde a la cabecera con información institucional y otro que tiene espacio libre para publicación de información relevante.



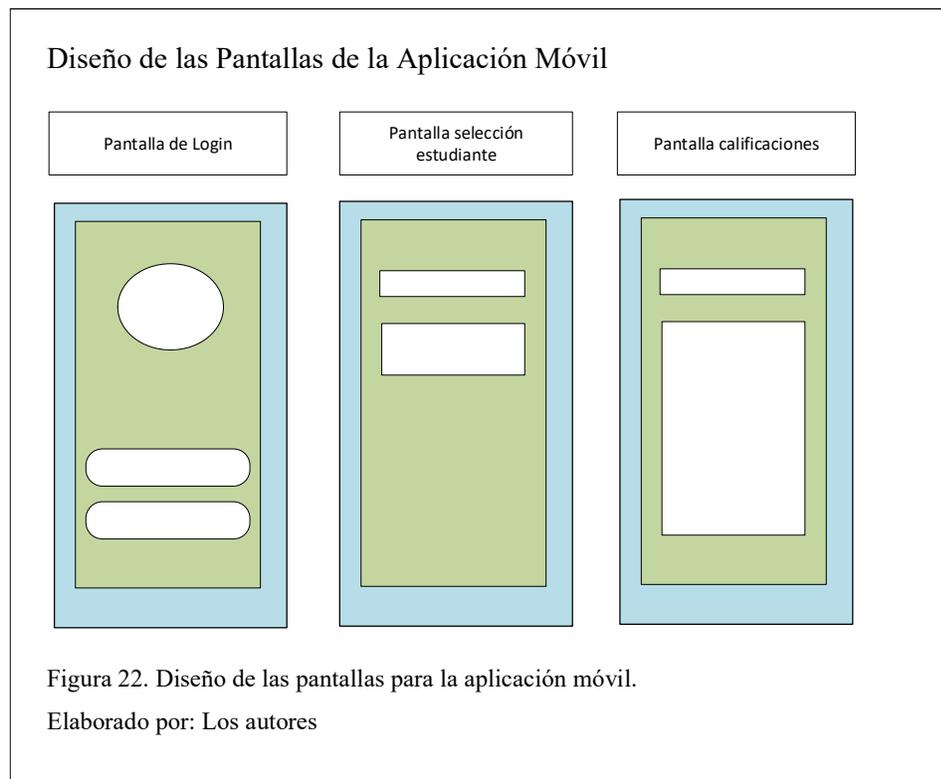
### 3.4.3 Diseño de Pantalla Interna

La pantalla interna es la base de la navegación en las opciones que se eligen en el menú principal. La página mantiene el menú lateral y tiene un nuevo menú en la parte superior y que podría extenderse en un bloque adicional inferior de filtros o parámetros de selección, cuenta con un bloque principal para la información primaria.



### 3.4.4 Diseño de Pantallas de la Aplicación Móvil

Como se ha mencionado anteriormente la aplicación móvil sirve para que los representantes puedan acceder a las calificaciones de los alumnos, en ese sentido las pantallas no contienen demasiada información y por lo tanto siguen un patrón de diseño simple pero intuitivo y ordenado como se puede ver en la figura 22.



### 3.5 Elementos de diseño

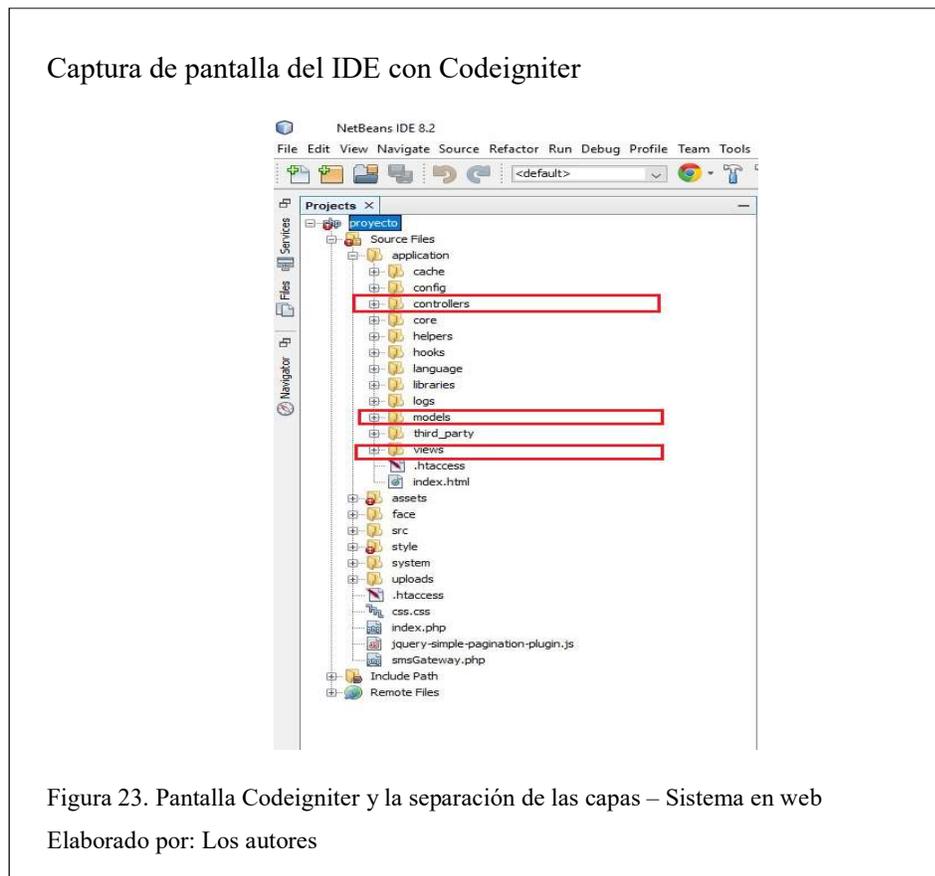
La Unidad Educativa Autogestionaria Solidaridad no cuenta con una guía de estilos de marca ni tampoco con un manual de diseño basado en identidad por lo cual no existen parámetros obligatorios en el marco entorno visual que se deban cumplir.

## CAPÍTULO 4

### Desarrollo del Sistema

#### 4.1 Configuración del IDE

Como se ha revisado anteriormente, para la implementación del proyecto para el entorno web, se usa el framework de desarrollo Codeigniter, el mismo que, como principal atributo genera la estructura del modelo MVC. Tras la instalación de la herramienta, y la configuración del ambiente en el IDE (Entorno de Desarrollo Integrado), en este caso, Netbeans 8.2, se pueden identificar la separación de las capas correspondientes por directorios y es manteniendo esta organización que se desarrollan las funcionalidades del sistema. (Figura 23).



Para el desarrollo de las pantallas de la aplicación móvil, de los servicios y de los métodos que permiten su funcionamiento se utiliza el IDE Visual Studio Code, en el cual se separan dos proyectos, el primero para la gestión y desarrollo de aplicación propiamente dicha y el segundo para la configuración y desarrollo de los servicios Rest. Se utiliza este IDE pues tienen incorporado una consola que permite ejecutar sentencias de instalación y configuración de las herramientas y módulos que se requieren en el desarrollo usando Node.js.

## **4.2 Plantilla Bootstrap**

En el desarrollo del proyecto, para la parte web, se recurre al uso de elementos pre-configurados en forma de plantillas de Bootstrap para fortalecer la usabilidad del aplicativo y que a la vez permita al usuario navegar de forma intuitiva a lo largo de las distintas opciones.

## **4.3 Clases del Sistema**

A continuación, se listan las clases que conforman el sistema y que contienen los métodos que permiten la funcionalidad del mismo. En el listado de la tabla no se incluye la descripción de las clases que son parte del framework Codeigniter, ni de las clases asociadas a Node.js, sino únicamente aquellas que se programan en el marco del desarrollo de este proyecto, tanto para el entorno web como para el entorno de la aplicación móvil.

Como se ha visto anteriormente las clases se desarrollan, en su mayoría, en función de los distintos usuarios del sistema, aunque en para la aplicación móvil también se organizan de acuerdo a las principales funcionalidades.

**Tabla 3.** Lista de las clases del sistema

<b>Nombre de la Clase</b>	<b>Ubicación</b>	<b>Descripción</b>
Admin	Aplicación Web	Esta clase permite la gestión de la información que tiene como responsabilidad el Administrador del sistema, es decir la creación de los usuarios, de los grados y paralelos, de las materias, la asignación de materias a profesores, la asignación de estudiantes a los grados y la configuración del cierre de año lectivo.
Teacher	Aplicación Web	Esta clase permite la gestión de la información que está a cargo de un maestro, principalmente el ingreso y asignación de calificaciones, la generación de reportes y la comunicación con los representantes.
Parent	Aplicación Web	Esta clase permite la consulta de calificaciones por parte de los representantes y la comunicación o el envío de mensajes a los maestros.
Login	Aplicación Web	Esta clase permite el proceso de autenticación de los usuarios, es decir la verificación de los datos ingresados por el usuario previo al acceso al resto del sistema.
Parent	Aplicación Móvil	Esta clase contiene las funciones para controlar el acceso del usuario representante desde la aplicación móvil.
Student	Aplicación Móvil	Es la clase en la que se encuentran las funciones de consulta de los estudiantes que podría tener un representante y otros métodos para establecer formatos adecuados en la información.

**Nota:** Esta tabla contiene el nombre de las clases del sistema y una descripción de las mismas.

#### **4.4 Métodos del Sistema**

En la tabla 4 se lista el nombre de los métodos que permiten la funcionalidad requerida en el sistema también se incluye el nombre de la clase a la que pertenecen y

la descripción o la función que cumplen. No se incluyen los servicios API REST puesto que se observan en el siguiente apartado.

**Tabla 4.** Lista de los métodos del sistema

Parte del sistema	Método	Es parte de la clase / servicio	Descripción
Web	auth	Login	Realiza la validación de contraseña correcta para el usuario que se ha enviado como parámetro
Web	student	Admin	Este método permite la inserción o edición de estudiantes. Recibe dos argumentos, en el primero se indica si se trata de una inserción o si es una actualización y en el segundo se recibe el id del estudiante cuando se trata de una actualización. El resto de variables que permiten la funcionalidad del método se adquieren mediante POST
Web	parents	Admin	El método inserta o edita la información de un representante. Recibe dos argumentos, en el primero se indica si se trata de una inserción o si es una actualización y en el segundo se recibe el id del representante cuando se trata de una actualización. El resto de variables que permiten la funcionalidad del método se adquieren mediante POST
Web	teachers	Admin	El método inserta o edita la información de un maestro. Recibe dos argumentos, en el primero se indica si se trata de una inserción o si es una actualización y en el segundo se recibe el id del profesor cuando se trata de una actualización. El resto de variables que permiten la funcionalidad del método se adquieren mediante POST

Web	level	Teacher	Permite la inserción o edición de un grado
Web	class	Teacher	Permite la inserción o edición de un paralelo
Web	subject	Teacher	Permite la inserción y edición de una materia
Web	marks_update	Teacher	Permite el registro de calificaciones obtenidas por los alumnos en las distintas actividades que conforman un tipo de actividad y el registro del promedio por tipo de actividad
Web	marks_update_test	Teacher	Permite el registro de la calificación obtenida en la prueba del parcial y el registro del promedio para cada parcial, además el registro del promedio de los quimestres y el promedio total
Web	generar_excel_est	Admin	Genera un archivo Excel con el listado de estudiantes por grado y paralelo, su identificación de matrícula y folio
Web	generar_excel_hist	Admin	Genera un archivo Excel con el resumen de las calificaciones obtenidas por un estudiante, cumple con el formato que establece el Min. De Educación
Web	system_settings	Admin	Permite el registro y edición de la información (datos de contacto) de la escuela
Web	send_mail	Admin	Permite el envío de correos electrónicos de forma masiva a una lista de usuarios.
Web	marks	Parent	Permite obtener las calificaciones obtenidas por el estudiante seleccionado por el representante
App	validateCredenciales	Login	Es uno método del servicio Login, se encarga de verificar que las credenciales ingresadas correspondan al usuario
App	formatStudents	Parent	Obtiene los estudiantes asignados al representante autenticado
App	getMarksBySubject	Student	Obtiene las calificaciones por materia obtenida en el parcial correspondiente

App	formatMarksAverage	Student	Obtiene el promedio final de las calificaciones quimestrales
-----	--------------------	---------	--

**Nota:** Esta tabla contiene el nombre de los métodos del sistema, su clase y su descripción funcional.

#### 4.5 Servicios API Rest del Sistema

Como consta en el apartado del Marco Teórico, en el proyecto se recurre al uso de servicios web y para ello se utiliza la tecnología REST. A continuación, se detallan los servicios del proyecto, como se ve en la descripción de cada uno, éstos permiten la conexión y comunicación con la Base de Datos para extraer la información que es utilizada para el proceso de autenticación del usuario y para obtener las calificaciones de los estudiantes.

**Tabla 5.** Lista de los servicios Api Rest del sistema

Nombre del Servicio	Ubicación	Descripción
login	Aplicación Web	La aplicación consume este servicio para obtener la confirmación de que las credenciales enviadas son correctas y permitir al usuario loguearse e ingresar al sistema.
parent	Aplicación Web	Este servicio obtiene el listado de estudiantes para los cuales el usuario autenticado es su representante
student	Aplicación Web	Este servicio permite obtener las calificaciones registrada para los estudiantes seleccionados en la pantalla de la aplicación, según el parcial y quimestre.

**Nota:** Esta tabla contiene el nombre de los servicios que consume la aplicación móvil del sistema.

## 4.6 Codificación

Gracias al uso de Codeigniter como marco de desarrollo de la parte web del sistema, se cumplen ciertos lineamientos que pueden repetirse en varios archivos. Como ejemplo se presentan las primeras líneas de código de la clase “Students”, en la figura 24 se puede observar que existe una sentencia para comprobar si la variable BASEPATH está definida, esto sirve para verificar que el sistema se haya iniciado correctamente y evitar que el servidor ejecute cualquier archivo PHP si esto no ha ocurrido. Luego se observa que la clase, mediante el uso de “extends”, hereda de la clase CI\_Controller que es una clase propia del framework, y que le permite ser una clase de tipo controlador en el marco de la arquitectura MVC. Estas herencias se repiten en cada uno de los archivos de las clases controlador. Finalmente, en la figura también se puede observar la creación del método constructor, en este caso además se aprecia que en el constructor se carga e inicializa la clase de base de datos en función de sus ajustes de configuración y que se carga la librería “session” para el manejo y gestión de las sesiones en la navegación del sistema.

Captura de pantalla de la clase Student

```
<?php
if (!defined('BASEPATH'))
    exit('No direct script access allowed');

class Student extends CI_Controller
{
    function __construct()
    {
        parent::__construct();
        $this->load->database();
        $this->load->library('session');
    }
}
```

Figura 24. Código de inicio de la clase Student.  
Elaborado por: Los autores

La figura 25 muestra el código de uno de los métodos del sistema en su parte web, en este caso se trata del método ‘students’ el mismo que, como se describe en la tabla 3, sirve para añadir o editar la información de estudiantes. Como se puede observar, el método recibe dos parámetros; el primero recibe el tipo de acción que debe realizar el método; “create”, si se trata de un estudiante nuevo y “update” si se trata de la actualización de la información de un estudiante, el segundo parámetro que recibe el método es el id del estudiante en el caso de que se trate de una actualización. En el array “data” se asigna cada uno de los valores recibidos mediante post desde el formulario previo gracias al apoyo de la clase “input” de Codeigniter. Finalmente se hace uso de la clase “db”, que permite enviar a insertar la información según corresponda.

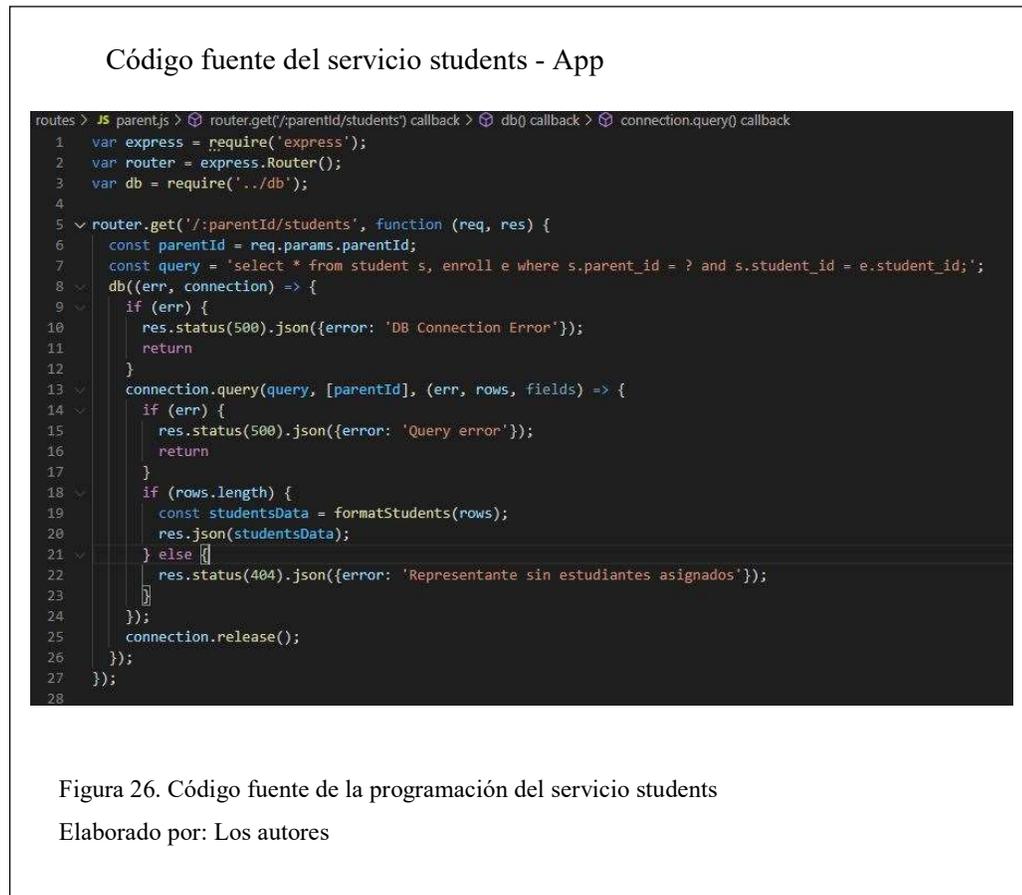
### Captura de pantalla de la función student

```
function student($param1 = '', $param2 = '')
{
    if ($this->session->userdata('admin_login') != 1)
    {
        redirect(base_url(), 'refresh');
    }
    $running_year = $this->db->get_where('settings', array('type' => 'running_year'))->row()->description;
    if ($param1 == 'create')
    {
        $data['cedula_identidad'] = $this->input->post('cedula');
        $data['last_name'] = $this->input->post('lastname');
        $data['name'] = $this->input->post('name');
        $data['username'] = $this->input->post('username');
        $data['birthday'] = $this->input->post('birthday');
        $data['date'] = strtotime(date("d M,Y"));
        $data['sex'] = $this->input->post('sex');
        $data['address'] = $this->input->post('address');
        $data['phone'] = $this->input->post('phone');
        $data['email'] = $this->input->post('email');
        $data['password'] = sha1($this->input->post('password'));
        $data['parent_id'] = $this->input->post('parent_id');
        $this->db->insert('student', $data);
        $student_id = $this->db->insert_id();
        $data2['student_id'] = $student_id;
        $data2['enroll_code'] = substr(md5(rand(0, 1000000)), 0, 7);
        $data2['class_id'] = $this->input->post('class_id');
        if ($this->input->post('section_id') != '')
        {
            $data2['section_id'] = $this->input->post('section_id');
        }
        $data2['roll'] = $this->input->post('cedula');
        $data2['date_added'] = strtotime(date("Y-m-d H:i:s"));
        $data2['year'] = $running_year;
        $this->db->insert('enroll', $data2);
        move_uploaded_file($_FILES['userfile']['tmp_name'], 'uploads/student_image/' . $student_id . '.jpg');
        $this->session->set_flashdata('flash_message', get_phrase('successfully_added'));
        redirect(base_url() . 'admin/add_student/', 'refresh');
    }
}
```

Figura 25. Código de la función student.

Elaborado por: Los autores

En la figura 26 se muestra uno de los servicios Api Rest desarrollados para la aplicación móvil del sistema, en este caso el servicio se encarga de realizar la consulta a la BDD para obtener los datos de los estudiantes que pertenecen a este representante.



Para el desarrollo de la aplicación móvil como se había mencionado, se utiliza Node JS, por lo cual la mayoría de sentencias en la programación son de tipo JavaScript, en cuanto a la lógica del sistema, mientras que para la implementación de la parte visual se utiliza el lenguaje de marcado de React Native, en dónde casi todos los elementos o etiquetas que se pueden encontrar para HTML tienen su equivalente, un ejemplo de esto se ve en la figura 27, en donde se muestra la estructura de la pantalla que recibe los datos para el proceso de autenticación.

## Código fuente de la pantalla de Login - App

```
app > screens > parent > JS AuthScreen.js > AuthScreen
30   return (
31     <TouchableWithoutFeedback onPress={() => {
32       Keyboard.dismiss();
33     }}>
34     <KeyboardAvoidingView behavior="padding" keyboardVerticalOffset={10} style={style.loginContainer}>
35       <View style={style.header}>
36         <View style={style.imageContainer}>
37           <Image style={style.image} source={require('../../assets/logo.png')}></Image>
38         </View>
39       </View>
40       <View style={style.formContainer}>
41         <Input
42           containerStyle={style.inputContainer}
43           inputContainerStyle={style.inputInputContainer}
44           leftIcon={
45             <Icon name='person-outline' type='material' color={Colors.primary} size={23}/>
46           }
47           placeholder="Usuario"
48           onChangeText={userHandler}
49           value={user}
50         />
51         <Input
52           containerStyle={style.inputContainer}
53           inputContainerStyle={style.inputInputContainer}
54           leftIcon={
55             <Icon name='lock-open' type='material' color={Colors.primary} size={23}/>
56           }
57           placeholder="Contraseña"
58           secureTextEntry={true}
59           onChangeText={passwordHandler}
60           value={password}
61         />

```

Figura 27. Código fuente de la pantalla para Login

Elaborado por: Los autores

Es importante mencionar que el uso de React Native genera una estructura del proyecto de la aplicación que consta de los archivos: Package.json en donde se encuentra la configuración básica de la aplicación y el archivo App.js como elemento principal de la estructura de esta parte del sistema. Por otro lado, también se hace uso de una librería llamada Expo, es relevante pues gracias a ella se puede exportar la aplicación a herramientas de ayuda como Android Studio la cual contiene emuladores que permiten pre-visualizar la aplicación en construcción.

Finalmente, gracias a que la aplicación se construye bajo el framework React Native, la misma puede enviarse a compilar y de esa manera se obtiene el SDK que

será el instalador de la aplicación para los dispositivos celulares ya sea mediante la publicación en la tienda Google Play o descargándola directamente en los teléfonos celulares.

#### 4.7 Perspectiva Visual de Aplicación Web

A continuación, se muestran algunas de las pantallas del sistema, se consideran las más importantes en los flujos de navegación de los usuarios.

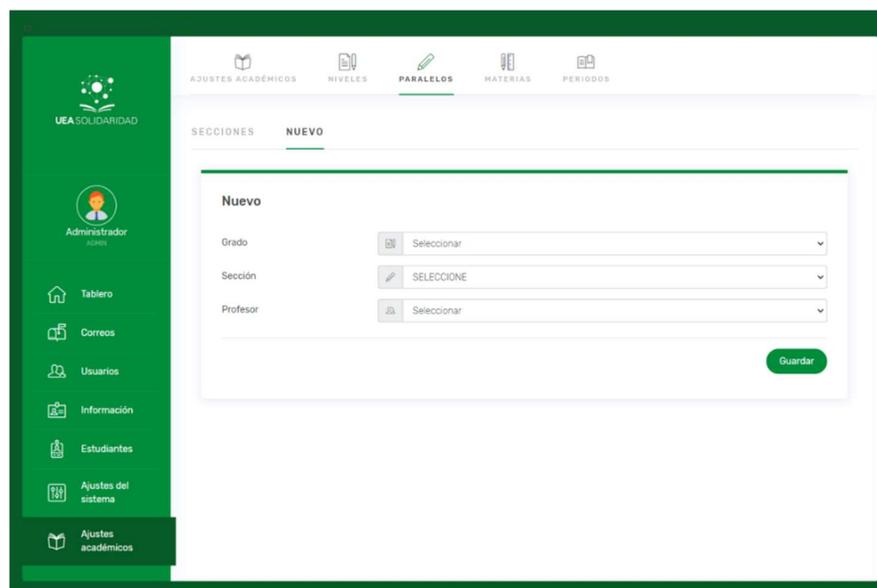
La figura 28 corresponde a la pantalla de Login, es decir la pantalla en dónde los usuarios se autentican ingresando su usuario y contraseña para poder acceder a las opciones que están disponibles según su perfil.



En la figura 29 se puede observar la pantalla para la creación de un paralelo, esta es una de las opciones del sistema para el perfil administrador a la que se accede

desde el menú izquierdo ingresando a “Ajustes académicos”. En esta opción del menú además se encuentra la configuración de la nota mínima para “aprobar” una materia, también constan el flujo para la creación de un grado o nivel, la creación de una materia, y la creación de parciales o períodos.

### Pantalla para la creación de un nuevo paralelo (administrador)



The screenshot shows a web interface for creating a new parallel class. On the left is a green sidebar with the UEA SOLIDARIDAD logo and a menu with options: Administrador (selected), Tablero, Correos, Usuarios, Información, Estudiantes, Ajustes del sistema, and Ajustes académicos. The top navigation bar includes: AJUSTES ACADÉMICOS, NIVELES, PARALELOS (active), MATERIAS, and PERIODOS. The main content area is titled 'SECCIONES' and 'NUEVO'. It contains a form with the following fields: 'Grado' with a dropdown menu showing 'Seleccionar'; 'Sección' with a dropdown menu showing 'SELECCIONE'; and 'Profesor' with a dropdown menu showing 'Seleccionar'. A green 'Guardar' button is located at the bottom right of the form.

Figura 29. Captura de pantalla de una opción del perfil administrador

Elaborado por: Los autores

En la figura 30 se presenta una de las pantallas más extensas del sistema, se trata de la pantalla en donde se ingresan los datos informativos del estudiante. Esta opción también es parte de los flujos habilitados para el perfil Administrador y a ésta se llega mediante el botón “Usuarios” del menú principal del lado izquierdo. En esta opción del menú además consta la pantalla para el ingreso de representantes, el cual es un paso previo para el registro de estudiantes, pues uno de los campos obligatorios es el ingreso del número de cédula del representante, es decir ya debe estar creado para

poder seleccionarlo, también consta la opción para ver la información, el ingreso o creación de maestros. En la opción de estudiantes se encuentra un submenú que permite seleccionar al usuario si quiere ver el listado de estudiantes o registrar uno nuevo.

### Pantalla para ingreso de estudiantes (administrador)

The screenshot displays the 'Formulario de registro' for students. The sidebar on the left contains the following menu items: 'Administrador', 'Tablero', 'Correos', 'Usuarios', 'Información', 'Estudiantes', 'Ajustes del sistema', and 'Ajustes académicos'. The main content area is titled 'Formulario de registro' and includes the following fields: 'Cédula de Identidad', 'Nombres', 'Apellidos', 'Usuario', 'Correo' (with the value 'user@school.edu'), 'Celular', 'Dirección', 'Cumpleaños' (with the value '06/29/2020'), 'Género' (with radio buttons for 'Masculino' and 'Femenino'), 'Representante' (with fields for 'Cedula Representante' and 'Nombre representante'), 'Contraseña', 'Grado' (with a 'Seleccionar' dropdown), 'Sección' (with a 'Seleccionar' dropdown), 'Numero De Matricula', 'Folio', and 'Fotografía'. A 'Guardar' button is located at the bottom of the form.

Figura 30. Captura de pantalla para el ingreso de estudiantes

Elaborado por: Los autores

La figura 31 muestra una de las pantallas habilitadas para el perfil maestro, se trata de la pantalla para el registro de las calificaciones obtenidas por los estudiantes.

Para ingresar las calificaciones el maestro debe seleccionar la siguiente información en ese orden: grado, quimestre, paralelo, materia, parcial y el tipo de actividad, se le presenta al maestro la matriz para que pueda registrar la nota para cada estudiante según corresponda.

### Pantalla para registro de calificaciones (maestro)

ESTUDIANTE	NOTA 1	NOTA 2	NOTA 3	NOTA 4	NOTA 5	NOTA 6	PROMEDIO	COMENTARIO
DEMIAN DIDIER CRUZ RODRIGUEZ	9	9	10	10	4	0	8.40	deje de hablar

Figura 31. Captura de pantalla para el registro de calificaciones

Elaborado por: Los autores

La figura 32 contiene la pantalla habilitada para el perfil representante. En ésta se publican las calificaciones obtenidas por los estudiantes de forma organizada y comprensible para que los representantes o padres de familia puedan consultarlo de forma periódica. Para llegar a esta pantalla el representante debe autenticarse y seleccionar en nombre del estudiante.

## Pantalla para la consulta de calificaciones (representante)

MATERIA	MAESTRO	PROMEDIO 1ER Q	PROMEDIO 2DO Q	PROMEDIO TOTAL	PROMOCIÓN
MATEMATICAS	ALEXANDER GORDILLO	4.53	3.46	4.53	NO APROBADO
SOCIALES	ALEXANDER GORDILLO	0.96	0.00	0.00	NO APLICA
PINTURA	MARGOTH TITO	4.03	0.00	0.00	NO APLICA

Figura 32. Captura de pantalla para la consulta de calificaciones

Elaborado por: Los autores

## 4.8 Perspectiva Visual de Aplicación Móvil

En la figura 33 se pueden apreciar las pantallas que conforman la navegación del representante en la aplicación móvil. La primera se trata de la pantalla en la que el usuario representante se autentica o realiza el proceso de login, la segunda es la pantalla en dónde el representante elige el alumno puesto que podría tener más de un representado en la institución, la tercera pantalla finalmente muestra las calificaciones del estudiante en un esquema de tipo acordeón. Una pantalla similar se muestra para la consulta de las calificaciones obtenidas en conducta o comportamiento.

## Pantallas para la consulta de calificaciones en móvil (representante)

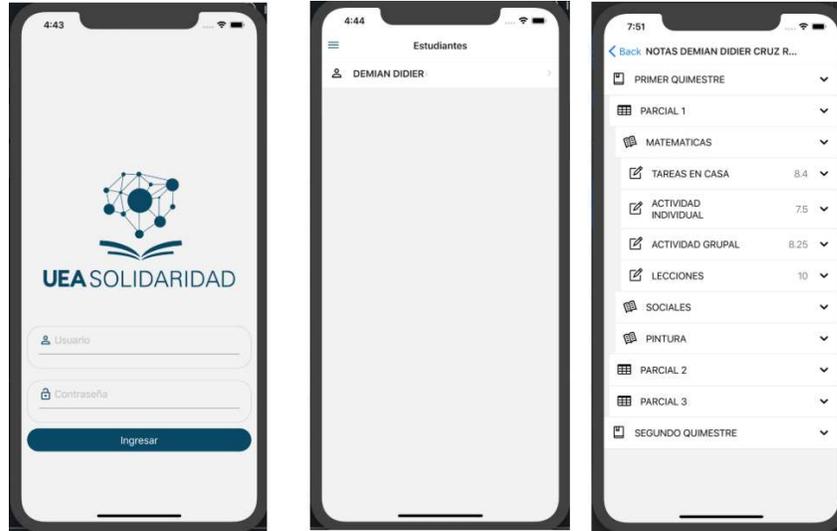


Figura 33. Capturas de pantalla para la consulta de calificaciones en app

Elaborado por: Los autores

## **CAPÍTULO 5**

### **Pruebas y Correcciones Realizadas**

#### **5.1 Pruebas del sistema**

Para asegurar que la aplicación web y móvil cumplan con los requerimientos identificados en el capítulo 3 se realizan pruebas de caja negra, pruebas funcionales y pruebas de carga. Dichas pruebas ayudan a lograr una mayor calidad del producto al cliente.

#### **5.2 Pruebas con técnica de caja negra**

Esta técnica comprueba la funcionalidad del sistema sin que sea relevante la estructura de código interna, sino más bien se basa totalmente en revisar las entradas y salidas esperadas que son levantadas en los requerimientos del cliente y/o del Product Owner.

En este proyecto se utilizan las siguientes técnicas:

- Pruebas de casos de uso (historias de usuario) o funcionales
- Partición de equivalencias
- Análisis de valores borde

##### **5.2.1 Pruebas funcionales (casos de uso / historias de usuario)**

En la siguiente tabla se describen las pruebas funcionales de cada uno de los actores desde el punto de vista del usuario final en sus diferentes roles.

**Tabla 6.** Esquema de las pruebas funcionales para los casos de uso

MÓDULO / ACTOR	REQUERIMIENTO	PROCESO	RESULTADO ESPERADO	FECHA (RESULTADO OK)
Administración / Administrador	Ingreso al sistema web con autenticación	Inicio de sesión con datos válidos	Ingreso y navegación por los menús	14 – Abril – 2020
		Inicio de sesión con datos inválidos	No permite el ingreso	14 – Abril – 2020
	Gestión de niveles, grados y materias	Ingresar, actualizar: niveles, grados y materias	Información ingresada o actualizada	02 – Mayo – 2020
	Gestión de maestros	Ingresar, actualizar: maestros	Información ingresada o actualizada	02 – Mayo – 2020
	Gestión de representantes	Ingresar, actualizar: representantes	Información ingresada o actualizada	10 – Mayo – 2020
	Gestión de estudiantes	Ingresar, actualizar: estudiantes	Información ingresada o actualizada	10 – Mayo – 2020
	Gestión de académicos	Ingresar, actualizar: datos institucionales	Información ingresada o actualizada	10 – Mayo – 2020
	Generar Reportes	Visualizar y generar los reportes solicitados por el MINDE por grado y por estudiante	Los reportes se visualizan y generan con toda la información correcta	30 – Junio – 2020
Maestros / Maestro	Ingreso al sistema web mediante autenticación	Inicio de sesión con datos válidos	Ingreso y navegación por los menús	14 – Abril – 2020
		Inicio de sesión con datos inválidos	No permite el ingreso	14 – Abril – 2020
	Ingreso de calificaciones	Ingreso de calificaciones por actividad, por periodo, y examen final	Las calificaciones se registran y se calculan promedios	24 – Mayo – 2020
		Actualización de calificaciones por actividad, por periodo, y examen final	Las calificaciones se actualizan y se calculan los promedios	24 – Mayo – 2020
	Generar Reportes	Visualizar y generar los reportes por grado y por estudiante	Los reportes se visualizan y generan con toda la información correcta	30 – Junio – 2020
Representante / Representante	Ingreso al sistema web mediante autenticación	Inicio de sesión con datos válidos	Ingreso y navegación por los menús	14 – Abril – 2020
		Inicio de sesión con datos inválidos	No permite el ingreso	14 – Abril – 2020
	Ingreso a la aplicación móvil con autenticación	Inicio de sesión con datos válidos	Ingreso y navegación por los menús	30 – Junio – 2020
		Inicio de sesión con datos inválidos	No permite el ingreso	30 – Junio – 2020
	Revisar calificaciones	Visualizar las calificaciones del estudiante	Las calificaciones se presentan	30 – Junio – 2020

**Nota:** Esta tabla contiene las pruebas funcionales a las que se somete al sistema.

## 5.2.2 Partición de equivalencias

Consiste en agrupar y clasificar las entradas de datos del sistema que compartan los mismos criterios. Se identifica un conjunto de entradas válidas e inválidas que se usan para tratar de detectar errores. En esta técnica se asume que se toma la mayor cantidad de valores representativos a probar.

La tabla 7 contiene la agrupación de las entradas por tipo que tendrá la aplicación web y las posibles clases válidas e inválidas con las que se realizan los casos de prueba.

Tabla 7. Clasificación de entradas

ENTRADA	Condición de entrada	Clase válida	Clase inválida
Cédula de identidad	Valor	1: 10 dígitos y algoritmo de validación de cédula	2: < 10 dígitos 3: >10 dígitos 4: 10 dígitos inválidos por algoritmo
Campos de texto	Valor	5: valor alfabético	6: valores alfanuméricos 7: vacío
Campos numéricos	Valor	8: valores numéricos	9: valores alfabéticos 10: vacío
Correo electrónico	Condición	11: debe contener @	12: vacío 13: no contener @
Campos fecha	Condición	14: fecha válida	15: fecha inválida 16: vacío

**Nota:** Esta tabla contiene las entradas para los campos de los formularios del sistema.

En la tabla 8 se listan los casos de prueba tomando en cuenta la agrupación de las entradas mapeadas en la tabla 7. Aquí se dividen y clasifican los casos de prueba en los diferentes módulos con la intención de contemplar la mayor parte de validaciones posibles que se pueden realizar sobre campos de entrada de los formularios para ingreso de datos del sistema.

Tabla 8. Casos de prueba

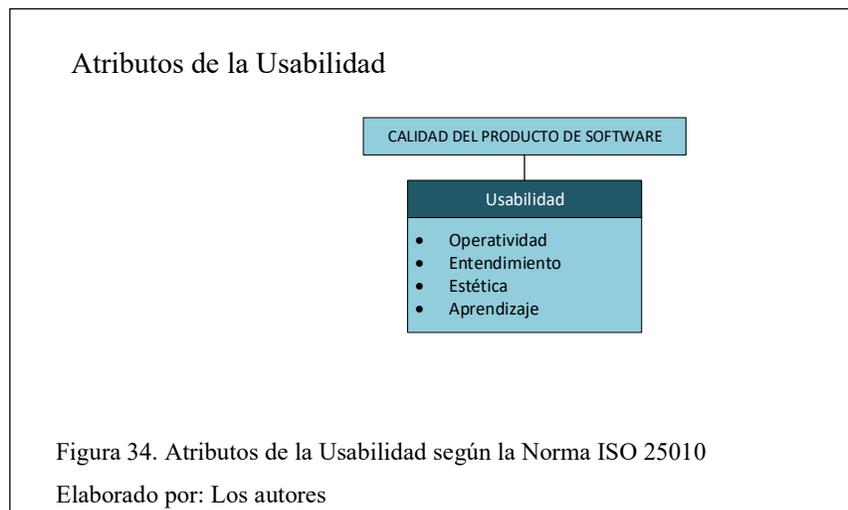
Módulo	Proceso	Entrada	Clases Cubiertas	Fecha Resultado OK
Administración	Ingreso Estudiante	cédula Identidad	(1),(2),(3),(4)	10 – Mayo – 2020
	Ingreso Maestro	cédula Identidad	(1),(2),(3),(4)	02 – Mayo – 2020
	Ingreso Representante	cédula Identidad	(1),(2),(3),(4)	10 – Mayo – 2020
Maestro	Ingreso de datos	campos texto	(5),(6),(7)	10 – Mayo – 2020
	Ingreso de datos	campos numéricos	(8),(9),(10)	24 – Mayo – 2020
Representante	Ingreso de datos	correo electrónico	(11),(12),(13)	10 – Mayo – 2020
	Ingreso de datos	campo fecha	(14),(15),(16)	10 – Mayo – 2020

**Nota:** La tabla contiene los casos de prueba para la validación de las entradas en formularios.

### 5.3 Pruebas de Usabilidad

La norma ISO/IEC 25010 señala que un sistema puede ser evaluado según 8 características: funcionalidad, seguridad, interoperabilidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. (ISO 25010, s.f.)

El sistema del proyecto es evaluado en base a la característica de usabilidad. Se elige esta propiedad porque es la más importante para los usuarios del sistema. La figura 34 muestra los atributos a ser evaluados en el sistema.



Para medir la percepción de usabilidad se utiliza el método de encuesta. Debido a que el flujo más complejo del sistema es en el que intervienen maestros y administrador se elige testear a este grupo de usuarios. La encuesta tiene preguntas que abarcan las principales características que sugiere la norma. El usuario debe marcar un valor entre 1 y 5, 1 si el usuario está totalmente en desacuerdo y 5 si está totalmente de acuerdo. La tabla 9 contiene el listado de preguntas que se utilizan en la encuesta de acuerdo a las características relevantes de usabilidad que se consideran en la norma.

**Tabla 9.** Listado de preguntas propuestas

Característica	Número	Pregunta
Operatividad	1	El sistema permite realizar las acciones y actividades que se esperan.
Operatividad	2	Los tiempos de respuesta del sistema son rápidos.
Capacidad para ser usado	3	Encuentra el sistema fácil de usar.
Entendimiento	4	No se necesita el apoyo de una persona técnica para poder entender el sistema.
Aprendizaje	5	Las acciones a realizar en el sistema son intuitivas y son fáciles de aprender.
Estético	6	Encuentra que el sistema tiene un diseño y funcionalidades apropiadas para la Institución.
Estético	7	Encuentra la interfaz gráfica atractiva.

**Nota:** La tabla contiene las preguntas a usarse en la encuesta para evaluar la usabilidad del sistema.

La tabla 10 contiene el listado de respuestas que los encuestados pueden elegir para cada una de las preguntas de la encuesta.

**Tabla 10.** Respuestas posibles para la encuesta

Valor	Respuesta
1	Respuesta Negativa
2	Respuesta Negativa
3	Respuesta Neutra
4	Respuesta Positiva
5	Respuesta Positiva

**Nota:** La tabla contiene las respuestas posibles para las preguntas de la encuesta.

Para realizar la encuesta se recurre al uso de una herramienta Forms de Google. La figura 35 muestra la captura de pantalla de la encuesta aplicada con las 7 preguntas y las respuestas posibles que los encuestados pueden seleccionar. Participan en la encuesta 11 de los 13 usuarios invitados a contestarla y se obtiene que un 84.4% de las respuestas son positivas, es decir, según los usuarios el sistema cumple con los atributos de usabilidad.

### Encuesta en Google Forms

**Sistema Web UEA Solidaridad**  
Por favor marcar para cada respuesta un valor entre 1 y 5 (1 para "totalmente en desacuerdo" y 5 para "totalmente de acuerdo") según su experiencia con el sistema.  
*\*Obligatorio*

1. El sistema permite realizar las acciones y actividades que esperaba. \*

1 2 3 4 5  
Totalmente en desacuerdo      Totalmente de acuerdo.

2. Los tiempos de respuesta del sistema son rápidos. \*

1 2 3 4 5  
Totalmente en desacuerdo      Totalmente de acuerdo.

3. Encontré el sistema fácil de usar. \*

1 2 3 4 5  
Totalmente en desacuerdo      Totalmente de acuerdo.

4. No necesito el apoyo de una persona técnica para poder entender el sistema. \*

1 2 3 4 5  
Totalmente en desacuerdo      Totalmente de acuerdo.

5. Las acciones a realizar en el sistema son intuitivas y fueron fáciles de aprender. \*

1 2 3 4 5  
Totalmente en desacuerdo      Totalmente de acuerdo.

6. Encontré que el sistema tiene una interfaz apropiada para la institución. \*

1 2 3 4 5  
Totalmente en desacuerdo      Totalmente de acuerdo.

7. Encontré la interfaz gráfica atractiva. \*

1 2 3 4 5  
Totalmente en desacuerdo      Totalmente de acuerdo.

**Enviar**

El formulario se creó en Equifax, Inc. [Demuestra más](#)  
Google Formularios

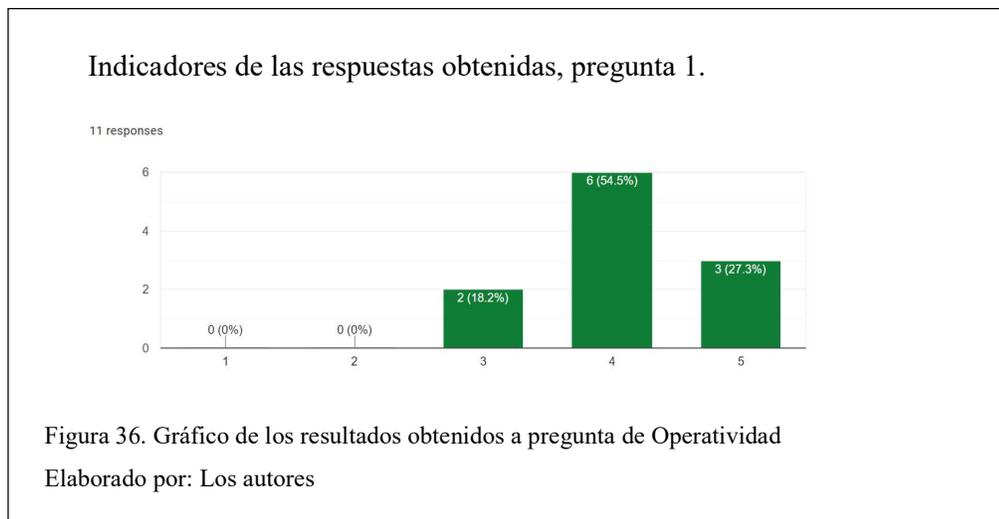
Figura 35. Captura de pantalla de la encuesta en Google Forms  
Elaborado por: Los autores

### 5.3.1 Resultados de las Pruebas

Luego de realizadas las encuestas se tabulan las respuestas y se detallan los resultados obtenidos, se muestran los mismos mediante gráficos de barras.

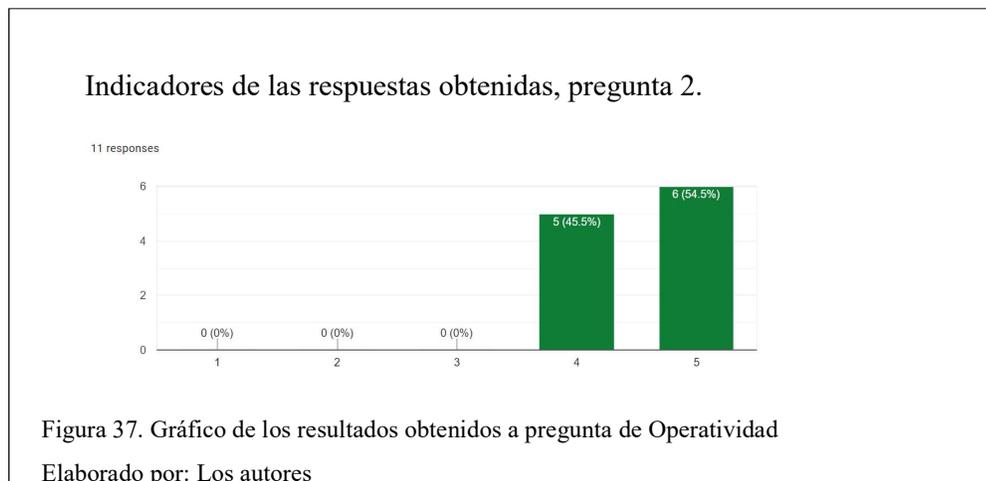
Pregunta 1: ¿El sistema permite realizar las acciones y actividades que se espera?

Resultado: El 81.8% de los usuarios responde afirmativamente (figura 36).



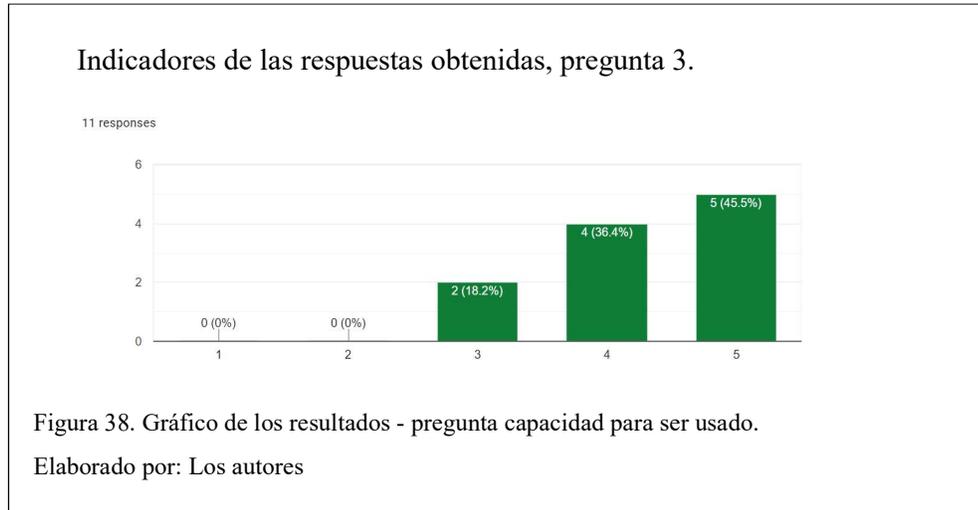
Pregunta 2: ¿Los tiempos de respuesta del sistema son los que se esperan?

Resultado: El 100% de los usuarios piensa que el sistema no es lento (fig. 37).



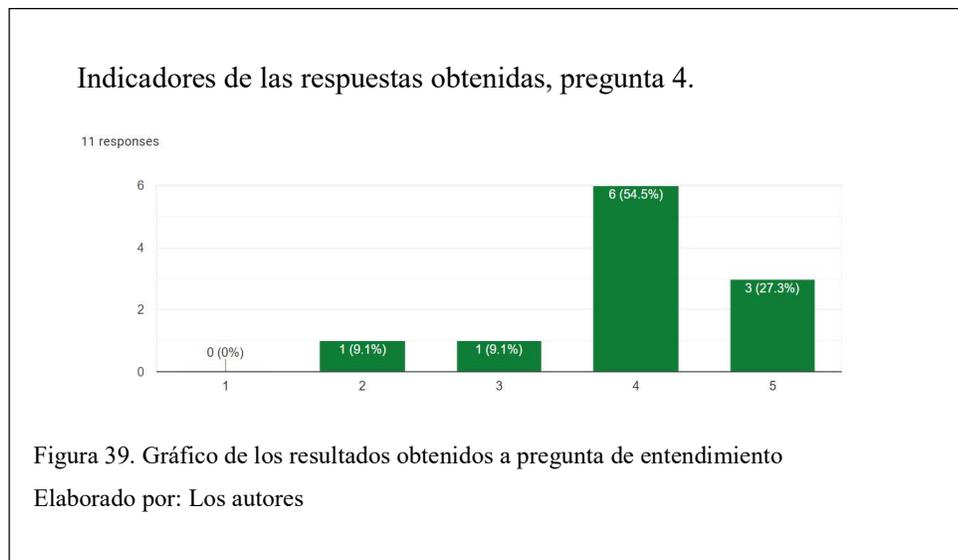
Pregunta 3: Encontré el sistema fácil de usar.

Resultado: El 81.9% de usuarios cree que el sistema es fácil de usar (figura 38)



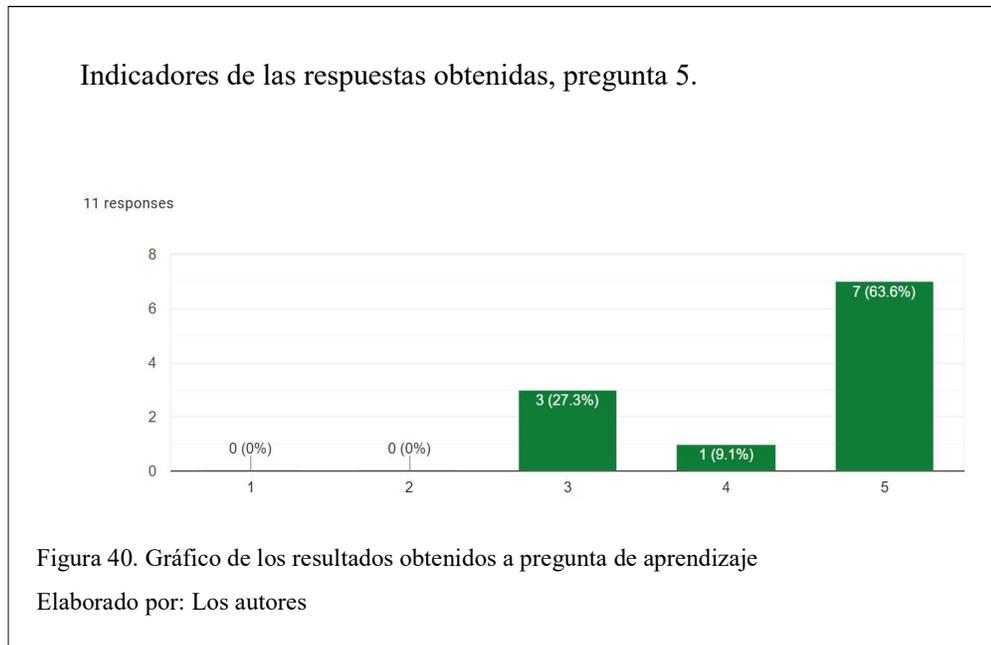
Pregunta 4: No necesito el apoyo de una persona técnica para poder entender el sistema.

Resultado: El 81.8% de los usuarios responde que pueden usar el sistema sin apoyo de una persona técnica (figura 39).



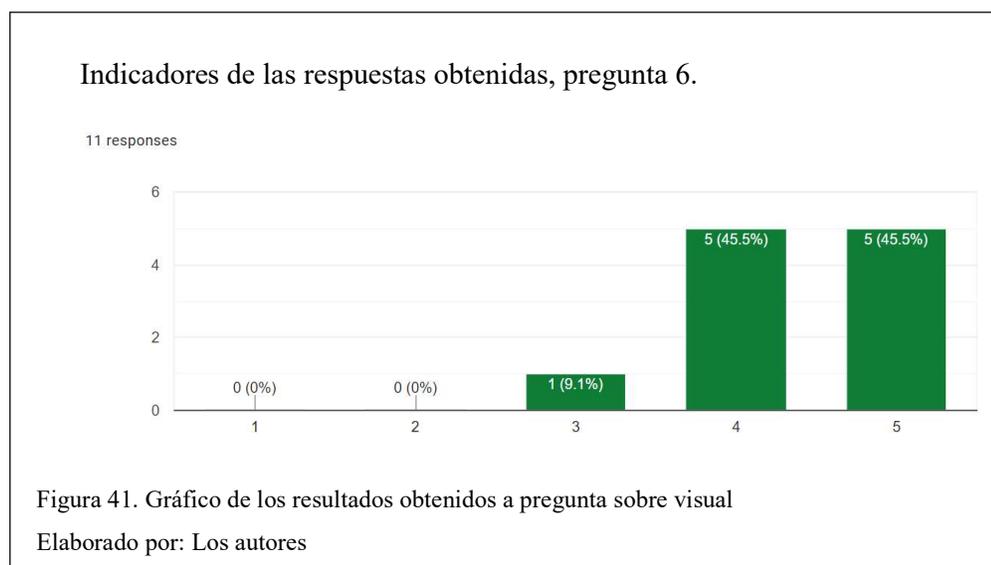
Pregunta 5: Las acciones a realizar en el sistema son intuitivas y fáciles.

Resultado: El 72.7% de los usuarios responde afirmativamente (figura 40).



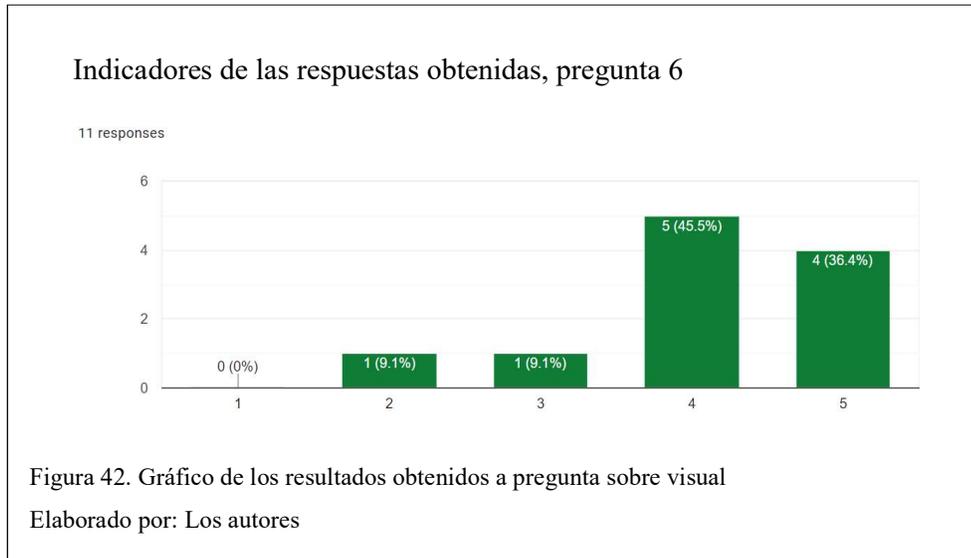
Pregunta 6: Encontré que el sistema tiene un diseño y funcionalidades apropiadas para la Institución.

Resultado: El 91% de los usuarios responde que el sistema tiene una interfaz apropiada para la Institución (figura 41).



Pregunta 7: Encontré la interfaz gráfica atractiva.

Resultado: El 81.9% de los usuarios responde que la interfaz les parece atractiva (figura 42).



### 5.3.2 Síntesis de los Resultados Obtenidos en las Pruebas de Usabilidad

Las 7 preguntas generan 77 respuestas, las mismas que se clasifican entre positivas, neutras y negativas.

Como síntesis de los resultados se obtiene que un 84.4% de las respuestas son positivas, es decir, según los usuarios el sistema cumple con los atributos de usabilidad. Existe un 13% de respuestas neutras y únicamente un 3% de respuestas negativas.

La figura 43 muestra un gráfico de tipo pastel de los resultados obtenidos en la encuesta realizada.

Gráfico pastel de los resultados obtenidos

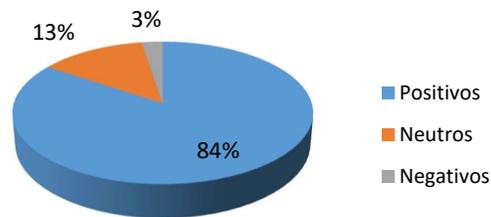


Figura 43. Porcentajes de los tipos de respuestas obtenidos en la encuesta

Elaborado por: Los autores

#### 5.4 Pruebas de rendimiento utilizando Jmeter

Con las pruebas de rendimiento se evalúa la capacidad del sistema en su componente web para soportar un número de visitas determinado. Para realizar este tipo de pruebas se utiliza la herramienta Jmeter.

Para realizar las pruebas es necesario subir el sistema a la web, para ello se acuerda con la escuela la adquisición del dominio “ueasolidaridad.com” y el alquiler del host para el siguiente año. Tras la implementación del sistema y comprobar que funciona correctamente se procede a realizar los testeos correspondientes.

Previo a las pruebas en Jmeter se utiliza el plugin del navegador Chrome llamado BlazeMeter para realizar una grabación del flujo a seguir y de las acciones en el sistema que están consideradas para la evaluación. Esta herramienta permite exportar estas acciones en el formato que Jmeter reconoce, de tal manera que se convierte en el plan de pruebas.

La figura 44 muestra el plan de pruebas que se obtiene de este ejercicio para el usuario maestro. El plan contempla el ingreso al sistema mediante el login, luego la navegación para visualizar un reporte de calificaciones, salir y navegar por otra opción del menú.



Este plan de pruebas se sube en Jmeter y se configura la carga en consumo que tiene o podría tener el sistema durante su uso. Para este ejercicio se parametriza un número de 3 usuarios por cada 180 segundos navegando y ejecutando acciones a través de las distintas páginas del sitio.

El resultado de la evaluación es exitoso, Jmeter indica que se logra 0% de error como se puede observar en la séptima columna de la tabla capturada en imagen y que se muestra en la figura 45.

### Captura de pantalla de los resultados de las pruebas con Jmeter

Label	# Sam...	Average	Min	Max	Std. Dev.	Error %	Through...
https://ueasolidaridad.com/login/auth/	3	664	657	669	5.10	0.00%	1.5/min
https://ueasolidaridad.com/teacher/teacher_dashboard/	6	2831	2023	3536	664.94	0.00%	2.3/min
https://ueasolidaridad.com/teacher/upload_marks/	3	1942	1908	2003	42.82	0.00%	1.5/min
https://ueasolidaridad.com/teacher/tab_quimestre/	3	1419	1351	1473	50.87	0.00%	1.5/min
https://ueasolidaridad.com/teacher/get_class_section/17	3	327	319	344	11.56	0.00%	1.5/min
https://ueasolidaridad.com/teacher/get_class_subject/17	3	339	329	351	8.99	0.00%	1.5/min
https://ueasolidaridad.com/teacher/marks_selector_quimestre	3	538	523	564	18.46	0.00%	1.5/min
https://ueasolidaridad.com/teacher/marks_upload_quimestre/17/38/48/2	3	1955	1838	2069	94.36	0.00%	1.5/min
Test	3	12850	12447	13159	298.18	0.00%	1.1/min
TOTAL	30	2570	319	13159	3558.95	0.00%	11.4/min

Figura 45. Resultados obtenidos en las pruebas con Jmeter

Elaborado por: Los autores

El resultado muestra que la página puede soportar hasta 11 usuarios por minuto sin problemas y el tiempo promedio de respuesta de las páginas es de 2570 milisegundos, es decir 2.5 segundos y el menor tiempo fue de 319 milisegundos. Se observa en la figura 44 que la transacción de login está por debajo del segundo y que la generación del reporte tiene un tiempo promedio de 2 segundos. Todos estos datos permiten concluir que el desempeño en la respuesta de peticiones es adecuado para el uso planteado. En otras pruebas se “estresa” al sistema aumentando la cantidad de usuarios. Se reporta un índice de error de 2.22% cuando llega a 20 usuarios concurrentes por minuto como se puede observar en la Figura 46.

### Captura de pantalla de los resultados de las pruebas con Jmeter

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
https://ueaso...	20	908	580	3422	659.19	0.00%	21.1/min
https://ueaso...	20	312	299	326	8.83	0.00%	22.3/min
https://ueaso...	20	468	296	1137	182.43	0.00%	21.0/min
https://ueaso...	20	315	288	337	12.08	0.00%	21.2/min
https://ueaso...	20	315	301	341	12.06	0.00%	21.2/min
https://ueaso...	20	333	296	613	67.61	0.00%	21.3/min
https://ueaso...	20	425	76	563	138.30	10.00%	22.2/min
https://ueaso...	20	333	290	497	54.76	0.00%	22.3/min
Test	20	3412	2671	6628	857.68	10.00%	15.7/min
TOTAL	180	758	76	6628	1024.50	2.22%	2.4/sec

Figura 46. Resultados con error obtenidos en las pruebas con Jmeter

Elaborado por: Los autores

## CONCLUSIONES

- El sistema propuesto en este proyecto, genera una optimización en los procesos de gestión de información educativa a la UEA Solidaridad, es así que en el registro de calificaciones se reduce el tiempo aproximado a la mitad de lo que usualmente les toma a los profesores y en la obtención de reportes la optimización es aún más evidente pues el proceso les toma un 30% del tiempo que antes destinaban a esta tarea.
- Los padres de familia que antes tenían que movilizarse hasta la institución para consultar novedades académicas de sus representados ahora tienen la facilidad de averiguar el desempeño académico de sus hijos desde cualquier lugar en donde se encuentren a través de la aplicación que es fácil de usar e intuitiva.
- El sistema es de fácil manejo y comprensible, es decir cumple con las principales características de usabilidad, conclusión a la que se llega gracias a la encuesta realizada en donde se obtuvo que el sistema obtiene un 84% de calificaciones positivas en las preguntas sobre los atributos de usabilidad evaluados.
- El uso del framework Codeigniter y la implementación de un layout basado en Bootstrap permitió acelerar el desarrollo del sistema, pues con el primero se generó la estructura básica de arquitectura MVC y con el segundo se estableció de forma muy práctica y en poco tiempo una plantilla para soportar la funcionalidad del sistema y navegar a través del mismo.

- El sistema facilita la obtención de información histórica de un estudiante pues los datos académicos se almacenan en la base de datos correspondiente y se tienen opciones para consultarlos. Esta información es muy útil ya que suele ser requerida en los procesos de matrícula para cambios de escuela o colegio, pero además puede servir para análisis más profundos, proyecciones o estadística que la institución requiera obtener.

## RECOMENDACIONES

- Es necesario realizar un backup de la base de datos al menos una vez cada 15 días para de esta forma garantizar que se cuenta con el respaldo de los datos ante cualquier eventualidad no programada que pudiera afectar la información.
- La UEA Solidaridad debe realizar esfuerzos de comunicación para generar una masiva descarga de la aplicación por parte de los padres de familia. Acciones como; mencionar la aplicación en los procesos de matrícula, las reuniones de padres de familia, y en publicaciones de sus redes sociales, sirven para que la adopción del uso en los representantes sea el esperado.
- Para un óptimo funcionamiento de la aplicación móvil se debe usar en versiones de Android 7.0 o superiores.
- Apalancar el desarrollo mediante el uso de plantillas para generar el entorno gráfico o layouts del sistema y que de esa forma los esfuerzos se enfoquen más en lograr cumplir los aspectos funcionales con alta calidad. Bootstrap es un buen ejemplo de una herramienta para lograr toda esta optimización.
- Utilizar metodologías ágiles como Scrum facilita un entorno de gestión del proyecto adecuado para generar avances notorios en corto tiempo. En lo posible se deben cumplir los lineamientos de la metodología como el tiempo de los sprints, reuniones diarias cortas, cumplir con los entregables por sprint, etc.

## LISTA DE REFERENCIAS

### Artículos Académicos

- Ahmad, M. S., Musa, N. E., Nadarajah, R., Hassan, R., & Othman, N. E. (2013). Comparison between android and iOS Operating System in terms of security. *2013 8th International Conference on Information Technology in Asia (CITA)*, 1-4. <https://doi.org/10.1109/CITA.2013.6637558>
- Camarena, J. (2012). *Automatización de la codificación del patrón modelo vista controlador ( mvc ) en proyectos orientados a la Web*.
- Can, M. H. (2016). Use of mobile application: Means of communication between Parents and Class Teacher. *World Journal on Educational Technology: Current Issues*, 8(3), 252. <https://doi.org/10.18844/wjet.v8i3.834>
- Carrera, P. F. De. (2014). *Aplicación Web de bases de datos usando el Framework Ruby on Rails*.
- Castro, M., & Sánchez, D. (2013). *Aplicación de Servicios Web SOAP / REST para funcionalidades existentes en sistemas informáticos provinciales*. 23-37.
- Cobo, Á., Gómez, P., Pérez, D., & Rocha, R. (2009). *R OCÍO R OCHA PHP y MySQL Tecnologías para el desarrollo de*. [www.diazdesantos.es/ediciones](http://www.diazdesantos.es/ediciones)
- Lobato Tapia, R. N. L. C. (2019). React Native: acortando las distancias entre desarrollo y diseño móvil multiplataforma. *Revista Digital Universitaria*, 20(5). <https://doi.org/10.22201/codeic.16076079e.2019.v20n5.a5>

### Libros

- Luján Mora, S. (2002). *Programación de aplicaciones web : historia, principios básicos y clientes web*. Editorial Club Universitario.  
<http://rua.ua.es/dspace/handle/10045/16995>

Pressman, R. (2002). *Ingeniería del Software. Un enfoque práctico*.

Rad, N. K., & Turley, F. (2013). *The Scrum Master Training Manual A Guide to Passing the Professional Scrum Master (PSM) Exam*.

<http://mgmtplaza.com/latestversion>

Winer, D., & Spurlock, J. (2010). *Bootstrap*.

Zakas, N. C., Mcpeak, J., & Fawcett, J. (2006). *Professional Ajax* (Número 978).

### **Sitios Web**

*¿Simplemente qué es Node.js?* (s. f.). Recuperado 27 de julio de 2020, de

<https://www.ibm.com/developerworks/ssa/opensource/library/os-nodejs/index.html>

*Architectural overview of Cordova platform - Apache Cordova*. (s. f.). Recuperado

26 de julio de 2020, de

<https://cordova.apache.org/docs/en/latest/guide/overview/index.html>

*HTML5 - HTML | MDN*. (s. f.). Recuperado 29 de junio de 2020, de

<https://developer.mozilla.org/es/docs/HTML/HTML5>

*IDC - Smartphone Market Share - OS*. (s. f.). Recuperado 29 de junio de 2020, de

<https://www.idc.com/promo/smartphone-market-share/os>

*Idukay – Optimiza el aprendizaje*. (s. f.). Recuperado 26 de julio de 2020, de

<https://idukay.com/>

*ISO 25010*. (s. f.). Recuperado 10 de junio de 2020, de

<https://iso25000.com/index.php/normas-iso-25000/iso-25010>

*PHP: ¿Qué es PHP? - Manual*. (s. f.). Recuperado 29 de junio de 2020, de

<https://www.php.net/manual/es/intro-what-is.php>

*PHP - EcuRed*. (s. f.). Recuperado 29 de junio de 2020, de

<https://www.ecured.cu/PHP>

*Scrum Guide* | *Scrum Guides*. (s. f.). Recuperado 29 de junio de 2020, de

<https://www.scrumguides.org/scrum-guide.html#team-sm>

*Web Services Glossary*. (s. f.). Recuperado 26 de mayo de 2020, de

<https://www.w3.org/TR/ws-gloss/#intro>

*What are Progressive Web Apps?* (s. f.). Recuperado 26 de julio de 2020, de

<https://web.dev/what-are-pwas/>