

**UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA**



TRABAJO DE GRADO

**SISTEMA INFORMÁTICO PARA EL REGISTRO Y CONTROL DEL EXPEDIENTE
ACADÉMICO DE LOS JÓVENES DEL PROGRAMA OPORTUNIDADES, DE LA
FUNDACIÓN GLORIA DE KRIETE, SEDE SANTA ANA**

PARA OPTAR AL GRADO DE

INGENIERO DE SISTEMAS INFORMÁTICOS

PRESENTADO POR

**DAVID MAURICIO FUENTES GARCIA
EDUARDO ENRIQUE PEÑA ESCALANTE
NELSON ALFREDO SALAZAR SALAZAR**

DOCENTE ASESOR

INGENIERO CARLOS ARTURO RUANO MORÁN

ABRIL, 2020

SANTA ANA, EL SALVADOR, CENTROAMÉRICA

UNIVERSIDAD DE EL SALVADOR

AUTORIDADES



M.Sc. ROGER ARMANDO ARIAS ALVARADO

RECTOR

DR. RAÚL ERNESTO AZCÚNAGA LÓPEZ

VICERRECTOR ACADÉMICO

ING. JUAN ROSA QUINTANILLA QUINTANILLA

VICERRECTOR ADMINISTRATIVO

ING. FRANCISCO ANTONIO ALARCÓN SANDOVAL

SECRETARIO GENERAL

LICDO. LUIS ANTONIO MEJÍA LIPE

DEFENSOR DE LOS DERECHOS UNIVERSITARIOS

LICDO. RAFAEL HUMBERTO PEÑA MARIN

FISCAL GENERAL

FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
AUTORIDADES



M.Ed. ROBERTO CARLOS SIGÜENZA CAMPOS
DECANO

M.Ed. RINA CLARIBEL BOLAÑOS DE ZOMETA
VICEDECANA

LICDO. JAIME ERNESTO SERMEÑO DE LA PEÑA
SECRETARIO

INGENIERO DOUGLAS GARCÍA RODEZNO
JEFE DEL DEPARTAMENTO DE INGENIERÍA Y ARQUITECTURA

AGRADECIMIENTOS

Como grupo de trabajo durante la realización de este proyecto de grado nos sentimos agradecidos por llegar a estas circunstancias de la vida en donde damos un paso más en nuestra carrera como profesionales. A las siguientes personas les debemos nuestra gratitud ya que nos ayudaron con su apoyo y palabras de aliento durante todo el proceso:

Primeramente, agradecidos con Dios Todopoderoso y la Virgen María por permitirnos terminar nuestro nivel académico como ingenieros y hacernos camino a las adversidades de las cuales esta implicó de la mejor forma. A nuestros padres de familia, los cuales siempre estuvieron con y para nosotros en todo momento no solo durante nuestro trabajo de grado sino también en toda nuestra vida como estudiantes que nos ayudaron a llegar a este punto importante. A nuestros amigos y compañeros de carrera, que siempre quedaran en nuestros recuerdos, los momentos durante toda la carrera aprendiendo juntos, aprendiendo de nuestros errores y triunfando en nuestras metas.

A los docentes encargados de impartir las materias por sus consejos y enseñanzas tanto dentro como fuera del salón de clases. A todo el personal de la fundación Gloria de Kriete de Santa Ana que estuvo con nosotros a lo largo de este proceso dando su apoyo, opiniones y materiales didácticos que llevaron a finalizar este trabajo. Por último, agradecidos entre nosotros por nuestro trabajo, cooperación y dedicación durante todo este tiempo para llegar a estas circunstancias y que la vida siga delante de la mejor forma posible.

INDICE

INTRODUCCIÓN	XVII
CAPITULO I: PLANTEAMIENTO DEL PROBLEMA	19
1.1 ANTECEDENTES	20
1.1.1 GENERALIDADES.	20
1.1.2 EL PROGRAMA OPORTUNIDADES	20
1.1.3 SURGIMIENTO DE LA NECESIDAD DE UN SISTEMA INFORMÁTICO.	22
1.1.4 PROYECTOS SIMILARES PREVIOS	23
1.2 PLANTEAMIENTO DEL PROBLEMA	25
1.2.1 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL.....	25
1.2.2 FORMULACIÓN DEL PROBLEMA	27
1.3 MARCO TEÓRICO	27
1.3.1 TERMINOS GENERALES.....	27
1.3.2 CONTEXTO TECNOLÓGICO	28
1.4 JUSTIFICACIÓN.....	34
1.5 OBJETIVOS.....	35
1.5.1 OBJETIVO GENERAL.....	35
1.5.2 OBJETIVOS ESPECIFICOS	35
1.6 ALCANCES.....	36
1.7 METODOLOGÍA DE TRABAJO.....	37
1.7.1 PROPUESTA DE METODOLOGIA.....	37
1.8 FACTIBILIDAD DEL PROYECTO.....	41
1.8.1 FACTIBILIDAD TÉCNICA.....	41

1.8.2 FACTIBILIDAD ECONÓMICA	46
1.8.3 FACTIBILIDAD OPERATIVA	47
1.9 ENTREGABLES	49
CAPITULO II: MARCO TEORICO	50
2.1 DOCUMENTACIÓN DE REQUERIMIENTOS	51
2.1.1 REPORTE DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE	51
2.2 CASOS DE USO	59
CAPITULO III: MARCO METODOLOGICO.....	80
3.1 DISEÑO DE BASE DE DATOS	81
3.1.1 MODELO CONCEPTUAL DE DATOS	81
3.1.2 DICCIONARIO DE DATOS	83
3.2 DEFINICIÓN DE TECNOLOGIAS.....	93
3.2.1 HOSTING COMPARTIDO	93
3.2.2 SERVIDOR WEB HTTP	95
3.2.3 LENGUAJE DE PROGRAMACIÓN Y FRAMEWORK DE DESARROLLO	96
3.3 DISEÑO DE VISTAS DEL SISTEMA.....	99
CAPITULO IV: DESARROLLO E IMPLEMENTACION DEL SISTEMA EELO	104
4.1 ESTRUCTURA DEL SISTEMA.....	105
4.1.1 GENERALIDADES DEL SISTEMA.....	106
4.2 MÓDULO ADMINISTRADOR	109
4.2.1 SUBMÓDULO: DATOS GENERALES	112
4.2.2 SUBMÓDULO: PERSONAL	129
4.2.3 SUBMODULO: ALUMNOS.....	134

4.2.4 SUBMODULO: AÑO EN CURSO	139
4.2.5 SUBMODULO: RECORD DE ALUMNOS	152
4.3 MODULO DOCENTE	162
4.4 MODULO EXPEDIENTE DEL ALUMNO	167
4.5 MANUAL DE USUARIO	170
4.6 ESTRUCTURA ORGANIZACIONAL PARA IMPLEMENTACIÓN.....	170
4.7 PLAN DE IMPLEMENTACIÓN	172
4.7.1 ENTORNO DE DESARROLLO Y PRODUCCIÓN.....	173
4.7.2 MÉTODO DE IMPLEMENTACIÓN.....	174
4.7.3 ACTIVIDADES ESTRATEGICAS PARA LA IMPLEMENTACIÓN.....	175
4.8 ESTADO DE LA IMPLEMENTACIÓN DEL SISTEMA	180
CAPITULO V: CONCLUSIONES Y RECOMENDACIONES	184
5.1 CONCLUSIONES	185
5.2 RECOMENDACIONES	187
REFERENCIAS BIBLIOGRÁFICAS	189
ANEXOS	192

INDICE DE TABLAS

Tabla 1.1 Principales lenguajes de programación.....	31
Tabla 1.2 Detalle del plan Premium del hosting compartido.....	44
Tabla 1.3 Principales navegadores en el trimestre final del 2018.	45
Tabla 1.4 Costo de servicios y software para la implementación.....	46
Tabla 1.5 Costos de desarrollo e implementación del sistema.....	47
Tabla 1.6 Herramientas de software para el desarrollo del código fuente.....	48
Tabla 2.1 Asignación de docentes a materias por año.....	53
Tabla 2.2 Clasificación del nivel académico de los estudiantes.	56
Tabla 2.3 Funciones de los elementos del diagrama de casos de uso.	59
Tabla 2.4 Correspondencia entre los casos de uso y requerimientos funcionales.	64
Tabla 2.5 Caso de Uso – Registro, Edición de Empleados.....	66
Tabla 2.6 Caso de Uso – Registro, Edición de Alumnos.....	67
Tabla 2.7 Caso de Uso – Registro de datos del responsable del alumno	68
Tabla 2.8 Caso de Uso – Registro, Edición de Clases.....	69
Tabla 2.9 Caso de Uso – Login al sistema EELO.	70
Tabla 2.10 Caso de Uso – Registro, Edición de asignaciones de alumnos.....	71
Tabla 2.11 Caso de Uso – Registro, Edición de asignación de docentes.	72
Tabla 2.12 Caso de Uso – Registro de evaluaciones.....	73
Tabla 2.13 Caso de Uso – Consulta de promedios.	73
Tabla 2.14 Caso de Uso – Exportar plantilla de notas FGK.....	74
Tabla 2.15 Caso de Uso – Importar notas FGK.	75
Tabla 2.16 Caso de Uso – Registro de notas FGK.	75

Tabla 2.17 Casos de uso - Edición de notas FGK.....	76
Tabla 2.18 Caso de Uso – Registro de notas del Centro Escolar.	77
Tabla 2.19 Caso de Uso – Consulta de expediente académico.....	77
Tabla 2.20 Caso de Uso – Generar boletas de notas FGK y RDA.....	78
Tabla 2.21 Caso de Uso – Actualizar credenciales de usuario.	78
Tabla 3.1 Tabla departamento de la base de datos.	83
Tabla 3.2 Tabla cargo de la base de datos.	83
Tabla 3.3 Tabla municipio de la base de datos.	83
Tabla 3.4 Tabla estado de la base de datos.....	84
Tabla 3.5 Tabla empleado de la base de datos.....	84
Tabla 3.6 Tabla class de la base de datos.	85
Tabla 3.7 Tabla centro educativo de la base de datos.	85
Tabla 3.8 Tabla alumno de la base de datos.....	86
Tabla 3.9 Tabla ocupacion de la base de datos.	86
Tabla 3.10 Tabla sección de la base de datos.	87
Tabla 3.11 Tabla responsable de la base de datos.....	87
Tabla 3.12 Tabla tipo de responsable de la base de datos.	87
Tabla 3.13 Tabla periodo de la base de datos.	88
Tabla 3.14 Tabla bitácora alumno centro educativo de la base de datos.....	88
Tabla 3.15 Tabla alumno sección de la base de datos.	88
Tabla 3.16 Tabla materia de la base de datos.	89
Tabla 3.17 Tabla grupo materia de la base de datos.	89
Tabla 3.18 Tabla cambio grupo docente de la base de datos.	89

Tabla 3.19 Tabla evaluaciones de la base de datos.	90
Tabla 3.20 Tabla notas FGK de la base de datos.	90
Tabla 3.21 Tabla cambio sección alumno de la base de datos.	90
Tabla 3.22 Tabla grado MINED de la base de datos.....	91
Tabla 3.23 Tabla notas MINED de la base de datos.	91
Tabla 3.24 Tabla criterio actitudinal de la base de datos.	91
Tabla 3.25 Tabla actitud de la base de datos.....	92
Tabla 3.26 Tabla nota actitudinal de la base de datos.	92
Tabla 3.27 Tabla usuarios de la base de datos.....	93
Tabla 3.28 Plan Startup de SiteGround.....	94
Tabla 4.1 Versiones de Herramientas de software implementadas.	107
Tabla 4.2 Funcionalidad de Sección Municipio.	119
Tabla 4.3 Funcionalidad de Sección Centro Educativo.	122
Tabla 4.4 Funcionalidad de Sección Materia.....	124
Tabla 4.5 Funcionalidad de Sección Criterio Actitudinal.	126
Tabla 4.6 Funcionalidad de Sección Actitud.....	126
Tabla 4.7 Métodos y funciones de las clases del submódulo Datos Generales.	128
Tabla 4.8 Funcionalidad de Sección Cargo.....	130
Tabla 4.9 Funcionalidad de Sección Empleado.	133
Tabla 4.10 Funcionalidad de submódulo Alumno.....	137
Tabla 4.11 Funcionalidad de Sección Clase.	141
Tabla 4.12 Funcionalidad de Secciones.....	143
Tabla 4.13 Funcionalidad de Asignación de alumnos a sección.	145

Tabla 4.14 Funcionalidad de Asignación de secciones a docente.	148
Tabla 4.15 Métodos y funciones de las clases del submódulo Año en Curso.	150
Tabla 4.16 Funcionalidades promedios de centro escolar.	153
Tabla 4.17 Funcionalidad del Record de materias oportunidades.....	157
Tabla 4.18 Funcionalidad de Record de promedios actitudinales.	159
Tabla 4.19 Funcionalidad de Sección Ranking.	160
Tabla 4.20 Métodos y funciones de las clases del submódulo Record de alumnos.	162
Tabla 4.21 Funcionalidades del módulo docente.	164
Tabla 4.22 Funcionalidades del Expediente académico del alumno.	169
Tabla 4.23 Personal designado del programa para la implementación del sistema.	172
Tabla 4.24 Actividades y tareas para la implementación del sistema EELO.	177
Tabla 4.25 Distribución de Vistas y Módulos con el equipo de desarrollo.	179
Tabla 4.26 Detalle del espacio en disco de la Base de datos EELO.....	181

INDICE DE FIGURAS

Figura 1.1 Archivo de Excel para el registro de calificaciones por materia y periodo.	25
Figura 1.2 Fases del ciclo de vida de desarrollo de un sistema.	38
Figura 2.1 Funciones Principales del Sistema EELO	52
Figura 2.2 Jerarquía de usuarios del Sistema EELO.	57
Figura 2.3 Principales actividades del Sistema EELO.....	58
Figura 2.4 Elementos del diagrama de Casos de Uso	59
Figura 2.5 Casos de Uso del Administrador del Sistema EELO.....	60
Figura 2.6 Casos de Uso del Administrador y Docente del Sistema EELO.....	61
Figura 2.7 Casos de Uso del Administrador y Alumno del Sistema EELO.....	62
Figura 3.1 Modelo conceptual de datos. Entidad-Relación.	82
Figura 3.2 Estructura de Tecnologías del Sistema EELO.	97
Figura 3.3 Patrón MVC con Laravel.	99
Figura 3.4 Prototipo de vista CRUD de docentes.....	100
Figura 3.5 Prototipo de vista CRUD alumnos.....	100
Figura 3.6 Prototipo vista de asignación de docentes a secciones.	101
Figura 3.7 Prototipo vista detalle de docentes.....	101
Figura 3.8 Prototipo de vista CRUD de evaluaciones.	102
Figura 3.9 Prototipo de vista de registro de notas FGK.....	102
Figura 4.1 Esquema de los módulos y submódulos del sistema EELO	105
Figura 4.2 Uso de directiva @yield en plantilla admin.blade.php	110
Figura 4.3 Panel de navegación del Layout admin.blade.php	111
Figura 4.4 Cabecera del Layout admin.blade.php	112

Figura 4.5 Rutas del submódulo Datos Generales, en el archivo web web.php.....	113
Figura 4.6 Enlaces del módulo Datos Generales en el menú de navegación	113
Figura 4.8 View Departamento.....	115
Figura 4.7 Uso de las directivas @extends y @section en View Departamento.	115
Figura 4.9 Rutas y métodos del controlador DepartamentoController.....	117
Figura 4.10 MVC de Sección Departamento	118
Figura 4.11 Vista de la sección Municipio.	119
Figura 4.12 MVC de la sección Municipio.	120
Figura 4.13 Vista de la Sección Centro Educativo.	121
Figura 4.14 Desplegable para registro de Centro Educativo.	121
Figura 4.15 MVC de Centro Educativo.....	122
Figura 4.16 Vista de Sección Materia.....	123
Figura 4.17 Formulario de registro de Materia.	123
Figura 4.18 MVC de Sección Materia.....	124
Figura 4.19 Vista de sección Criterio actitudinal de evaluación.....	125
Figura 4.20 Vista de sección Actitud.	125
Figura 4.21 MVC de la sección Actitud.	127
Figura 4.22 MVC de la sección Criterio Actitudinal.	127
Figura 4.23 Rutas del módulo Personal.	129
Figura 4.24 Enlaces del módulo personal en el menú de navegación.	129
Figura 4.25 Vista de la sección Cargo.....	130
Figura 4.26 MVC de la sección Cargo.....	131
Figura 4.27 Formulario para registro de Empleado.	132

Figura 4.28 Vista de la sección Empleado.	132
Figura 4.29 MVC de sección Empleado.	133
Figura 4.30 Rutas del submódulo Alumnos.....	134
Figura 4.31 Enlaces del submódulo Alumno en el menú de navegación.	134
Figura 4.32 Vista del submódulo Alumnos.	135
Figura 4.33 Vista de registro de nuevo alumno.	135
Figura 4.34 Vista actualización de un alumno.	136
Figura 4.35 MVC del submódulo Alumno.....	138
Figura 4.36 Rutas de las secciones del submódulo Año en curso.	139
Figura 4.37 Enlaces en el menú del submódulo Año en curso.....	139
Figura 4.38 Vista de la sección Class.	140
Figura 4.39 MVC de la sección Clase.	141
Figura 4.40 Vista secciones por clase.....	142
Figura 4.41 MVC de sección.	143
Figura 4.42 Selector de Clase en vista Asignación de alumnos a secciones.....	144
Figura 4.43 Vista Asignación de alumnos a secciones.	144
Figura 4.44 Modal de edición de asignaciones alumno-sección.	145
Figura 4.45 MVC de sección asignación alumnos a sección.....	146
Figura 4.46 Vista de asignaciones de secciones a docentes.	147
Figura 4.47 Formulario para registrar asignación de sección a docente.	147
Figura 4.48 Formulario para editar la asignación de sección a docente.	148
Figura 4.49 MVC de Bitácora de Asignación de secciones a docentes.	149
Figura 4.50 MVC de asignación de secciones a docente.....	149

Figura 4.51 Rutas de submódulo Record de alumnos.	152
Figura 4.52 Enlaces en el menú del submódulo Record de Alumnos.	152
Figura 4.53 Vista Promedios Centro Escolar.....	153
Figura 4.54 Formulario de ingreso de promedios por alumno.	153
Figura 4.55 Detalle de promedios por materias y periodos.	154
Figura 4.56 MVC de la Sección Record Escolar.	155
Figura 4.57 Vista de Sección Promedios de Materias Oportunidades.	156
Figura 4.58 Edición de notas de Materias Oportunidades.....	156
Figura 4.59 Detalle de Promedios por Materias Oportunidades.....	156
Figura 4.60 MVC de Promedios de Materias Oportunidades.	157
Figura 4.61 Detalle de promedios actitudinales por periodo.	158
Figura 4.62 Vista de promedios actitudinales.....	158
Figura 4.63 MVC de Record de Promedios Actitudinales.....	159
Figura 4.64 Vista del Ranking de promedios globales.	160
Figura 4.65 MVC del Ranking de promedios.....	161
Figura 4.66 Vista de Home del módulo docente.....	162
Figura 4.67 Vista del módulo docente con notas por sección, materia y periodo. ...	162
Figura 4.68 Notas actitudinales cargadas en pantalla.	163
Figura 4.69 Ventana para el registro y edición de Evaluaciones por periodo.....	163
Figura 4.70 MVC Registro de Notas.....	165
Figura 4.71 Exportación e importación de Notas en Excel.	166
Figura 4.72 Vista de Notas FGK del expediente del alumno.	167
Figura 4.73 Vista de notas MINED del expediente del alumno.	168

Figura 4.74 Vista de notas actitudinales del expediente del alumno.	168
Figura 4.75 MVC del módulo del expediente del alumno.	169
Figura 4.76 Organización estructural del programa Oportunidades.	171
Figura 4.77 Fases de implementación del sistema EELO.	172
Figura 4.78 Entorno de desarrollo y producción.	174
Figura 4.79 Diagrama de la ruta crítica para la implementación del sistema EELO.	178
Figura 4.80 Espacio utilizado en disco del Sistema EELO.	180
Figura 4.81 1798 Visitas realizadas el 5 de junio 2020.	182
Figura 4.82 143 usuarios únicos el 5 de junio 2020.	182

INTRODUCCIÓN

En el presente documento se expone la planeación, desarrollo y puesta en marcha del proyecto llamado SISTEMA INFORMÁTICO PARA EL REGISTRO Y CONTROL DEL EXPEDIENTE ACADÉMICO DE LOS JÓVENES DEL PROGRAMA OPORTUNIDADES, DE LA FUNDACIÓN GLORIA DE KRIETE, SEDE SANTA ANA.

Las fases para llevar a cabo el desarrollo del proyecto se encuentran documentadas en cada uno de los capítulos. En el primer capítulo se dan a conocer las generalidades y antecedentes del programa Oportunidades referente al manejo del expediente académico de los jóvenes participantes del programa. Tomando lo anterior en cuenta, se plantea y justifica la necesidad de implementar un sistema web para la gestión y consulta del expediente académico de los jóvenes. Como puntos finales del capítulo, se define la metodología de trabajo y se determina la factibilidad de la ejecución del proyecto, tomando en cuenta los recursos técnicos, económicos y humanos disponibles.

En la segunda fase del proyecto, se procede a realizar la determinación y análisis de los requerimientos con los que debe cumplir el sistema informático, definiendo los diferentes roles del personal involucrado en el registro y control del expediente académico de los alumnos y documentando de forma detallada todas las actividades y tareas que corresponden a dichos roles. Una vez definido el listado de requerimientos se procede a utilizar los Diagramas de Uso para poder establecer de qué forma se debería interactuar con el sistema web para llevar a cabo las actividades y tareas de cada rol de usuario.

Luego de validar los requerimientos con el director y docentes representantes del programa Oportunidades, se procede al diseño del sistema web, el cual es nombrado EELO (Expediente en línea Oportunidades). En el capítulo tres se documenta la estructuración diseñada para el registro de toda la información en la base de datos del sistema, utilizando el modelo conceptual entidad-relación y su diccionario de datos correspondiente.

Luego se definen las tecnologías a utilizar para llevar a cabo el desarrollo del código fuente e implementación del sistema EELO y finalmente en el capítulo se presenta la maquetación de las vistas que conforman la interfaz gráfica del mismo.

Al terminar la etapa de diseño se procede a realizar el plan de implementación en el cual se detallan las actividades principales para llevar a cabo el desarrollo del código fuente del sistema y para la puesta en marcha del mismo. Las generalidades de desarrollo del código fuente de cada una de las vistas del sistema con el framework de Laravel son documentadas en el capítulo cuatro. El plan de implementación, el estado de la misma y sus resultados y conclusiones son presentados en el último capítulo.

CAPITULO I: PLANTEAMIENTO DEL PROBLEMA

Este capítulo abarca los elementos que describen los aspectos generales del proyecto. Se dan a conocer los antecedentes y la situación actual de la fundación FGK y su programa Oportunidades. Se plantean los objetivos y alcances del proyecto y se justifican los beneficios que se lograrán con la realización del mismo. En este capítulo se define la metodología que se adoptará para llevar a cabo la ejecución del proyecto y también la planificación de los recursos.

1.1 ANTECEDENTES

1.1.1 GENERALIDADES.

La Fundación Gloria de Kriete, FGK de forma abreviada, es una fundación de carácter familiar, sin fines de lucro, constituida en diciembre de 2004. La Fundación opera a través de proyectos propios enfocados a la salud, educación, desarrollo comunitario, emprendedurismo y juventud/prevención, así como también apoyando a organizaciones comprometidas con el bienestar y la superación de la familia salvadoreña.

Una de las iniciativas de gran impacto social de la fundación Gloria de Kriete es el premio Ayudando a Quienes Ayudan, el cual es un reconocimiento económico que se brinda por medio de un concurso a instituciones no gubernamentales de carácter social que estén comprometidas con el desarrollo de proyectos humanos y comunitarios en cualquier parte de El Salvador. Con la entrega de dicho premio y los proyectos permanentes de la fundación, los cuales son programas operados por diferentes ONG´s, FGK busca servir y ayudar a la población salvadoreña atendiendo sus necesidades primordiales en educación y salud, así como promover el desarrollo comunitario y el emprendedurismo en la juventud del país.

1.1.2 EL PROGRAMA OPORTUNIDADES

Oportunidades es uno de los programas de la fundación enfocado al área de la educación, el cual inicio en el año 2007 y brinda estudios complementarios a jóvenes de excelencia académica y recursos limitados durante su bachillerato, también otorga becas y financiamiento educativo para asegurar sus estudios superiores y además ofrece apoyo en el proceso de intermediación laboral. El programa inició operaciones con el nombre Supérate y contaba con múltiples sedes en todo el país, en el 2012 se convirtió en el programa Oportunidades y actualmente solo cuenta con sede en San Salvador y Santa Ana.

- **VISIÓN DEL PROGRAMA OPORTUNIDADES:**

Ser el programa líder de excelencia académica y formación integral, que transforme a la juventud salvadoreña de recursos limitados y alto desempeño, en agentes de cambio para su comunidad y El Salvador.

- **MISIÓN DEL PROGRAMA OPORTUNIDADES:**

Aumentar las oportunidades de inclusión social y éxito de jóvenes de ambos sexos, recursos limitados y excelencia académica, brindando una formación integral que les acompañe y oriente a lo largo del bachillerato, estudios superiores y la primera inserción laboral. (Programa Oportunidades - Quienes Somos, s.f.)

El programa consta de tres fases las cuales son: Complemento Académico, Estudios Superiores e Intermediación Laboral.

- **FASE 1: COMPLEMENTO ACADÉMICO**

La fase inicial del programa Oportunidades tiene una duración de 3 años y consiste en una modalidad de tiempo completo en la que los jóvenes continúan cursando el bachillerato en su instituto público y además asisten a las clases del programa durante los días de semana, completando la jornada de estudio diaria con ambos compromisos. Las materias que reciben en esta fase abarcan las disciplinas de matemáticas, computación, inglés, formación lingüística y científica, emprendimiento y orientación vocacional. Además, en esta fase se les brinda a los jóvenes todos los recursos básicos que les permiten centrar sus esfuerzos y dedicación en los estudios.

- **FASE 2: ESTUDIOS SUPERIORES**

Los alumnos ingresan a la segunda etapa del programa tras haber logrado graduarse de su bachillerato y cumplir con todos los requisitos de la fase anterior. La segunda etapa, llamada estudios superiores, consiste en brindar financiamiento educativo por medio de becas y créditos educativos con los cuales los jóvenes tienen la oportunidad de continuar sus estudios en carreras de diferente índole y duración en distintas instituciones públicas y privadas.

La selección de la carrera que realizan los beneficiarios, es el resultado de un proceso personalizado de Orientación Vocacional que el programa desarrolla con cada joven a lo largo de la primera etapa. El financiamiento aplica para Técnicos, Profesorados, Diplomados, Licenciaturas e Ingenierías. El programa cuenta con convenios firmados con veinte universidades nacionales y tres instituciones extranjeras.

- **FASE 3: INTERMEDIACIÓN LABORAL**

Con la intención de alcanzar el objetivo final del programa, en la tercera etapa se apoya a cada joven universitario o graduado de manera personalizada para incrementar sus oportunidades de contratación en el mercado laboral formal dándole acceso a cursos, talleres sobre las tendencias de la empleabilidad, conversatorios con emprendedores y empresarios de éxito, mentorías, pasantías y prácticas remuneradas. De esta manera se busca que cada joven fortalezca su perfil profesional y se convierte en la mejor opción de talento humano preparado para la contratación o para emprender sus propias iniciativas.

El programa Oportunidades busca realizar una doble función de beneficio, tanto para los jóvenes profesionales como para las empresas y organizaciones en general, quienes obtienen la garantía de un joven con formación sólida e integral que ha estado en el programa durante 5 a 8 años y ha recibido una serie de herramientas que lo diferencian de otros jóvenes aspirantes a la contratación. (El Programa - Programa Oportunidades, s.f.).

1.1.3 SURGIMIENTO DE LA NECESIDAD DE UN SISTEMA INFORMÁTICO.

El programa inició sus funciones como tal en la sede de Santa Ana, en el año 2012. Desde entonces, en la fundación se han apoyado en herramientas de software para llevar a cabo la administración y control de todos los datos correspondientes al expediente académico de los jóvenes participantes del programa.

Para el presente año 2020, la sede de Santa Ana cuenta con la participación de 210 estudiantes en la primera fase del programa. Actualmente todo el registro de

los datos académicos se realizar por medio de la herramienta Excel, un software robusto de hojas de cálculo que ofrece los recursos necesarios para llevar a cabo dichos registros.

Sin embargo, el empleo de Excel se ve limitado al almacenamiento de los datos y su representación estadística. Uno de los administradores del proyecto, el Director Luis Roberto Flores ha identificado la necesidad de un sistema de software que se adapte de forma integral a todas las actividades que deben realizar todos los involucrados en el desarrollo del programa, desde los docentes que imparten las clases, quienes deben realizar el registro de las calificaciones de los jóvenes; el director, quien recolecta los archivos de calificaciones de todas las materias para actualizar el expediente académico de todos los alumnos; hasta los padres de familia, quienes deben estar informados con el progreso académico de sus hijos.

Para el director del programa resulta evidente que la información académica de los jóvenes debe estar unificada y consolidada en una base de datos y no estar distribuida en múltiples archivos. Además, se debería poder acceder a dicha información de forma sencilla y rápida por medio de una interfaz intuitiva para ingresar y actualizar los datos en el caso de los docentes; así como para la consulta y verificación en el caso de los padres. De igual forma para el personal administrativo del programa resultaría más fácil realizar reportes estadísticos de control.

1.1.4 PROYECTOS SIMILARES PREVIOS

Actualmente existen muchas herramientas de software que han sido creadas con el objetivo de facilitar las tareas de administración de los expedientes académicos de las instituciones educativas. Aplicaciones de propósito general, así como aplicaciones específicas para expedientes académicos se encuentran disponibles en el mercado de la informática, ya sea software propietario o software libre.

Sin embargo, estas tecnologías no se adaptan a las necesidades específicas que se tienen con la forma de operar en el programa Oportunidades.

En nuestro país se ha llevado a cabo en los últimos años la realización de proyectos con propósitos similares al de desarrollar un expediente académico para instituciones educativas, en una infraestructura informática que permita el fácil acceso a la información desde cualquier dispositivo con conexión a internet. Si bien dichos proyectos no satisfacen los requerimientos específicos del expediente académico para Oportunidades, representan una fuente de información y resultados que pueden ser utilizados como referentes y guías para el desarrollo de un proyecto que busca alcanzar metas y resultados similares.

- **Sistema Informático para la Administración Académica en el Centro de Enseñanza de Idiomas Extranjeros de la Universidad de El Salvador.**

Trabajo de investigación realizado para optar al título de Ingeniero en Sistemas Informáticos en el año 2011, el cual tiene definido como objetivo el desarrollo de un Sistema Informático para la Administración Académica del Centro de Enseñanza de Idiomas Extranjeros de la Universidad de El Salvador (CENIUES), que permita agilizar y asegurar el correcto procesamiento de los datos, optimizando el uso de los recursos. (Rivera Vázquez, Rodríguez Meléndez, Sánchez Garay, & Sánchez Montoya, 2011).

- **Sistema Informático de Administración del Expediente Académico en Línea para la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador.**

Proyecto llevado a cabo para obtener el título de Ingeniero en Sistemas Informáticos, en el año 2011, con el que se documentó todo el proceso para desarrollar un Sistema Informático en línea que permita administrar el Expediente Académico de la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador. (Andrade Ruiz, Aquino Pineda, Mestizo Aguilar, & Monge, 2011).

1.2 PLANTEAMIENTO DEL PROBLEMA

1.2.1 DESCRIPCIÓN DE LA SITUACIÓN ACTUAL

Desde la creación del Programa Oportunidades, se ha contado con el uso de herramientas informáticas que facilitan las tareas necesarias para llevar a cabo el registro y control de calificaciones.

Actualmente se maneja un archivo de Excel por cada materia, en cada periodo del año académico. Dicho archivo contiene una hoja por cada nivel académico (Primer, Segundo y tercer año). El director del programa crea una carpeta compartida en red, la cual contiene el paquete de hojas de cálculo de todas las materias. Cada docente debe editar el archivo correspondiente a su materia de acuerdo al número de evaluaciones que realizará, y la ponderación que tendrá cada una.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1					NO1	10%	10%	10%	10%		10.0	10.0	20.0	20.0	
2	No.	APELLIDOS	NOMBRES	SECCIÓN	100.0%	GUIA 9,1	GUIA 9,2	GUIA 9,3	ESTANDARIZADO	40.0%	GUIA 9,4	GUIA 9,5	JUEGOS	EXAM	60.0%
3	1	Alvarado García	Wendy Marielos	A2	9.0	10.0	10.0	10.0	4.5	3.5	10.0	10.0	10.0	7.5	5.5
4	2	Avelar Avelar	Magerly Epifanía	A1	8.0	10.0	10.0	10.0	9.5	4.0	10.0	10.0	10.0		4.0
5	3	Barillas García	Odalis Alexandra	A2	8.0	10.0	10.0	10.0	3.0	3.3	10.0	10.0	9.3	4.2	4.7
6	4	Calderón Calasin	José Efraín	A1	7.4	10.0	10.0	10.0	4.5	3.5	10.0	10.0	9.5		3.9
7	5	Calderón Flores	Ronald Alexis	A1	9.5	10.0	10.0	10.0	10.0	4.0	10.0	10.0	10.0	7.5	5.5
8	6	Candelario Contreras	Wilber Antonio	A2	9.3	10.0	10.0	10.0	7.5	3.8	10.0	10.0	10.0	7.5	5.5
9	7	Castellanos Musun	Marjorie Esmeralda	A1	8.7	10.0	10.0	10.0	4.0	3.4	10.0	10.0	9.5	7.0	5.3
10	8	Castro Aguilar	Jennifer Lorena	A1	8.9	10.0	10.0	10.0	5.0	3.5	10.0	10.0	9.3	7.5	5.4
11	9	Centeno Alarcón	Mayra Elizabeth	A2	8.6	10.0	10.0	10.0	4.0	3.4	10.0	10.0	9.3	6.5	5.2
12	10	Cerna Nerio	Caín de Jesús	A2	9.2	10.0	10.0	10.0	10.0	4.0	10.0	10.0	10.0	5.8	5.2

Figura 1.1 Archivo de Excel para el registro de calificaciones por materia y periodo.

Cada docente realiza el registro de las calificaciones, conforme se van realizando las distintas evaluaciones. Al finalizar el periodo, el director utiliza otra hoja de cálculo con la cual obtiene la nota final del periodo de cada materia por alumno. El director utiliza dicho archivo para realizar un monitoreo de los alumnos en base a los promedios obtenidos y el número de evaluaciones reprobadas. Además, realiza

reportes estadísticos con los cuales describe el progreso general del programa. Este proceso se lleva a cabo por cada periodo del año académico.

Los padres de familia conocen los resultados de las evaluaciones de los alumnos hasta el momento en que se realizan las reuniones de entrega de calificaciones, en las cuales solamente se entregan las notas promedio de cada asignatura por periodo, ya que es la forma en que se genera el reporte de notas por cada alumno. De esta forma ni los padres, ni los alumnos pueden apreciar el detalle de las calificaciones obtenidas en cada una de las evaluaciones y verificar que dichas calificaciones estén de acuerdo a los resultados obtenidos.

1.2.2 FORMULACIÓN DEL PROBLEMA

En la sede de Santa Ana del programa Oportunidades se ha llegado a identificar una deficiencia en los procesos administrativos y de gestión de notas de los jóvenes que participan en el programa. Dichas deficiencias se deben a que no se cuenta con una herramienta de software que se adapte a los procedimientos de registro y control de calificaciones específicos del programa y no se cuenta con una forma de unificar la información para su fácil acceso y actualización.

Con la falta de tal herramienta el director del programa y el personal administrativo deben invertir más tiempo y realizar un mayor esfuerzo para llevar a cabo las tareas del control académico debido a que la información se encuentra distribuida en múltiples archivos, manejados por los diferentes docentes de las materias; con lo cual resulta necesario estar verificando la actualización de los mismos para poder elaborar los reportes estadísticos y de progreso académico de los jóvenes. Además, los responsables de los jóvenes tienen un acceso limitado y tardío a la información académica debido a que es proporcionada en un formato poco detallado.

1.3 MARCO TEÓRICO

1.3.1 TERMINOS GENERALES

- **SISTEMA INFORMÁTICO**

En la actualidad los sistemas informáticos están abarcando un gran porcentaje de las diferentes organizaciones en cuanto a su uso, estos están representando la solución a diversos problemas que ocurren dentro de ellas debido a la gran cantidad de ventajas que estos proporcionan en comparación con un trabajo manual, desde los procesos más complicados siendo elaborados por otros recursos a llegar a realizarlos en un par de minutos utilizando estos tipos de sistemas.

Estos sistemas están siendo utilizados de forma cotidiana en gran cantidad por diferentes áreas de trabajo, pero ¿que los hace tan útiles? Estos sistemas informáticos

son capaces de automatizar, agilizar y reducir una gran cantidad de tareas que se realizan para un determinado proceso, pero uno de los puntos más importantes es estos sistemas es que puedan ofrecer un resultado esperado y de forma confiable.

- **ADMINISTRACIÓN ACADÉMICA**

La Administración está enfocada a mejorar y sintetizar un conjunto de procesos los cuales se encargan de gestionar todos los componentes que están implicados en realizar diferentes actividades, la administración académica en específico se desenvuelve en el sector educativo y es de suma importancia la cantidad de componentes y subcomponentes con los que se trabaja, perfiles de estudiantes como todo su record académico.

La administración académica se encarga del manejo, coordinación y procesamiento de datos, registros y estadísticas de resultados todos estos con la finalidad de llevar un mejor control y cumplir con la finalidad de las instituciones.

1.3.2 CONTEXTO TECNOLÓGICO

- **DESARROLLO WEB**

Desarrollo web es un término que define la creación de sistemas informáticos en el ámbito del Internet o una intranet; a dichos sistemas se les puede llamar sitios o aplicaciones web. Para conseguirlo se hace uso de tecnologías de software del lado del servidor y del cliente que involucran una combinación de procesos de base de datos con el uso de un navegador web a fin de realizar determinadas tareas o mostrar información.

Funcionalmente, el desarrollador web, que es quien realiza esta labor, normalmente sólo se preocupa por el funcionamiento del software, es tarea del diseñador web preocuparse del aspecto final (layout) de la página y del webmaster el integrar ambas partes. En ocasiones el webmaster también se encarga de actualizar los contenidos de la página.

Los desarrolladores web lo hacen a través de diversos lenguajes de programación. El lenguaje que usan en cada momento depende del tipo de tarea que están haciendo. El desarrollo web se divide, de forma general, en Frontend (la parte cliente) y Backend (la parte servidor).

- **ARQUITECTURA BÁSICA DE UNA APLICACIÓN WEB: FRONTEND Y BACKEND.**

Un desarrollador Frontend se encarga de la composición, diseño e interactividad usando tecnologías como HTML, CSS y JavaScript. El desarrollador Frontend toma una idea y la convierte en realidad. Lo que se ve y lo que se usa, como por ejemplo el aspecto visual del sitio web, los menús desplegados y el texto, son creados por el desarrollador Frontend, que escribe una serie de programas para dar estructura, forma e interactividad a estos elementos.

El desarrollador Backend se encarga de lo que no se ve, es decir, dónde se almacenan los datos. Sin datos no hay Frontend. El Backend consiste en el servidor que acoge la web, una aplicación para ejecutarlo y una base de datos. El desarrollador Backend utiliza programas de computación para asegurar que el servidor, la aplicación y la base de datos tengan un desempeño regular conjunto. Además, analiza qué necesita la empresa y proporciona soluciones de programación eficientes. Para hacer este increíble trabajo, utiliza una serie de lenguajes del lado del servidor, como PHP, Ruby, Python y Java.

Los lenguajes de programación más usados en desarrollo web son principalmente ASP.NET, PHP y JSP.

- **TECNOLOGIAS CLIENTE/SERVIDOR**

Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que paso el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar

solución. Esto dio lugar a desarrollar lenguajes de programación para la web, dinámica, que permitieran interactuar con los usuarios.

Muchas veces tendemos a enumerar los diferentes lenguajes que usamos tanto en diseño como en programación web confeccionando una lista rápida. Es importante clasificarlos según la función que realiza cada uno, si son estáticos o dinámicos, interpretados o compilados, etc.

No vamos a decir que unos son más importantes que otros, no obstante, el programador web va a dedicar la mayor parte del trabajo, con diferencia, al lenguaje que está del lado del servidor, en lo ya antes mencionado 'BACK-END'. En este lado se realizan las labores para la futura administración del sitio web: altas, bajas, modificaciones, cómo se construye la presentación del contenido y tantas otras como necesite el proyecto.

Lenguajes de Programación para el desarrollo de aplicaciones web.	
Java	Reconocido por su legibilidad y simplicidad, Java es uno de los lenguajes de programación más adoptados: más 9 millones de desarrolladores lo usan y está presente en 7 mil millones de dispositivos en todo el mundo. Desde 2001 se mantiene en las primeras posiciones, llegando al número puesto número 2 como la más baja de todas en marzo de 2015. Su enorme popularidad se debe a su poder de permanencia, cuestión que asegura el funcionamiento a largo plazo de las aplicaciones que lo utilizan.
C	Creado entre 1969 y 1972 en los Laboratorios Bell, es uno de los más utilizados en el mundo. Si bien es ejecutado en la mayoría de los sistemas operativos, es de propósito general, con lo cual es muy flexible. Es muy popular para el desarrollo de aplicaciones de escritorio.
C++	Conocido por el nombre "C Plus Plus", este lenguaje de programación orientado a objetos surge como una continuación y ampliación del C. Hay una gran cantidad de programas escritos en C++, como por ejemplo los paquetes de Adobe.
Python	Un lenguaje de programación multiplataforma y multiparadigma, que también es de propósito general. Esto significa que soporta la orientación a objetos, la programación imperativa y funcional. Su simpleza, legibilidad y similitud con el idioma inglés lo convierten en un gran lenguaje ideal para principiantes.
C#	También llamado "C Sharp", este lenguaje de programación orientado a objetos fue desarrollado en el año 2000 por Microsoft para ser empleado en una amplia gama de aplicaciones empresariales ejecutadas en el <i>framework</i> .NET. C Sharp es una evolución del C y C++ que se destaca por su sencillez y modernidad.
Visual Basic .NET	Pasó del número 9 en junio de 2016 al sexto lugar en 2017. Es utilizado por una gran cantidad de personas a lo largo del mundo que no cuentan con conocimientos profundos como desarrolladores, quienes encuentran en visual basic, además de una sintaxis sencilla, la posibilidad de automatizar sus propios procesos y crear sus propias aplicaciones web. Es visto como uno de los lenguajes más amigables para los que recién comienzan, sobre todo a comparación de C#.
Javascript	Antes que nada, no confundir con Java. Son lenguajes distintos. JavaScript es un lenguaje de programación que puede ser utilizado para crear programas que luego son acoplados a una página web o dentro de programas más grandes. Sirve para crear efectos y realizar acciones interactivas. Podemos ver funcionando este lenguaje en servicios como el chat, calculadoras, buscadores de información y un sin fin de utilidades más.
PHP	Creado en 1994 por el programador canadiense Rasmus Lerdorf, nunca pretendió ser un lenguaje de programación, sino que fue creado con la intención de contar con un conjunto de herramientas para el mantenimiento de las páginas web. Es de fácil acceso para nuevos programadores y a su vez ofrece a los más experimentados muchas posibilidades.

Tabla 1.1 Principales lenguajes de programación.

- **GESTORES DE BASES DE DATOS**

Una base de datos no es más que un programa informático capaz de almacenar información de forma estructurada y ofreciendo mecanismos de búsqueda de esa información almacenada óptimos y rápidos. La información se suele organizar en diferentes tablas que suelen estar relacionadas entre ellas.

Cada tabla suele contener información en diferentes registros de un tipo de entidad (clientes, facturas, etc). Sobre esas tablas se crean claves o índices que permiten referenciar los contenidos y optimizar la búsqueda de los mismos.

Aunque existen diferentes tipos de bases de datos, el más usado es el de tipo relacional, que como su nombre indica, permite relacionar los contenidos que alberga mediante diferentes tablas y relaciones entre ellas las cuales llevan un orden lógico que permite su correcta correlación.

En el mundo de los desarrollos web, MySQL es una de las bases de datos más utilizadas por los desarrolladores debido a las funciones que es capaz de cumplir y su agradable interfaz para trabajar, en otras alternativas que se toman en cuenta se pueden mencionar Microsoft SQL Server, PostgreSQL o MongoDB.

En un desarrollo web, las bases de datos nos permiten actuar como almacén de datos y extraer el contenido de la web y disponerlo fácilmente, tal y como el desarrollador requiere en su proyecto web o el internauta necesita visualizarlo.

Aunque mucha gente puede que no sea consciente, la base de datos es una parte fundamental del rendimiento de una web por la importante información que esta se almacena, además de otros factores.

- **PATRONES DE DISEÑO**

Los patrones de diseño son soluciones para problemas típicos y recurrentes que nos podemos encontrar a la hora de desarrollar una aplicación.

En el momento que un desarrollador de sistemas trabaja se puede encontrar con muchos problemas que pueden ser muy comunes con otras aplicaciones buscar

la causa del problema puede tomar bastante tiempo en resolverlo, en estos casos es donde se puede recurrir a usar patrones de diseño para resolverlo de manera rápida.

Los patrones nos dan soluciones concretas a problemas que encontramos normalmente en cualquier lenguaje de programación.

Simplemente facilitan los problemas en los momentos más comunes en los que pueden llegar a ocurrir al momento de trabajar en una aplicación, pero muchas veces o no son aplicables, o la solución no se adapta al 100% a nuestros requerimientos, en dicho caso, el programador es libre de usar su creatividad y adaptar el patrón a sus especificaciones personales, Por supuesto al mismo tiempo en que la tecnología avanza y se pueden crear nuevos patrones de diseños para dar mejores soluciones a los problemas.

1.4 JUSTIFICACIÓN

Disponer de un sistema informático que agilice el proceso de registro de calificaciones y que a su vez permita a los estudiantes consultar dicha información por medio de internet, es un hecho que se vuelve cada vez más necesario e indispensable para todas las instituciones educativas. El MINED, por ejemplo, ha iniciado la implementación de la consulta de notas en línea para los estudiantes de educación media. Con este método de consulta, el Ministerio de Educación ofrece a los padres de familia o responsable de los estudiantes una manera funcional de corroborar el rendimiento escolar de los estudiantes.

En tal contexto, Oportunidades siendo un programa en el cual se imparten materias de complemento académico para jóvenes de educación media, se ve en la necesidad de implementar un sistema que le permita proporcionar la información detallada del progreso de cada estudiante, y que además facilite la generación y actualización de dicha información a los docentes y personal administrativo del programa. Por lo tanto, el proyecto propuesto se enfoca en diseñar, desarrollar e implementar un sistema informático, como una solución que facilite y optimice el proceso de registro y control de calificaciones para el programa.

Con el desarrollo de dicho sistema, los docentes del programa Oportunidades podrán disponer de un mecanismo que les permita organizar las evaluaciones de las materias que imparten, así como también que puedan llevar a cabo el registro de las calificaciones de los alumnos en dichas evaluaciones de forma fácil y segura a través de una interfaz inteligible.

El expediente académico de cada alumno será almacenado en la base de datos del sistema, con la cual se implementarán los mecanismos necesarios para garantizar la integridad y seguridad de dichos datos. Además, con el sistema se brindará un medio a los jóvenes del programa y sus padres para que puedan acceder por medio de internet a su expediente académico y ver el detalle de las calificaciones obtenidas en cada asignatura.

1.5 OBJETIVOS

1.5.1 OBJETIVO GENERAL

- Agilizar los procesos de registro y control de calificaciones del programa Oportunidades, y permitir el fácil acceso de dicha información a los jóvenes participantes del programa y sus padres, a través de la creación de un sistema informático para la gestión del expediente académico.

1.5.2 OBJETIVOS ESPECIFICOS

- Identificar y analizar los requerimientos provistos por el personal involucrado en el registro y control del expediente académico de los estudiantes del programa.
- Diseñar un modelo para el sistema que brinde una solución eficiente a los requerimientos.
- Crear una base de datos permita la unificación y centralización de la información pertinente al progreso académico de los alumnos.
- Desarrollar y probar el sistema informático, llevando a cabo las especificaciones de diseño y garantizando el cumplimiento de los requerimientos.
- Implementar el sistema en una infraestructura informática que garantice la seguridad y fácil acceso de la información a los docentes y alumnos del programa.
- Crear un sistema informático que sea escalable, permitiendo la futura ampliación y mejora de funcionalidades, agregando nuevos módulos y servicios.

1.6 ALCANCES

- El sistema informático de registro y control del expediente académico para el programa oportunidades, será implementado en la fundación Gloria de Kriete, sede Santa Ana.
- El sistema informático estará enfocado en agilizar los procesos de registro y control del expediente académico de los alumnos participantes del programa Oportunidades en la primera fase llamada Complemento Académico.
- La funcionalidad del sistema informático estará enfocada al registro y administración de la información académica de los estudiantes, incluyendo el registro de las calificaciones obtenidas en sus centros escolares de educación media ya que es uno de los requisitos para la participación en el programa y medir su progreso académico.
- Las interfaces gráficas de usuario permitirán el acceso a la información del sistema, por medio de los navegadores de equipos de computador de escritorio, y dispositivos móviles como tablets y smartphones que cuenten con conexión a internet.

1.7 METODOLOGÍA DE TRABAJO

La metodología hace referencia al conjunto de procedimientos racionales utilizados para alcanzar el objetivo o la gama de objetivos que rige una investigación científica, una exposición doctrinal o tareas que requieran habilidades, conocimientos o cuidados específicos. (Colaboradores Wikipedia, 2020). En el área de desarrollo de software, las metodologías son un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de aplicaciones y sistemas de información.

1.7.1 PROPUESTA DE METODOLOGIA

La Metodología estructurada propuesta por los autores Kendall y Kendall es expresada en lo que se conoce como el Ciclo de Vida de Desarrollo de Software. El ciclo de vida de vida del desarrollo de sistemas es un enfoque por fases para el análisis y el diseño, cuya premisa principal consiste en que los sistemas se desarrollan mejor utilizando un ciclo específico de actividades para el analista y el usuario. (Kendall & Kendall, 2011).

Las fases del ciclo de vida del desarrollo de sistemas son siete y cada una se explica por separado. Sin embargo, nunca se realizan como pasos aislados; más bien, es posible que algunas actividades se realicen de manera simultánea, y algunas de ellas podrían repetirse.



Figura 1.2 Fases del ciclo de vida de desarrollo de un sistema.

A continuación, se detallan las principales actividades y características de cada una de las fases del ciclo de vida de sistemas y se describe de qué manera dichas fases serán ejecutadas en el desarrollo del expediente en línea para el programa Oportunidades.

- **Fase 1. Identificación de problemas, oportunidades y objetivos.**

En la fase inicial se identificó la oportunidad de desarrollar el expediente en línea para el programa oportunidades, estudiando la situación actual y planteando el problema a solucionar, con la información obtenida por medio de reuniones y entrevistas con el personal administrativo del programa. Dicha información se encuentra plasmada en el presente capítulo, en el que también se han definido los objetivos y el alcance del proyecto. Esta fase tiene como finalidad determinar la viabilidad del proyecto, y ha sido presentada como anteproyecto al personal administrativo del programa Oportunidades.

- **Fase 2. Determinación de los requerimientos humanos de información.**

En la segunda fase se realizan una serie de reuniones con los usuarios involucrados, docentes y personal administrativo, con la finalidad de determinar cuáles son los procedimientos y actividades para llevar a cabo su trabajo actualmente; así como también la información que necesitan y de qué forma interactúan con ella. En esta fase se busca determinar de qué forma el nuevo sistema resultará útil para el usuario, mejorando su forma de interactuar con la información en comparación con los procesos actuales, creando funcionalidades que le permitan el acceso y edición de dicha información de forma más eficiente.

- **Fase 3. Análisis de las necesidades del sistema.**

En esta fase se realiza el análisis de las necesidades de los usuarios y requerimientos, documentados en la fase anterior, con el apoyo de herramientas y técnicas especiales que permiten representar de manera gráfica los procedimientos y tareas de los usuarios. En esta fase se realiza una propuesta del sistema en la que se sintetiza todo lo que se ha averiguado sobre los usuarios, sus necesidades y la forma en que realizan su trabajo y de qué forma dichas actividades serán abarcadas en el sistema.

- **Fase 4. Diseño del sistema recomendado.**

En la cuarta fase del ciclo de vida de desarrollo de sistemas, se procede a crear el diseño lógico del sistema. Se diseñan los procedimientos y funciones del sistema que permitirán a los usuarios ingresar la información de forma efectiva. En esta etapa se diseñan las interfaces graficas de usuario como los menús de navegación, formularios y pantallas de consultas, de forma que se garantice una correcta introducción y consulta de datos. También se realiza el diseño de la base de datos, en la cual se debe garantizar que se guarden todos los datos que generan información relevante para todos los usuarios del sistema.

- **Fase 5. Desarrollo y documentación del sistema.**

Esta fase consiste en desarrollar y codificar el software del sistema propuesto y eliminar los errores sintácticos y de lógica. En esta etapa es llevada a cabo la ejecución de todo el análisis y diseño realizado en las fases anteriores. Además, se realiza la documentación del sistema por medio de manuales del programador y manuales de uso, en los cuales se detalla la estructura del código fuente del sistema y de qué forma los usuarios deben interactuar con las interfaces del sistema para poder llevar a cabo las actividades y tareas definidas.

- **Fase 6. Prueba y mantenimiento del sistema.**

Esta fase se lleva a cabo de manera simultánea con la fase de desarrollo a nivel de codificación de las funciones del software ya que al desarrollar cada una, se realizan las pruebas unitarias de funcionalidad de forma individual y también en conjunto cuando se integran las funcionalidades nuevas al sistema. Además, cada nueva funcionalidad e interfaz se valida con los usuarios, permitiéndole interactuar con el sistema en un entorno de prueba con el fin de detectar modificaciones, correcciones y errores que se pueden corregir antes de la implementación.

- **Fase 7. Implementación y evaluación del sistema.**

En esta etapa se abarca la puesta en producción del sistema desarrollado. Además, se debe incluir la planificación de una transición del sistema antiguo sin generar problemas de integridad de los datos. En esta fase se debe incluir la capacitación de los usuarios para operar el sistema de manera correcta.

Este marco de trabajo sugiere las siete fases que se deben llevar a cabo para el desarrollo de un sistema. La forma en que se ejecutan las fases, y las técnicas y herramientas utilizadas en cada una de ellas, dependerá de factores como la experiencia del equipo de trabajo, el tiempo del proyecto, la disponibilidad de recursos etc.

1.8 FACTIBILIDAD DEL PROYECTO

1.8.1 FACTIBILIDAD TÉCNICA

La factibilidad técnica del presente proyecto se describe en base a los recursos necesarios y disponibles para llevar a cabo el adecuado desarrollo de las fases de la metodología de trabajo.

- **TÉCNICAS Y HERRAMIENTAS DE ANÁLISIS Y DISEÑO**

El equipo de trabajo del proyecto cuenta con una serie de conocimientos y herramientas para poder realizar una adecuada comprensión, representación y análisis de los requerimientos del sistema, así como también para poder esquematizar y representar la estructura del sistema en la etapa de diseño.

- ❖ **Documentación de Requerimientos – Estándar IEE 830**

El Estándar IEE 830 es una herramienta que define una estructura para una buena documentación de requerimientos para sistemas de software. Dicho formato presenta varias secciones, que tienen el fin de recolectar toda la información que asegure la completa comprensión de las necesidades de los usuarios del sistema a desarrollar.

- ❖ **Diagrama de Casos de Uso**

Parte de los diagramas de comportamiento de los diagramas UML (Lenguaje Unificado de Modelado). Un modelo de caso de uso describe qué hace un sistema sin describir cómo lo hace; es decir, es un modelo lógico del sistema. Presenta al sistema desde la perspectiva de un usuario fuera del mismo (Kendall & Kendall, 2011).

Los diagramas de Caso de uso son una herramienta útil para la representación y descripción de los requerimientos, ya que proporcionan toda la información necesaria de las interacciones que cada uno de los usuarios tiene con el sistema para realizar cada una de sus tareas. Es decir, presenta toda la información clave para la definición del diseño del sistema, para el equipo de desarrolladores.

❖ Metodología de diseño de Base de Datos

El diseño de la base de datos es un punto clave para el desarrollo del sistema, por lo tanto, es necesario llevar a cabo un correcto modelado de todos los datos utilizados para generar información en el sistema. El modelado de los datos se puede trabajar en diferentes niveles de abstracción dependiendo el nivel de detalle.

Para el presente proyecto, es adecuado trabajar a un nivel conceptual, definiendo las principales entidades y sus relaciones con el modelo Entidad-Relación. Así como también a un nivel lógico representando las entidades en tablas y todos sus atributos en columnas, incluyendo el tipo de dato, tamaño, llaves primarias, etc. Además, dichos modelos se acompañan de un diccionario de datos, para describir cada uno de los campos de datos y poder identificarlos fácilmente al momento de llevar a cabo el desarrollo y codificación del sistema.

❖ Diagrama de Clases

Gracias a este diagrama es posible representar todas las entidades del sistema con su estructura interna y además con métodos definidos, para comunicarse y compartir información con otras entidades. Es un diagrama de estructura que permite definir la lógica de todo el sistema, a un nivel de abstracción adecuado para llevar a cabo el desarrollo del sistema sin hacer referencia a la implementación de tecnología concreta.

El equipo de trabajo del proyecto cuenta con los conocimientos para poder aplicar más herramientas de análisis y diseño, considerando las mencionadas anteriormente las adecuadas para poder establecer una adecuada comunicación con los interesados y usuarios del sistema y validar que todos sus requerimientos y necesidades han sido comprendidas y se les ha dado una acertada solución.

- **DISPONIBILIDAD DE HARDWARE PARA LOS USUARIOS**

Los docentes y administradores del programa cuentan con equipos informáticos asignados en las instalaciones de la fundación Gloria de Kriete.

Dado que el desarrollo del expediente en línea está basado en una infraestructura cliente/servidor; los docentes, administradores y alumnos del programa pueden utilizar el equipo asignado y sus computadoras y dispositivos móviles personales. Dichos dispositivos llevarán a cabo la función de clientes del sistema; de tal manera que, los usuarios harán peticiones y envíos de datos al servidor para realizar sus tareas correspondientes. Ya que el almacenamiento y procesamiento de los datos se lleva a cabo en el servidor del sistema y no del lado del cliente, las especificaciones de hardware de un equipo computacional personal serán suficientes.

- **SERVIDOR: HOSTING COMPARTIDO**

El programa Oportunidades cuenta con los servicios de hosting contratados con el proveedor Hostinger.com. Dichos servicios fueron contratados originalmente para levantar un sitio web informativo del programa Oportunidades.

El director del programa realizó el requerimiento específico de utilizar los servicios de hosting contratados para el desarrollo e implementación del expediente en línea de los alumnos. El hosting contratado es de tipo compartido, por lo que los recursos de hardware son compartidos con otros sitios web administrados por terceros.

En el hosting compartido, todos los sitios web alojados comparten los recursos disponibles. Dependiendo del tipo de procesador, memoria, espacio de disco y carga del sistema, la cantidad de sitios alojados puede variar desde unos cuantos pocos a cientos e incluso miles. El alojamiento compartido es una alternativa económica frente a los servidores dedicados y los servidores virtuales. (Colaboradores Wikipedia, 2020).

Hostinger: Web Hosting Premium	
Espacio en Disco (SSD)	20 GB
Ancho de banda	Ilimitado
Bases de datos MySQL	Ilimitado
Certificado SSL	Si
Copias de seguridad diaria	No
Cuentas FTP	Ilimitado
FTP sobre SSL	si
Copias de seguridad semanales	si
Administrador de archivos	Si
Múltiples versiones PHP	Si
Tamaño de base de datos	3 GB
Núcleos de CPU	1 núcleo
RAM	512 MB

Tabla 1.2 Detalle del plan Premium del hosting compartido.

La definición de dicho requerimiento hace que se establezcan unas limitaciones para el proyecto:

- ❖ Utilizar MySQL como motor de Base de datos para el sistema del expediente en línea.
- ❖ Desarrollar el código fuente del sistema informático utilizando el lenguaje PHP.

Estas especificaciones delimitan las tecnologías a utilizar; Sin embargo, estas son altamente utilizadas en el mercado para el desarrollo de aplicaciones web y el almacenamiento de datos. De hecho, el soporte para tales tecnologías se encuentra incluido en la mayoría de los planes de los servicios de hosting por su amplio uso para el desarrollo de sistemas web. Dichas tecnologías serán descritas técnicamente con mayor detalle en la etapa de Diseño.

En cuanto a las especificaciones de hardware ofrecidas en el plan de hosting, se considera que son prestaciones básicas, tomando en cuenta que por lo general el hosting compartido es utilizado para sitios web informativos, blogs, tiendas en línea, etc. Sin embargo, las mismas tecnologías abordadas anteriormente, permiten que el sistema sea completamente portable, facilitando la tarea de trasladar el sistema de expediente en línea a otro tipo de hosting o proveedor HaaS (*Hardware as a service*)

con recursos de mayor capacidad en caso de que la carga del sistema lo demande. Además, la escalabilidad de las especificaciones en los planes de hosting es otra característica a destacar, ya que esto permite que la infraestructura del sistema se vaya adaptando a la demanda de recursos, reduciendo esfuerzos y costos de desarrollo e implementación.

Por último, cabe destacar que cada proveedor de hosting ofrece múltiples tecnologías y servicios que brindan capas de seguridad en las conexiones e integridad de los datos. Dichas prestaciones se contratan de forma adicional a los planes y serán exploradas con más detalle en la fase de diseño.

- **SOFTWARE REQUERIDO PARA LOS USUARIOS DEL SISTEMA**

Cualquiera de los usuarios del sistema, accederá al mismo por medio de una computadora o dispositivo móvil que realizará la función de cliente, haciendo peticiones y envíos de datos al servidor. Dichas operaciones serán realizadas a través de un navegador web, los cuales se encuentran disponibles para descargar e instalarse y no tienen costo de adquisición y uso.

En la tabla 1.3 se muestra el uso de los principales navegadores en el último trimestre del 2018 en base a las estadísticas de W3Schools. Dichos navegadores cuentan con una amplia compatibilidad con las funciones y estructuras de los lenguajes usados del lado del cliente (*HTML5, CSS3, JavaScript*) para interpretar, procesar y mostrar la información recibida desde el servidor.






2018	Chrome 	IE/Edge 	Firefox 	Safari 	Opera 
Diciembre	79.60%	3.80%	10.10%	3.40%	1.70%
Noviembre	79.10%	4.10%	10.20%	3.80%	1.60%
Octubre	79.60%	4.00%	10.10%	3.50%	1.50%

Tabla 1.3 Principales navegadores en el trimestre final del 2018.

Cada uno de los navegadores más utilizados cuenta con la versión móvil, para que los mismos puedan ser utilizados desde smartphones o tablets para acceder al sistema.

1.8.2 FACTIBILIDAD ECONÓMICA

- **COSTO DE HOSTING**

Utilizar el servicio de hosting ya contratado es un requerimiento específico realizado por el director del programa. La principal razón de dicho requerimiento es porque el costo de los servicios no está incluido en el presupuesto del programa, y es financiado personalmente por el director. Tomando en cuenta esto, el gasto que se debe incurrir para contar con la disponibilidad de dichos servicios debe mantenerse en el rango de costos del servicio actual.

Costos de Servicios y Recursos para implementación de Expediente en línea	
Servicio de Hosting compartido Proveedor: Hostinger.es Plan: Web Hosting Premium	\$11.98 / mes
Dominio para acceder a la aplicación Web	\$15 / año
Certificado SSL	Incluido en plan
Lenguaje de programación PHP	PHP License v3.01, (Licencia de Código Abierto)
Servidor Apache	Apache License v2.0, (Licencia de Software libre)
Motor de Base de datos MySQL	GPLv2 (Software libre)

Tabla 1.4 Costo de servicios y software para la implementación.

- **COSTO DE DESARROLLO E IMPLEMENTACIÓN**

El proyecto de desarrollo e implementación del sistema para el expediente en línea será ejecutado como trabajo de grado, por parte de los miembros del equipo de

trabajo; por lo tanto, la FGK sede Santa Ana no incurrirá en gastos para la ejecución del proyecto. Sin embargo, se presenta una estimación del costo en base al trabajo que se llevara a cabo a lo largo del desarrollo del proyecto.

Tabla 1.5 Costos de desarrollo e implementación del sistema.

Costos de Desarrollo e Implementación			
Puesto	Nº de puestos	Salario	Tiempo
Desarrollador Junior: Análisis, diseño, desarrollo e implementación del expediente en línea.	3	\$400 / mes	4 meses
Costo del Sistema de expediente en línea para el programa Oportunidades: \$4,500.			

El valor del salario como desarrollador junior ha sido estimado tomando como base las ofertas laborales para desarrolladores web en sitios de bolsas de trabajo como computrabajo.com y unmejorempelo.com.sv. Y para dicha estimación han sido tomadas en cuenta variables como el lenguaje de programación y el nivel de experiencia. (unmejorempelo.com.sv, 2021).

1.8.3 FACTIBILIDAD OPERATIVA

- **EQUIPO DE DESARROLLO**

Como ha sido descrito anteriormente, el equipo de trabajo cuenta con todas las habilidades técnicas para llevar a cabo la ejecución del proyecto, desde la etapa de obtención y análisis de los requerimientos del sistema, hasta la implementación del mismo. Para el trabajo es necesario apoyarse en múltiples herramientas que permitan realizar el trabajo de manera más fácil y eficiente.

Herramientas de trabajo para el desarrollo de sistema	
Navegador Web	Cualquier navegador web, utilizado para acceder a los servicios en línea de las demás herramientas. Así como también para realizar pruebas de desarrollo del código fuente del sistema. Principales navegadores: Chrome, Firefox, Opera, Safari.
Filezilla	Ciente FTP de código abierto y software libre, utilizado para realizar la transferencia de archivos al servidor del hosting. Soporta los protocolos FTP, SFTP y FTP sobre SSL/TLS (FTPS).
Github	GitHub es una plataforma utilizada para alojar proyectos usando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.
vscode	Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.
workbench	MySQL Workbench es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL.
Google Drive	Servicio de almacenamiento de archivos en la nube ofrecido por Google. Utilizado para unificar los documentos y archivos pertinentes al desarrollo del sistema de expediente en línea.
Google Docs	Servicio de ofimática en la nube ofrecido por Google, para la creación de documentos, hojas de cálculo, presentaciones, formularios, etc.

Tabla 1.6 Herramientas de software para el desarrollo del código fuente.

- **PERSONAL DEL PROGRAMA**

El sistema incurrirá en la modificación de los procedimientos que ya se llevan a cabo actualmente dentro del programa y como se ha establecido en los alcances del proyecto, se limitará a los procedimientos del ámbito académico de la primera fase del programa. Por lo tanto, el personal administrativo y los docentes, no necesitan adquirir nuevas competencias técnicas para trabajar con el sistema; no obstante, en la fase de implementación se contará con un plan de capacitación para dichos usuarios, de modo que puedan aprender a utilizar el nuevo sistema de forma correcta en la ejecución de sus tareas correspondientes para el programa Oportunidades.

1.9 ENTREGABLES

Los diferentes documentos que a continuación se detallarán serán entregados una vez finalizada la implementación del sistema informático:

1. **Documentación General del Proyecto:** Información general de Análisis, Diseño y Desarrollo.
2. **Código Fuente:** Todo el código fuente implementado será entregado.
3. **Base de Datos:** Estructura de la base de datos para el correcto funcionamiento del sistema.
4. **Manual de Usuario:** Manual de ayuda para el fácil manejo del sistema por parte del usuario encargado (secretarias y docentes).
5. **Manual de Administrador:** Manual de guía para la administración y gestión de los diferentes módulos del sistema.
6. **Manual de Capacitación:** Manual auxiliar para la fácil capacitación en el uso del sistema a nuevos involucrados en su manejo.

CAPITULO II: MARCO TEORICO

Capítulo en el cual se realizará la documentación de la obtención y análisis de los requerimientos. Se busca representar de forma precisa las necesidades de los usuarios finales del sistema, realizando una lista en forma de requerimientos, los cuales serán clasificados como funcionales, no funcionales y operativos.

2.1 DOCUMENTACIÓN DE REQUERIMIENTOS

2.1.1 REPORTE DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

En este capítulo se documenta la especificación de los requerimientos del sistema tomando como base el Estándar IEEE 830, 1998. De acuerdo con la IEEE, toda buena especificación de requisitos de software debe incluir toda la información sugerida en dicho estándar, aun cuando no se siga de forma obligatoria con la organización y estructura de la información planteada en el mismo.

De acuerdo con el Estándar de requerimientos se debe preparar un documento dirigido a los usuarios del sistema, el cual debe contar con una serie de apartados que permiten especificar con claridad el propósito, objetivo y alcance del sistema, el funcionamiento que tendrá, el entorno en el que será implementado, así como también los requerimientos específicos del sistema. Dicho documento es el reporte ERS (*Especificación de Requerimientos del Sistema*) que se presenta como ANEXO I del presente proyecto.

El reporte ERS fue entregado al directo y docentes del programa con el fin de poder obtener su validación y determinar si todas las necesidades han sido abarcadas en los requerimientos del sistema y poder tener una guía clara para el equipo de diseño y desarrollo; así también, para que los usuarios tengan una idea clara del propósito y alcance del sistema como producto final.

A continuación, se retoman las principales secciones del reporte ERS que permitirán realizar un análisis de los requerimientos bajo el enfoque de sistemas de información.

- **USUARIOS Y FUNCIONES PRINCIPALES DEL PRODUCTO**



Figura 2.1 Funciones Principales del Sistema EELO

En el ERS se hace referencia al sistema web para el expediente en línea como el producto, y este ha sido nombrado Sistema EELO (*Expediente en Línea Oportunidades*). En la figura 2.1 se plantean las principales funciones que se realizarán dentro del sistema y que sirven como base para determinar las actividades que corresponden a cada uno de los diferentes usuarios en el sistema.

A continuación, se describen con mayor detalle algunos de los puntos abarcados en el ERS que se presentaron de manera puntual para mantener el formato sugerido por el Estándar IEEE 830 y mantener el reporte con información accesible para la interpretación por parte de los usuarios a quienes está dirigido.

Dentro del programa se lleva registro del progreso académico de los estudiantes en 3 áreas distintas:

- ❖ **Calificaciones obtenidas en las materias impartidas por el programa (Notas FGK).**
- ❖ **Calificaciones obtenidas en las materias principales cursadas en el centro escolar de cada alumno (Notas CE).**
- ❖ **Evaluación actitudinal de los estudiantes dentro del programa Oportunidades (RDA – Reporte de desarrollo de Actitudinal).**

El registro de las notas FGK y notas del RDA son realizadas por los docentes que imparten las materias del programa. A cada docente se le asigna la materia que impartirá y con qué secciones de cada año desarrollará dichas materias.

Materias	1º año		2º año		3º año	
	CLASS 2020		CLASS 2019		CLASS 2018	
	C1	C2	B1	B2	A1	A2
Matemática	Lorena Ramirez	Elmer Galdamez	Lorena Ramirez	Elmer Galdamez	Iris Peraza	Iris Peraza
Computación						
Inglés						

Tabla 2.1 Asignación de docentes a materias por año.

- **ASIGNACION DE DOCENTES A SECCIONES POR MATERIA**

En la tabla 2.1 se puede observar la forma en que se realizan las asignaciones cada año. Cada nuevo grupo de alumnos que ingresa cada año se le asigna como nombre el año que terminaran la primera fase del programa; por ejemplo, los alumnos que ingresaron al programa en el 2018 se nombran parte de la Class 2020 ya que ese año es en el que se graduarían de bachillerato y de la primera fase del programa Oportunidades. Cada clase es dividida en dos o tres secciones dependiendo el número de alumnos de la misma. La asignación de los alumnos a las secciones se realiza de forma arbitraria, tomando en cuenta los horarios que convengan a un estudiante.

Por cada materia se realiza la asignación del docente que la impartirá, en cada una de las secciones de los tres años.

- **REGISTRO DE NOTAS FGK**

Como fue mencionado en la descripción de la situación actual, en el capítulo I; los docentes utilizan archivos de Excel para llevar el registro de las calificaciones de los alumnos. En el ANEXO II se incluyen los archivos utilizados para dichos registros, en el tercer periodo del año 2018. Por cada uno de los periodos del año escolar, el docente establece el número de evaluaciones que se realizarán, asignándoles un nombre y el porcentaje que representará de la nota final. Por lo tanto, el docente maneja un archivo de Excel por cada periodo, de cada materia que imparte. El archivo está estructurado de tal manera que forma una cuadrícula con el listado de alumnos de cada sección y las evaluaciones del periodo.

Todas las secciones de una misma Clase, deben realizar las mismas evaluaciones en cada materia. Tomando como ejemplo la asignación de docentes de la tabla 2.1 podemos observar que para las secciones C1 y C2 de la Class 2020, la materia matemática es impartida por diferentes docentes. Sin embargo, ambas secciones deben realizar las mismas evaluaciones para la materia de matemáticas en cada periodo.

- **REGISTRO DE NOTAS RDA**

Para realizar la evaluación actitudinal, se utiliza el mismo método que el de las notas FGK. Los docentes también utilizan un archivo llamado RDA (Reporte de desarrollo actitudinal) para registrar las notas de los alumnos en cada periodo. En este caso, en lugar de las evaluaciones se utilizan criterios de evaluación y cada uno se divide en puntos actitudinales específicos en base a los cuales el docente asigna una nota al alumno de acuerdo a su comportamiento. El listado de criterios actitudinales se encuentra en los archivos Excel de los RDA incluidos en el ANEXO III. Los criterios actitudinales que se toman en cuenta para las evaluaciones son los mismos a lo largo del año; sin embargo, están sujetos a modificaciones y por dicha razón el registro y edición de criterios es presentado como un efecto colateral del requerimiento de registro de notas actitudinales.

Los requerimientos funcionales establecen que se debe realizar la asignación de las secciones que un docente deberá evaluar a nivel actitudinal cada periodo. Sin embargo, después de la presentación del ERS se hizo un cambio de política en el programa, especificando que el docente deberá evaluar todas las secciones con las que interactúa al impartirles una materia. Si un docente imparte clases a una misma sección en más de una materia, deberá realizar la evaluación por cada materia ya que el comportamiento del alumno puede cambiar en base a su interés por la materia. Es por tal razón que el requerimiento de asignación de docentes a secciones para la evaluación actitudinal no ha sido incluido en la tabla 2.3 en el siguiente apartado del presente capítulo.

- **TAREAS DEL ASISTENTE ADMINISTRATIVO**

Las notas CE son registradas por un asistente administrativo, quien recibe las notas directamente de los alumnos y realiza los registros correspondientes. Otras tareas que corresponden al asistente administrativo son las siguientes:

- ❖ **Registrar datos de contacto de los alumnos.**
- ❖ **Registro de las calificaciones del centro escolar de los alumnos, por año y periodo.**
- ❖ **Asignar alumnos a una sección, por año escolar.**
- ❖ **Asignar Docentes a materias por sección.**

El asistente administrativo realiza la mayor parte del trabajo de registro de datos de los alumnos y asignaciones para el año escolar, por lo que tal usuario tiene acceso a la mayor parte de información que está disponible para el administrador, reservándose para dicho usuario, ciertas funciones exclusivas como sería el registro de nuevos usuarios del sistema, realizar los reportes de notas de los alumnos en cada periodo, y aprobar la modificación de las notas que ya han sido registradas.

- **NIVEL ACADÉMICO EN BASE A LOS PROMEDIOS**

Como se observa en la figura 2.1 el monitoreo y control del progreso académico es una de las principales funciones y corresponde al director del programa, quien desempeña el rol de administrador en el sistema. Es por eso que el administrador debe

poder consultar de manera fácil los promedios de las notas que se han descrito anteriormente. Esta tarea la realiza actualmente por medio de un archivo Excel que funciona como colector de notas haciendo referencia a las notas finales registradas en los archivos Excel de cada materia en el caso del colector de notas FGK, o a los promedios de cada criterio actitudinal en el caso del colector de los RDA. Los colectores de notas se incluyen en los anexos correspondientes al tipo de notas.

Tanto el administrador como el alumno deben poder consultar en todo momento el nivel académico en el que se encuentra, en base a la consolidación de sus promedios. En la tabla 2.2 se presenta la clasificación del nivel académico en base a los promedios obtenidos por el alumno. En el ANEXO IV se incluyen las boletas de notas FGK y RDA que se entregan cada periodo a los padres de familia y que solamente muestran el promedio final por materia y el nivel académico actual del alumno.

Nivel académico	
Mastermind	Cuadro de honor, son los alumnos con los mejores promedios. Se incluyen las notas actitudinales
Principal's Honor roll	Todos sus promedios arriba de 9
Honor roll	Todos los promedios arriba de 8

Tabla 2.2 Clasificación del nivel académico de los estudiantes.

Tomando en cuenta todos los puntos anteriores, en la Figura 2.2 se muestra la jerarquía de los usuarios. El administrador como se mencionaba, es el único que cuenta con el acceso para registrar nuevos usuarios en el sistema. Por lo tanto, todos los demás usuarios se encuentran en un segundo nivel, dependiente del administrador.

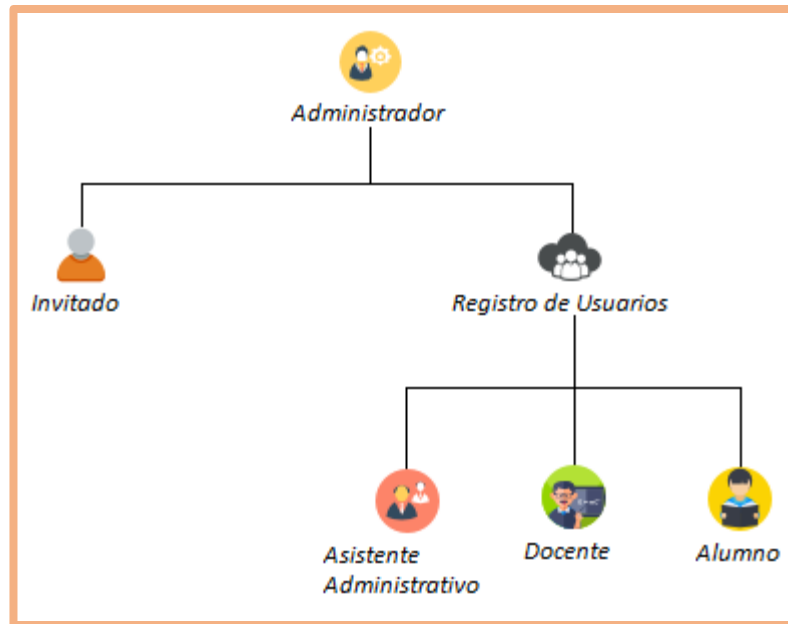


Figura 2.2 Jerarquía de usuarios del Sistema EELO.

El Administrador, asistente administrativo, docente y el alumno son los usuarios que contarán con un rol definido en el sistema. Se representan a un mismo nivel, debido a que su acceso al sistema estará limitado por el mismo rol del usuario. Cada uno de dichos usuarios tendrá una interfaz definida, que les permitirá realizar las tareas y acciones que les corresponden. Dichas actividades se muestran delimitadas por usuario en la figura 2.3.

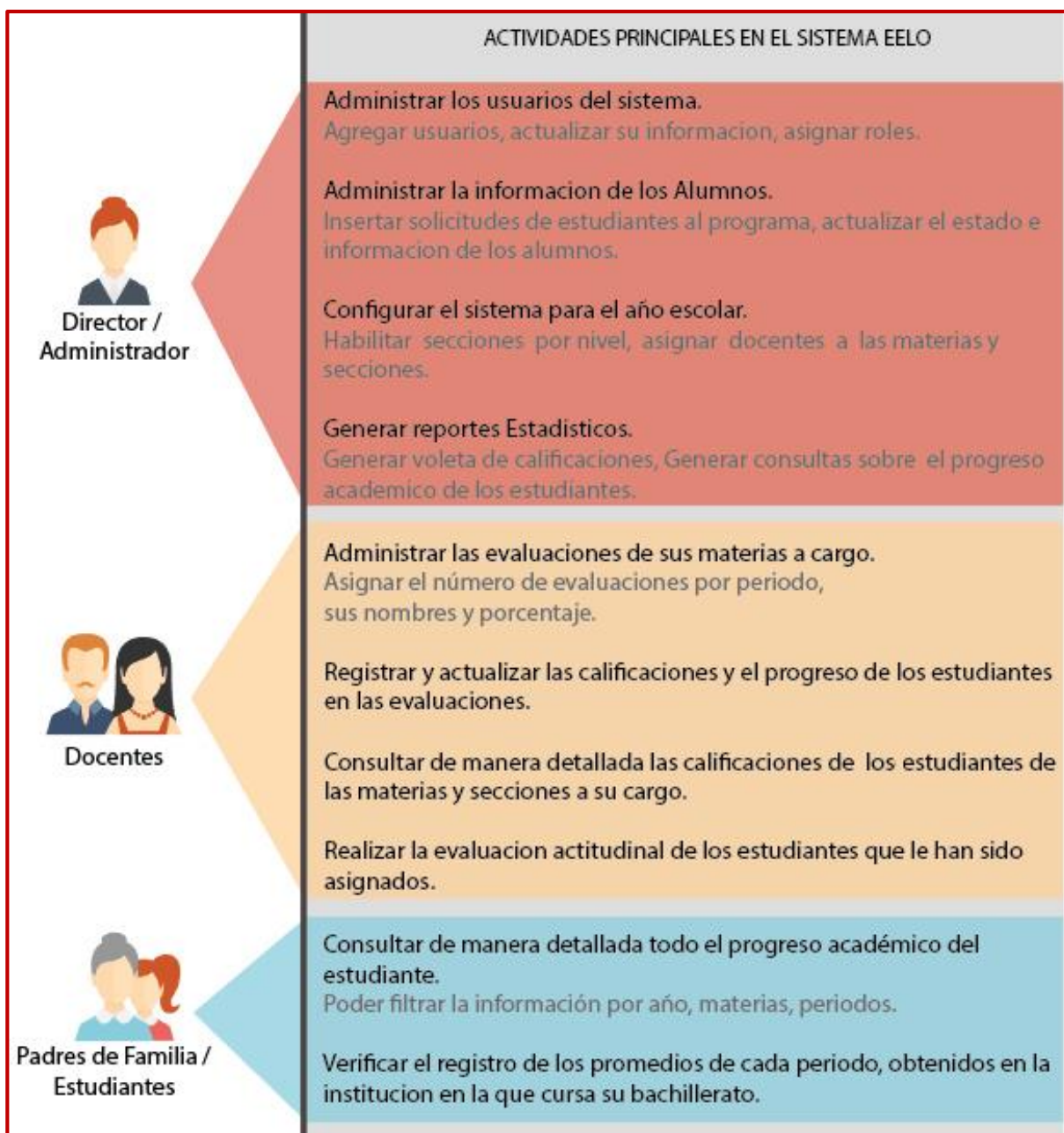


Figura 2.3 Principales actividades del Sistema EELO.

2.2 CASOS DE USO

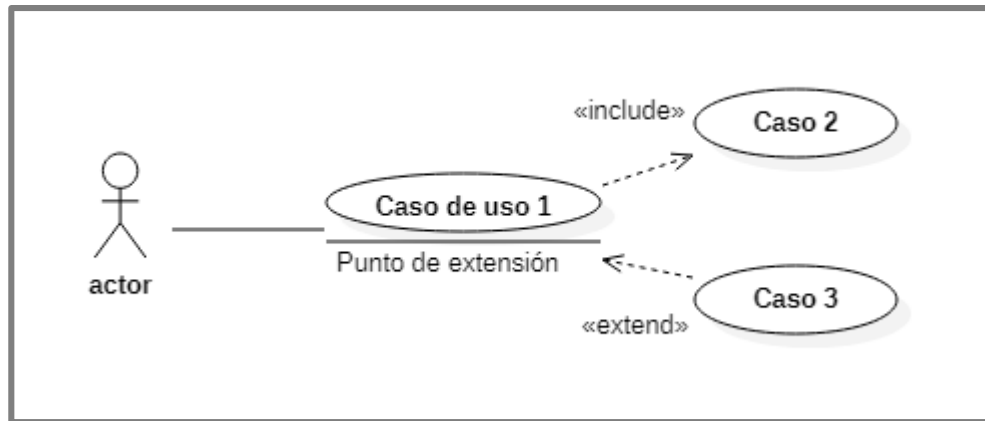


Figura 2.4 Elementos del diagrama de Casos de Uso

Actor	Los actores son similares a las entidades externas; existen fuera del sistema. El término actor se refiere a un rol específico de un usuario del sistema.
Caso de Uso	Representa la acción o tarea que realiza un actor, para conseguir un objetivo determinado.
Asociación	Para conectar un actor con un caso de uso se utiliza una línea sin puntas de flecha.
Extiende	Un caso de uso distinto maneja las excepciones del caso de uso básico. La flecha apunta del caso de uso extendido al básico. El caso 3 extiende el caso de uso 1.
Incluye	Un caso de uso contiene un comportamiento común para más de un caso de uso. La flecha apunta al caso de uso común. El caso de uso 1 incluye el caso 2.
Punto de extensión	Representación de eventos o acciones desencadenantes de la extensión de un caso de uso con otro.

Tabla 2.3 Funciones de los elementos del diagrama de casos de uso.

Gracias al documento de ERS se cuenta con un listado de requerimientos funcionales con información detallada sobre los datos de entrada y salida para llevar a cabo una función y una descripción de cómo se espera que sea realizada. Documentar los requerimientos en el formato sugerido por el Estándar IEE 830 nos

permitió asegurar que los requerimientos por parte de los usuarios han sido comprendidos y abarcados completamente. Para iniciar la fase de análisis de los requerimientos, estos serán representados en un diagrama de casos de uso.

Utilizamos los diagramas de uso como primera herramienta de análisis, ya que el diagrama permite modelar la interpretación de los requerimientos desde la perspectiva del usuario final, ya que son representadas como funciones del sistema sin especificar la forma en que dichas funciones se llevan a cabo dentro del sistema. Un caso de uso provee a los desarrolladores una perspectiva de lo que quieren los usuarios, sin detalles técnicos o implementación. Podemos considerar un caso de uso como una secuencia de transacciones en un sistema (Kendall & Kendall, 2011).

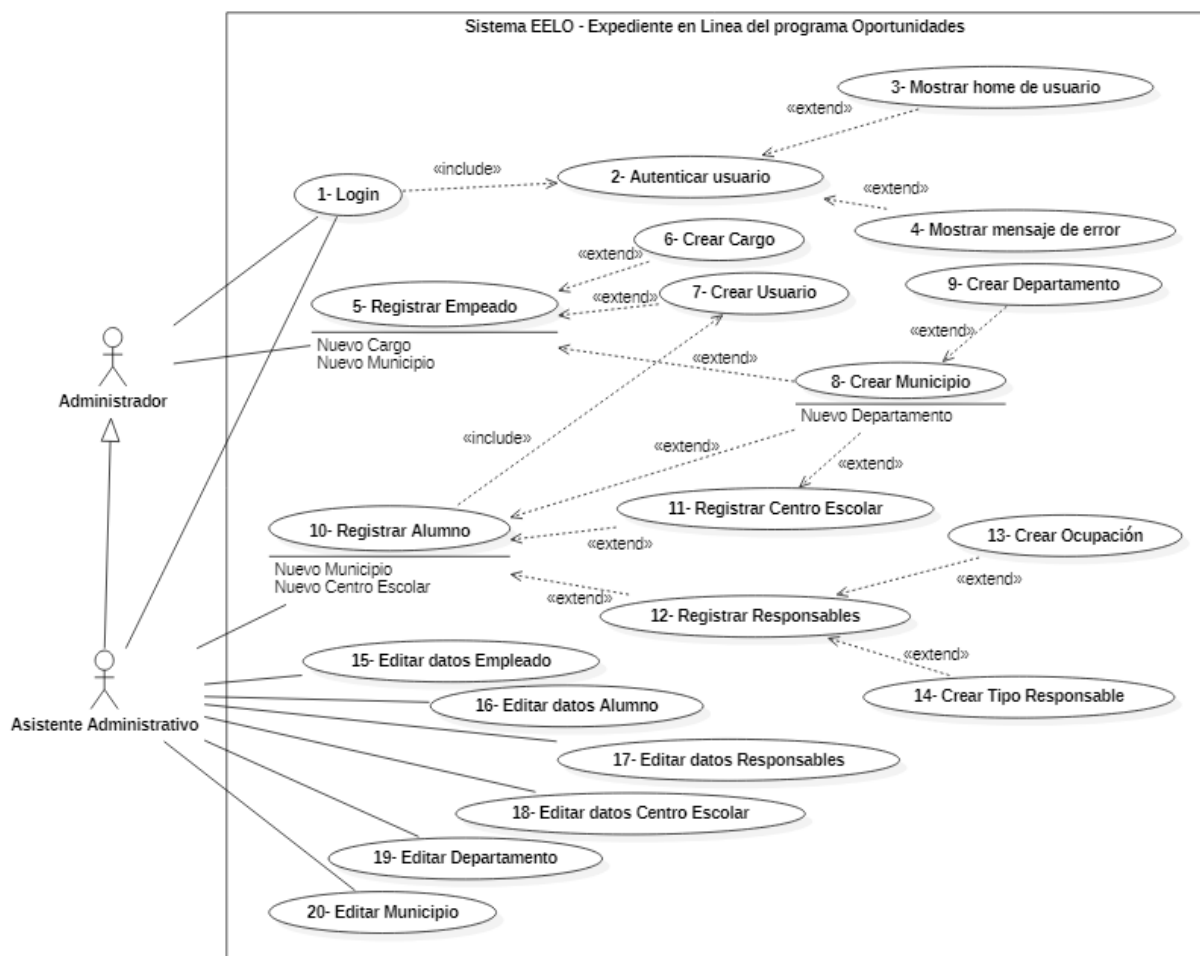


Figura 2.5 Casos de Uso del Administrador del Sistema EELO.

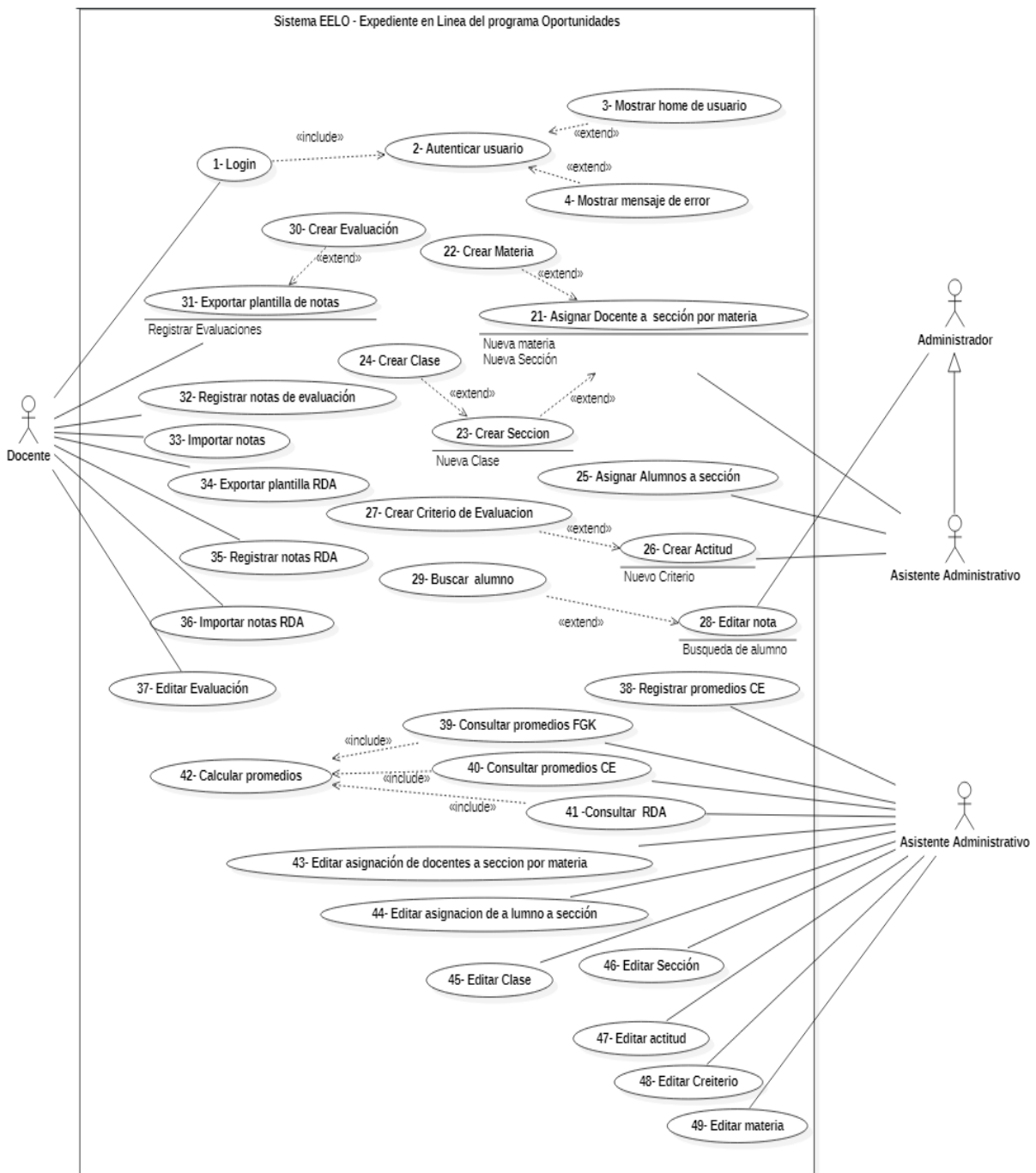


Figura 2.6 Casos de Uso del Administrador y Docente del Sistema EELO.

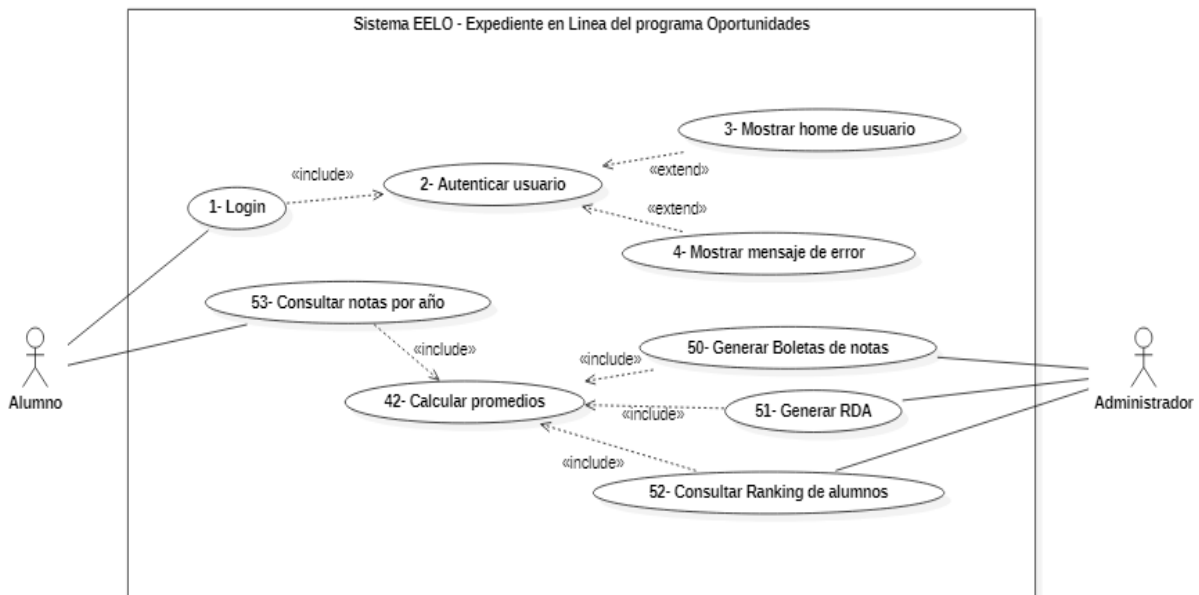


Figura 2.7 Casos de Uso del Administrador y Alumno del Sistema EELO.

NOTAS A LOS CASOS DE USO

- La conexión realizada entre el actor Administrador y el Asistente administrativo es de generalización; de acuerdo a las especificaciones del diagrama de casos de uso esto indica que el asistente administrativo es un tipo específico de un actor más general que es el administrador. En la implementación del diagrama para el Sistema EELO, se ha utilizado la generalización para representar que el Asistente Administrativo cuenta con un rol igual al del Administrador, y todos los casos de uso ejecutados por el Asistente administrativo, pueden ser llevados a cabo también por el Administrador. La asociación de los casos de uso con el actor, solamente se ha hecho con el asistente administrativo para mantener orden y claridad en el diagrama; y se ha realizado la asociación del Administrador solamente con los casos de uso que son exclusivos de dicho usuario y que no corresponden al Asistente administrativo.

- En el diagrama, los puntos de extensión se utilizan para hacer referencia a la acción que desencadena la extensión de un caso de uso. Por ejemplo, el Caso de uso 5 consiste en registrar un empleado en el sistema; para lo cual se debe especificar su cargo. Si el cargo no está registrado, se debe crear un nuevo cargo en el sistema, dando lugar al Caso de uso 6.
- Todos los casos de uso que se generan a partir de un punto de extensión, como los casos 8 y 9, pueden ser ejecutados sin extender un caso de uso inicial. Por ejemplo, el asistente administrativo puede acceder al sistema para registrar uno o más cargos (caso de uso 6), sin que dicha acción dependa del registro de un empleado primero.
- De igual forma, la edición o modificación de datos registrados a partir de un caso de uso, como el registro de un nuevo alumno; es considerado un caso de uso independiente ya que es una acción que se genera en otro momento y no al mismo tiempo que la información es registrada por primera vez.
- En un mismo diagrama se ha duplicado un actor, como el caso del Asistente administrativo en la figura, con la única intención de mantener claridad y orden en el diagrama. Ambas figuras hacen referencia al mismo actor.

Código	Requerimiento Funcional	Casos de Uso
RF-01	Login para ingresar al Sistema.	1 - 4
RF-02	Registro de Estudiantes en el Sistema.	10, 7, 16
RF-02-1	Registro y edición de Departamentos y Municipios en el Sistema.	8 - 9, 19 - 20
RF-02-2	Registro y edición de Instituciones educativas en el Sistema.	11, 18
RF-02-3	Registro y edición de los responsables del estudiante en el Sistema.	12 - 14, 17
RF-03	Registro de Empleados al Sistema.	5 - 6, 7, 15
RF-04	Asignación de Materias a Docentes por año y sección.	21, 43
RF-04-1	Registro y Edición de Materias.	22, 49
RF-04-2	Registro y Edición de Secciones.	23 - 24, 45 - 46
RF-05	Registro de Evaluaciones de materias por periodo.	30
RF-05-1	Edición de las Evaluaciones.	37
RF-06	Asignación de estudiantes a las secciones.	25, 44
RF-07	Registro de Calificaciones.	32
RF-07-1	Edición de calificación con autorización de administrador.	28 - 29
RF-07-2	Exportar plantilla de Excel para el registro de calificaciones.	31
RF-07-3	Importar calificaciones desde un archivo de Excel.	33
RF-08	Mostrar el nivel académico alcanzado por cada estudiante, por periodo.	53
RF-09	Registro de Promedios por periodo de cada materia del Centro Escolar.	38
RF-10	Consulta del promedio global de los estudiantes.	39 - 42, 51
RF-11	Registro y edición de Criterios de evaluación actitudinal.	26 - 27, 47 - 48
RF-12	Registro de evaluaciones actitudinales.	34 - 36
RF-13	Generación de Boleta de notas por alumno, cada periodo.	50
RF-14	Generación de Reporte de carácter por alumno, cada periodo.	51
RF-15	Consulta del expediente de notas del Estudiante.	53

Tabla 2.4 Correspondencia entre los casos de uso y requerimientos funcionales.

En la tabla 2.4 se presenta la relación entre los requisitos de usuario y los casos de uso. Como se mencionó anteriormente dichos requisitos se encuentran detallados en el documento ERS; por lo tanto, al analizarlos como casos de uso se hace énfasis en la serie de pasos que se deben realizar para ejecutar cada uno de ellos.

- **REGISTRO Y EDICIÓN DE DATOS: CRUD**

En informática existe el término *CRUD* como acrónimo de “Crear, Leer, Actualizar y Borrar” en inglés, y que hace referencia a las principales funciones que se realizan en una base de datos por medio de la capa de persistencia de un sistema informático.

De acuerdo a los diagramas de casos de uso podemos observar que algunas de las interacciones que realiza el actor con el sistema es para llevar a cabo las funciones básicas de creación y actualización de una serie de datos específicos; los cuales, desde un primer análisis han sido identificados como entidades para formar parte del modelado de la base de datos del sistema. Utilizamos el término *Entidad* para hacer referencia a cualquier persona, cosa, objeto o concepto del mundo real, el cual cuenta con atributos o características que pueden ser representadas como datos.

- Cargo.
- Empleado.
- Departamento.
- Municipio.
- Centro Escolar.
- Alumno.
- Responsable.
- Class.
- Sección.
- Materia.
- Criterio de Evaluación
actitudinal.
- Actitud.
- Evaluación.

A continuación se presenta la descripción de los pasos en que se desarrollan los casos de uso identificados. Se plantea el escenario principal de cada caso de uso, desde la perspectiva de una secuencia lógica de interacciones entre actor y sistema.

Registrar y editar datos de empleado	
Objetivo:	Registrar y modificar la información de un empleado en la base de datos del sistema.
Actor:	Administrador.
Alias:	usuario
Pre-condición:	El Administrador debe haber iniciado sesión, haciendo login en el sistema.
Escenario Principal	Registro de nuevo empleado
	a. El usuario selecciona desde el menú, la opción de empleados.
	b. El sistema carga en pantalla el listado de los empleados registrados. Carga en pantalla el formulario para ingresar los datos del nuevo empleado. Carga en el formulario los cargos registrados en la base de datos.
	c. El usuario ingresa los datos del nuevo registro: nombre, apellidos, sexo, fecha de nacimiento, DUI, NIT, Carnet, teléfono, municipio, dirección, estado.
	d. El usuario confirma el registro del nuevo empleado.
	e. El sistema realiza la verificación del formato y tipo de los datos ingresados.
	f. El sistema envía la información a la base de datos del nuevo empleado.
	g. El sistema crea un usuario para el nuevo empleado si el cargo es: docente, administrador o asistente administrativo.
	h. El sistema muestra un mensaje al usuario de registro exitoso.
	post-condición
	Nuevo empleado registrado en el sistema.
	flujo alternativo
	g.1 El sistema muestra mensaje de error al realizar el registro en la base de datos.
	g.2 El mensaje describe el tipo de error.
	Actualización de empleado
	a. El usuario selecciona desde el menú, la pestaña de empleados.
	b. El sistema carga en pantalla el listado de los empleados registrados. Carga en pantalla el formulario para ingresar los datos del nuevo empleado. Carga en el formulario los cargos registrados en la base de datos.
	c. El usuario selecciona en la lista el empleado cuya información desea modificar.
	d. El sistema muestra una ventana emergente con todos los datos actuales del empleado seleccionado.
	e. El usuario edita los datos del empleado y confirma la actualización de los mismos.
	f. El sistema realiza la verificación del formato y tipo de los datos ingresados.
	g. El sistema envía los datos editados a la base de datos.
	h. El sistema muestra un mensaje de actualización exitosa.
	post-condición
	Información actualizada de un empleado determinado.
	flujo alternativo
	g.1 El sistema muestra mensaje de error al realizar el registro en la base de datos.
	g.2 El mensaje describe el tipo de error.

Tabla 2.5 Caso de Uso – Registro, Edición de Empleados.

Registrar y editar datos de alumno	
Objetivo:	Registrar y modificar la información de un alumno en el sistema para poder llevar a cabo la administración de su expediente académico.
Actor:	Administrador, Asistente Administrativo.
Alias:	usuario
Pre-condición:	El Administrador debe haber iniciado sesión, haciendo login en el sistema.
Escenario Principal	Registro de nuevo Alumno
	a. El usuario selecciona desde el menú, la opción de alumnos.
	b. El sistema carga en pantalla el listado de los alumnos registrados de las clases activas y la opción para acceder al formulario de nuevo registro.
	c. El usuario abre el formulario para registrar un nuevo alumno.
	d. El usuario ingresa todos los datos del nuevo alumno en el formulario: carnet, nombre, apellidos, fecha de nacimiento, sexo, dirección, municipio, email, teléfono, centro escolar, turno educativo, foto.
	e. El usuario confirma el registro del nuevo alumno.
	f. El sistema realiza la verificación del formato y tipo de los datos ingresados.
	g. El sistema envía la información a la base de datos del nuevo alumno.
	h. El sistema crea un usuario para el nuevo alumno.
	i. El sistema muestra un mensaje al usuario de registro exitoso.
	post-condición
	Nuevo alumno registrado en el sistema.
	flujo alternativo
	g.1 El sistema muestra mensaje de error al realizar el registro en la base de datos.
	g.2 El mensaje describe el tipo de error.
	Actualización de datos de alumno
	a. el usuario selecciona desde el menú, la opción de alumnos.
	b. El sistema carga en pantalla el listado de los alumnos registrados de las clases activas.
	c. El usuario selecciona en la lista el alumno cuya información desea modificar.
	d. El sistema carga el formulario con todos los datos actuales del alumno seleccionado.
	e. El usuario edita los datos del alumno y confirma la actualización de los mismos.
	f. El sistema realiza la verificación del formato y tipo de los datos ingresados.
	g. El sistema envía los datos editados a la base de datos.
h. El sistema muestra un mensaje de actualización exitosa.	
post-condición	
Información actualizada de un alumno.	
flujo alternativo	
g.1 El sistema muestra mensaje de error al realizar el registro en la base de datos.	
g.2 El mensaje describe el tipo de error.	

Tabla 2.6 Caso de Uso – Registro, Edición de Alumnos.

Registrar responsable del alumno	
Objetivo:	Registrar la información de los adultos responsables de un alumno.
Actor:	Administrador, Asistente Administrativo.
Alias:	usuario
Pre-condición:	El Administrador debe haber iniciado sesión, haciendo login en el sistema.
Escenario Principal	Registrar responsable del alumno
	a. El usuario selecciona desde el menú, la opción de alumnos.
	b. El sistema carga en pantalla el listado de los alumnos registrados de las clases del año en curso.
	c. El usuario selecciono de la lista, el alumno cuyo responsable será registrado.
	d. El sistema carga la vista con todos los datos actuales del alumno seleccionado.
	e. El usuario abre el formulario para el registro del responsable del alumno.
	f. El usuario ingresa todos los datos de contacto del responsable del alumno.
	g. El usuario confirma el registro de los datos.
	h. El sistema verifica el formato y tipo de los datos ingresados.
	i. El sistema envía los datos editados a la base de datos.
	j. El sistema muestra un mensaje de registro exitoso.
	post-condición
	Información del adulto responsable del alumno registrada en el sistema.
flujo alternativo	
f.1 El usuario registra una nueva ocupación que corresponda a l responsable que desea registrar.	
f.2 El usuario registra un nuevo tipo de responsable (parentesco) que corresponda al responsable que desea registrar.	
i.1 El sistema muestra mensaje de error al realizar el registro en la base de datos.	

Tabla 2.7 Caso de Uso – Registro de datos del responsable del alumno

Registro y edición de Clase	
Objetivo:	Registrar nueva clase que representa al grupo de alumnos que ingresaron al programa en un año específico.
Actor:	Administrador, Asistente Administrativo.
Alias:	Usuario.
Pre-condición:	El Administrador debe haber iniciado sesión, haciendo login en el sistema.
Escenario Principal	Registrar Clase
	a. El usuario selecciona desde el menú del año en curso, la opción Clases.
	b. El sistema carga en pantalla el listado de clases registradas en el sistema y el formulario para ingresar una nueva clase.
	c. El usuario especifica el año de ingreso de la clase y confirma el registro.
	d. El sistema verifica el formato de los datos ingresados.
	e. El sistema calcula el año de salida de la clase y lo asigna como identificador de la clase.
	f. El sistema envía el registro de la nueva clase a la base de datos.
	g. El sistema muestra un mensaje de registro exitoso al usuario.
	post-condición
	Nueva clase registrada en el sistema.
	flujo alternativo
f.1 El sistema muestra mensaje de error en la base de datos.	

Editar Clase	
a.	El usuario selecciona desde el menú año en curso, la opción Clases.
b.	El sistema carga en pantalla el listado de clases registradas en el sistema y el formulario para ingresar una nueva clase.
c.	El usuario selecciona la clase que se quiere modificar.
d.	El sistema muestra una ventana emergente con los datos de la clase seleccionada.
e.	El usuario introduce las modificaciones de los datos.
f.	El usuario confirma la modificación de la clase.
g.	El sistema verifica el formato de los datos ingresados.
h.	El sistema envía las modificaciones de la clase a la base de datos.
i.	El sistema muestra un mensaje de registro exitoso al usuario.
post-condición	
Datos de la clase actualizados en la base de datos.	
flujo alternativo	
h.1	El sistema muestra mensaje de error en la base de datos.

Tabla 2.8 Caso de Uso – Registro, Edición de Clases.

Como se puede observar en las tablas de los Casos de Uso se han combinado los casos de registro y edición de una misma entidad. Por ejemplo; para el alumno, el registro de datos se representa en el caso de uso 10 y la edición de los datos en el caso de uso 16 en el diagrama de la figura 2.5. En el diagrama se representan de esa forma porque las acciones que dan lugar a cada caso de uso se ejecutan en momentos diferentes.

Los casos de uso de registro y edición de datos se han presentado de forma que se pueda observar que todos los casos de uso relacionados con la gestión de datos de una misma entidad como el alumno, el empleado o la clase de alumnos, se pueden implementar dentro del sistema, siguiendo un mismo formato o modelo de pasos. Por tal razón es que anteriormente se hacía referencia al término *CRUD* ya que representa las funciones que se llevan a cabo para la gestión de datos de una entidad y cada función puede ser presentada como un caso de uso en el sistema.

Login	
Objetivo:	Que un usuario ingresar al sistema EELO para llevar a cabo las tareas que le corresponden.
Actor:	Administrador, Asistente Administrativo, Alumno, Docente.
Alias:	Usuario.
Pre-condición:	El usuario debe estar registrado en el sistema.
Escenario Principal	a. El usuario ingresa la dirección URL del login del sistema desde un navegador web.
	b. El sistema devuelve la vista para ingresar las credenciales del usuario: Carnet y contraseña.
	c. El usuario ingresa sus credenciales.
	d. El sistema realiza la autenticación del usuario.
	e. El sistema realiza el inicio de sesión del usuario.
	f. El sistema devuelve la vista de inicio correspondiente al rol del usuario logueado.
	post-condición
	El usuario se encuentra ha iniciado sesión en el sistema, con su interfaz de inicio cargada.
flujo alternativo	
i.1 El sistema muestra mensaje de que las credenciales no corresponden a usuarios registrados en el sistema.	

Tabla 2.9 Caso de Uso – Login al sistema EELO.

Asignar alumnos a sección	
Objetivo:	Realizar la asignación de alumnos a las diferentes secciones de una clase para el año en curso.
Actor:	Administrador, Asistente Administrativo.
Alias:	Usuario.
Pre-condición:	El usuario debe estar logueado en el sistema.
	Deben haber alumnos registrados en las clases activas del año en curso.
	Deben estar creadas las secciones de cada clase para el año en curso.
Escenario Principal	Asignación de alumnos a secciones
	a. El usuario selecciona desde el menú año en curso, la opción asignación de alumnos a secciones.
	b. El sistema devuelve la vista para realizar las asignaciones.
	c. El usuario selecciona una de las clases del año en curso.
	d. El sistema carga el listado de alumnos y las secciones correspondientes a la Clase seleccionada.
	e. El usuario marca por cada alumno la sección a la que será asignado.
	f. El usuario especifica cuáles de los alumnos serán considerados flotantes.
	g. El usuario confirma las asignaciones realizadas.
	h. El sistema guarda las asignaciones en la base de datos.
	i. El sistema muestra un mensaje de asignaciones realizadas con éxito.
	post-condición
todos los alumnos de una clase asignados en las diferentes secciones	
flujo alternativo	
h.1 El sistema muestra mensaje detallado de error en la base de datos.	
Cambio de sección para alumno	

	a. El usuario selecciona desde el menú año en curso, la opción asignación de alumnos a secciones.
	b. El sistema devuelve la vista para realizar las asignaciones.
	c. El usuario selecciona la clase a la que corresponde el alumno que se cambiara de sección.
	d. El sistema carga el listado de alumnos y las secciones correspondientes a la Clase seleccionada.
	e. El usuario selecciona el alumno y habilita la edición de la sección asignada.
	f. El usuario marca la nueva sección a la que será asignado el alumno.
	g. El usuario confirma el cambio de sección realizado.
	h. El sistema realiza la actualización en la base de datos.
	i. El sistema realiza un registro en la bitácora de cambios de sección en la base de datos.
	i. El sistema muestra un mensaje de asignación realizada con éxito.
	flujo alternativo
	h.1 El sistema muestra mensaje detallado de error en la base de datos.
	post-condición
	todos los alumnos de una clase asignados en las diferentes secciones

Tabla 2.10 Caso de Uso – Registro, Edición de asignaciones de alumnos.

Asignación de docentes a sección	
Objetivo:	Realizar la asignación de los docentes que impartirán las materias del programa a cada sección del año en curso.
Actor:	Administrador, Asistente Administrativo.
Alias:	Usuario.
Pre-condición:	El usuario debe estar logueado en el sistema.
	Los docentes deben estar registrados en el sistema.
	Deben estar creadas las secciones de cada clase para el año en curso.
	Deben estar registradas las materias a impartir
Escenario Principal	Asignación de docentes a secciones
	a. El usuario selecciona desde el menú año en curso, la opción asignación de docentes a secciones.
	b. El sistema devuelve la vista para realizar las asignaciones de los docentes y carga en pantalla el listado de asignaciones registradas para el año en curso.
	c. El usuario realiza la selección del docente, la materia y la sección; para hacer la asignación.
	d. El usuario confirma la asignación realizada.
	e. El sistema registra la asignación realizada en la base de datos.
	f. El sistema muestra un mensaje de asignación realizada con éxito.
	post-condición
	Cada docente tiene asignadas las secciones que le corresponden, para las materias que impartirá en el año en curso.
	flujo alternativo
	i.1 El sistema muestra mensaje detallado de error con la base de datos.
	Cambio de asignación de docente

	a. El usuario selecciona desde el menú año en curso, la opción asignación de docentes a secciones.
	b. El sistema devuelve la vista para realizar las asignaciones de los docentes y carga en pantalla el listado de asignaciones registradas para el año en curso.
	c. El usuario selecciona la sección y materia que tendrá un cambio de docente.
	d. El usuario hace el cambio de docente para la sección, especificando en qué periodo del año escolar se realiza el cambio.
	e. El usuario realiza la confirmación del cambio de docente.
	f. El sistema realiza la actualización de la asignación en la base de datos.
	g. El sistema realiza un registro en la bitácora de asignaciones de docentes para mantener registro del periodo en el que el docente anterior tuvo asignada la sección.
	i. El sistema muestra un mensaje de actualización realizada con éxito.
	post-condición
	todos los alumnos de una clase asignados en las diferentes secciones
	flujo alternativo
	g.1 El sistema muestra mensaje de error de conexión con la base de datos.

Tabla 2.11 Caso de Uso – Registro, Edición de asignación de docentes.

Registro de Evaluaciones	
Objetivo:	Llevar el registro del progreso académico en el centro escolar de cada alumno.
Actor:	Docente.
Alias:	Usuario.
Pre-condición:	El usuario debe estar logueado en el sistema.
	Deben estar creadas las clases del año en curso.
	Deben estar realizadas las asignaciones por materia de los docentes a las secciones.
Escenario Principal	Crear evaluaciones por periodo.
	a. El sistema carga el menú del usuario con las materias que tiene asignadas y las secciones que le corresponden por dichas materias.
	b. El usuario selecciona en el menú la materia y sección para la que se crearán las evaluaciones.
	c. El sistema carga la vista para realizar la gestión de las evaluaciones y notas para la sección seleccionada.
	d. El usuario selecciona el periodo del año escolar correspondiente a las evaluaciones que serán creadas.
	e. El usuario abre el formulario para gestionar las evaluaciones por periodo.
	f. El usuario especifica el nombre y el porcentaje de la evaluación.
	g. El usuario confirma el registro de la evaluación.
	h. El sistema verifica el formato de los datos y los envía a la base de datos.
	i. El sistema muestra un mensaje de evaluación creada con éxito.
	post-condición
	Todas las secciones de una misma clase, tendrán registradas las evaluaciones que le corresponden en una materia y periodo determinado.
	flujo alternativo
	h.1 El sistema muestra mensaje detallado de error en la base de datos.
	Edición de las evaluaciones
a. El sistema carga el menú del usuario con las materias que tiene asignadas.	
b. El usuario selecciona en el menú la materia y sección para la que se crearán las evaluaciones.	
c. El sistema carga la vista para realizar la gestión de las evaluaciones y notas para la sección seleccionada.	

	d. El usuario selecciona el periodo del año escolar correspondiente a las evaluaciones que serán creadas.
	e. El usuario abre el formulario para gestionar las evaluaciones por periodo.
	f. El sistema carga las evaluaciones creadas para el periodo especificado.
	g. El usuario identifica la evaluación a modificar, actualiza el nombre y porcentaje de la evaluación y confirma la modificación.
	h. El sistema verifica el formato de los datos y los envía a la base de datos.
	i. El sistema muestra un mensaje de actualización realizada con éxito.
	post-condición
	Todas las secciones de una misma clase, tendrán registradas las evaluaciones que le corresponden en una materia y periodo determinado.
	flujo alternativo
	h.1 El sistema muestra mensaje de error en la base de datos.

Tabla 2.12 Caso de Uso – Registro de evaluaciones.

Consulta de promedios de los alumnos	
Objetivo:	Conocer el progreso académico de un alumno a nivel de calificaciones.
Actor:	Administrador, asistente administrativo.
Alias:	Usuario, director.
Pre-condición:	El usuario debe estar logueado en el sistema.
	Los alumnos deben contar con notas registradas del centro escolar, del programa oportunidades y de las evaluaciones actitudinales.
Consulta de Promedios	
Escenario Principal	a. El usuario selecciona en el menú Record de alumnos el tipo de notas que desea consultar: notas FGK, notas CE o notas actitudinales.
	b. El sistema devuelve la vista con los listados de alumnos cargados y los promedios finales (promedio global) por periodo y materia de cada alumno.
	c. El usuario selecciona el año escolar y la clase de los alumnos que desea consultar.
	d. El sistema filtra el listado de los alumnos con el año y clase especificados.
	post-condición
	El director puede saber a detalle el progreso académico de los alumnos de una clase o un alumno en específico.
	flujo alternativo
	d.1 El usuario selecciona un periodo específico para ver el detalle de promedios por materia.

Tabla 2.13 Caso de Uso – Consulta de promedios.

Exportar plantilla de notas FGK	
Objetivo:	Generar una plantilla de Excel para poder ir registrando de forma parcial las notas por evaluación de los alumnos de una sección.
Actor:	Docente.
Alias:	Usuario.
Pre-condición:	El usuario debe estar logueado en el sistema.
	Deben estar realizadas las asignaciones de los alumnos a las secciones.
	Deben estar realizadas las asignaciones por materia de los docentes a las secciones.
	Deben estar creadas las evaluaciones para el periodo del que se generará la plantilla.
Escenario Principal	Exportar plantilla de notas FGK
	a. El sistema carga el menú del usuario con las materias que tiene asignadas y las secciones que le corresponden por dichas materias.
	b. El usuario selecciona en el menú la materia y sección de las que se generará la plantilla.
	c. El sistema carga el listado de alumnos correspondiente a la sección seleccionada.
	d. El usuario selecciona el periodo correspondiente.
	e. El sistema carga las evaluaciones creadas para el periodo seleccionado, creando una cuadrícula de notas por alumnos en cada evaluación.
	f. El usuario confirma la exportación de la plantilla de notas.
	g. El sistema genera el archivo de Excel con la cuadrícula de notas por alumno en cada evaluación.
	h. El usuario descarga el archivo generado.
	post-condición
	El docente cuenta con un archivo para poder ir registrando las notas de los alumnos fuera del sistema EELO
	flujo alternativo
	g.1 El sistema muestra mensaje de error al generar el archivo.

Tabla 2.14 Caso de Uso – Exportar plantilla de notas FGK.

Importar notas FGK	
Objetivo:	Importar las notas de una materia y periodo específico de una sección desde un archivo de Excel.
Actor:	Docente.
Alias:	Usuario.
Pre-condición:	El usuario debe estar logueado en el sistema.
	Deben estar realizadas las asignaciones de los alumnos a las secciones.
	Deben estar realizadas las asignaciones por materia de los docentes a las secciones.
	El archivo de Excel debe mantener el formato con el que fue generado desde el sistema.
Importar notas FGK	a. El sistema carga el menú del usuario con las materias que tiene asignadas y las secciones que le corresponden por dichas materias.

	b. El usuario selecciona en el menú la materia y sección de las que se importarán las notas.
	c. El sistema carga el listado de alumnos correspondiente a la sección seleccionada.
	d. El usuario especifica el periodo correspondiente al archivo que será importado.
Escenario Principal	e. El usuario carga el archivo con las notas que corresponde a la sección y periodo especificados de la materia.
	f. El usuario confirma la importación del archivo.
	g. El sistema lee el archivo y envía las notas registradas en el archivo a la base de datos del sistema.
	h. El sistema muestra un mensaje de importación de notas realizada con éxito y muestra la cuadrícula de notas actualizada en pantalla.
	post-condición
	El docente realiza la actualización de las notas de todos los alumnos de una sección en una o varias evaluaciones en una sola acción.
	flujo alternativo
	g.1 El sistema muestra mensaje de error al importar el archivo.

Tabla 2.15 Caso de Uso – Importar notas FGK.

Registro de notas FGK	
Objetivo:	El docente realiza el registro de las notas obtenidas por los alumnos de una sección en una evaluación de una materia y periodo determinado.
Actor:	Docente.
Alias:	Usuario.
Pre-condición:	El usuario debe estar logueado en el sistema.
	Deben estar realizadas las asignaciones de los alumnos a las secciones.
	Deben estar realizadas las asignaciones por materia de los docentes a las secciones.
	Deben estar creadas las evaluaciones para el periodo del que se generará la plantilla.
Escenario Principal	a. El sistema carga el menú del usuario con las materias que tiene asignadas y las secciones que le corresponden por dichas materias.
	b. El usuario selecciona en el menú la materia y sección de las que realizará el registro de las notas.
	c. El sistema carga el listado de alumnos correspondiente a la sección seleccionada.
	d. El usuario especifica el periodo correspondiente a la evaluación de la que se registrarán las notas.
	e. El usuario habilita el registro de las notas para la evaluación seleccionada.
	f. El usuario introduce la nota por cada alumno del listado.
	g. El usuario confirma el registro de notas realizado.
	h. El sistema realiza el registro de las notas en la base de datos.
	i. El sistema muestra un mensaje de registro de notas realizado con éxito.
	post-condición
	El docente realiza el registro de las notas de una evaluación específica para una sección de alumnos.
flujo alternativo	
	h.1 El sistema muestra mensaje de error en la base de datos.

Tabla 2.16 Caso de Uso – Registro de notas FGK.

Edición de notas FGK		
Objetivo:	El docente realiza el registro de las notas obtenidas por los alumnos de una sección en una evaluación de una materia y periodo determinado.	
Actor:	Administrador.	
Alias:	Usuario.	
Pre-condición:	El usuario debe estar logueado en el sistema.	
	Deben estar realizadas las asignaciones de los alumnos a las secciones.	
	El alumno debe tener notas registradas en una materia determinada.	
Escenario Principal	a. El usuario selecciona en el menú Record de alumnos la opción de editar notas FGK.	
	b. El usuario especifica el nombre del alumno.	
	c. El sistema confirma la existencia del alumno en la base de datos.	
	d. El usuario especifica la materia y periodo que corresponde a la nota a modificar.	
	e. El sistema carga las notas por evaluación correspondientes a la materia y periodo especificados del año en curso.	
	f. El usuario edita la nota correspondiente y confirma la modificación realizada.	
	g. El sistema registra la modificación en la base de datos.	
	h. El sistema muestra un mensaje de actualización de notas realizada con éxito.	
	post-condición	
	El expediente del alumno se encuentra actualizado con las notas que dicho alumno ha obtenido en las evaluaciones.	
flujo alternativo		
g.1 El sistema muestra mensaje de error al realizar los registros en la base de datos.		

Tabla 2.17 Casos de uso - Edición de notas FGK

Registro de Notas del Centro Escolar		
Objetivo:	Llevar el registro del progreso académico en el centro escolar de cada alumno.	
Actor:	Administrador, Asistente Administrativo.	
Alias:	Usuario.	
Pre-condición:	El usuario debe estar logueado en el sistema.	
	Los alumnos deben estar registrados en el sistema.	
Escenario Principal	Registro de notas del Centro Escolar	
	a. El usuario selecciona desde el menú record de alumnos, la pestaña asignación de promedios del centro escolar.	
	b. El sistema devuelve la vista para realizar el registro y edición de las notas del centro escolar de los alumnos. Carga los listados de alumnos que ya cuentan con registros de notas, divididos en pestañas por año: noveno, primero y segundo año de bachillerato.	
	c. El usuario abre el formulario para ingresar las notas.	
	d. El usuario ingresa el nombre del alumno y especifica el año y periodo de las notas que ingresará.	
	e. El sistema confirma la existencia del alumno en el sistema.	
	f. El usuario ingresa cada una de las notas de las materias para el año y periodo especificados.	
	g. El usuario confirma el registro de las notas.	
	h. El sistema guarda las notas en la base de datos.	
	i. El sistema muestra un mensaje de registro de notas realizado con éxito.	

	post-condición
	Expediente de alumno actualizado con las notas del centro escolar por año, periodo y materia.
	flujo alternativo
	h.1 El sistema muestra mensaje de error en la base de datos.
	Edición de notas del centro escolar
	a. El usuario selecciona desde el menú record de alumnos, la pestaña asignación de promedios del centro escolar.
	b. El sistema devuelve la vista para realizar el registro y edición de las notas del centro escolar de los alumnos. Carga los listados de alumnos que ya cuentan con registros de notas, divididos en pestañas por año: noveno, primero y segundo año de bachillerato.
	c. El usuario selecciona el año escolar que corresponde a las notas que serán modificadas.
	d. El usuario identifica al alumno en el listado y selecciona el periodo correspondiente a las notas que serán modificadas.
	e. El sistema muestra el formulario con las notas actuales para realizar la edición.
	f. El usuario actualiza las notas por materias y confirma la actualización realizada.
	g. El sistema verifica el formato de los datos y realiza la actualización en la base de datos.
	i. El sistema muestra un mensaje de actualización realizada con éxito.
	post-condición
	Expediente de alumno actualizado con las notas del centro escolar por año, periodo y materia.
	flujo alternativo
	g.1 El sistema muestra mensaje de error de conexión con la base de datos.

Tabla 2.18 Caso de Uso – Registro de notas del Centro Escolar.

Consulta de expediente académico	
Objetivo:	El alumno y su padre de familia pueda ver en detalle todas las notas obtenidas en el programa oportunidades y en su centro escolar.
Actor:	Alumno.
Pre-condición:	El usuario debe estar logueado en el sistema.
Escenario Principal	a. El sistema carga el menú del alumno los tipos de notas por cada año escolar
	b. El alumno selecciona el tipo de notas que desea consultar (FGK, CE, RDA) de un año especificado.
	c. El sistema muestra las notas obtenidas en cada una de las materias por cada periodo del año especificado.
	post-condición
	El alumno conoce a detalle su progreso académico en el programa oportunidades.
	flujo alternativo
	c1. El alumno selecciona el periodo que desea consultar.
c2. El sistema filtra las notas por el periodo seleccionado.	

Tabla 2.19 Caso de Uso – Consulta de expediente académico.

Generar Boletas de notas FGK y RDA	
Objetivo:	El director genera las boletas de notas obtenidas por los alumnos en cada periodo, para ser entregadas a los responsables en las reuniones de padre de familia.
Actor:	Administrador.
Alias:	Usuario.
Pre-condición:	Todas las notas de los alumnos deben estar registradas en el sistema.
Escenario	a. El usuario selecciona en el menú "Record de alumnos" la opción Generar boletas de notas.
	b. El sistema carga la vista correspondiente con la lista de todos los alumnos.
	c. El usuario especifica la clase y sección de la cual serán generadas todas las boletas de los alumnos.
	d. El sistema filtra el listado de alumnos con los datos especificados.
	e. El usuario marca en el listado los alumnos a los que se le generarán las boletas.
	f. El usuario especifica el tipo de notas para las boletas (FGK o RDA).
	f. El usuario confirma la creación del documento.
	g. El sistema genera un solo documento PDF con las boletas de los alumnos seleccionados y pide al usuario que descargue dicho documento.
	h. El usuario descarga el documento con todas las boletas de notas.
	post-condición
El director obtiene un documento con todas las boletas de notas de los alumnos de un periodo determinado.	
flujo alternativo	
g1. El sistema muestra mensaje de error al generar el documento.	

Tabla 2.20 Caso de Uso – Generar boletas de notas FGK y RDA.

Actualizar las credenciales de Usuario	
Objetivo:	Permitir a los Administradores y docentes del Sistema EELO actualizar sus credenciales de usuario.
Actor:	Administrador, Asistente Administrativo, Docente y Alumno.
Alias:	Usuario.
Pre-condición:	El usuario debe estar registrado en el sistema y haber iniciado sesión con sus credenciales actuales.
Escenario	a. El usuario ingresa al sistema ingresando sus credenciales actuales: nombre de usuario y contraseña.
	b. El sistema carga la vista de inicio correspondiente de acuerdo al cargo del usuario.
	c. El usuario selecciona la pestaña configuración de usuario dentro del menú de navegación.
	d. El sistema carga un formulario para validar las credenciales actuales e ingresar el nuevo nombre de usuario o nueva contraseña.
	e. El usuario confirma la actualización de las credenciales.
	f. El sistema realiza la actualización de las credenciales en la base de datos.
	post-condición
	Credenciales del usuario actualizadas en la base de datos.
	flujo alternativo
	f1. El sistema muestra mensaje de error al realizar conexión con la base de datos.

Tabla 2.21 Caso de Uso – Actualizar credenciales de usuario.

Con las tablas que presentan el escenario principal de cada uno de los casos de uso se logra tener una idea más clara y detallada de cómo estos se ejecutan, y también se conocen las respuestas que se esperan del sistema cuando los usuarios interactúen con el mismo para realizar una tarea específica. En las tablas de los casos de uso también se pueden identificar los datos que son necesarios de registrar y las entidades a las que corresponden; así como también, se pueden identificar como una entidad se relaciona con los datos de otra.

Con los casos de uso se ha realizado el análisis a nivel de comportamiento, tanto del usuario como el sistema, lo cual permitirá definir el diseño estructural del sistema en los niveles que sean necesarios, para lograr un buen desarrollo del mismo.

CAPITULO III: MARCO METODOLOGICO

En este capítulo se lleva a cabo el modelado de la solución, utilizando el diagrama Entidad-Relación para diseñar la base de datos del sistema y definiendo el diccionario de datos para documentar su estructura. Se define la estructura a nivel de tecnologías, sobre la cual será implementado el sistema. Y por último se presenta el modelado de las interfaces gráficas del sistema para asegurar que los usuarios puedan realizar sus funciones interactuando de forma fácil y eficiente.

3.1 DISEÑO DE BASE DE DATOS

3.1.1 MODELO CONCEPTUAL DE DATOS

Para llevar a cabo el diseño de la base de datos del sistema EELO, se ha creado el modelado de datos a nivel conceptual. En la figura 3.1 se presenta el modelo de datos basado en la identificación de las entidades y sus relaciones.

Los atributos de cada una de las entidades son detallados en el diccionario de datos, con el cual se abarca el nivel lógico del modelado. En el diccionario de datos, las entidades son definidas como tablas y los atributos de las entidades son presentados como columnas de las tablas. Se define también, el tipo de dato y tamaño de cada una de las columnas. Y además se definen las llaves primarias y foranes de las tablas, como equivalentes de las relaciones del modelo conceptual.

El modelado de la base de datos se trabajó con la herramienta MySQL Workbench, la cual permite generar la base de datos como tal en MySQL a partir del modelo creado. Así como también permite crear un Script, el cual equivale al modelo físico de la base de datos ya que representa la implementación de la base de datos en MySQL, el cual es el gestor de base de datos requerido para la implementación del sistema.

3.1.2 DICCIONARIO DE DATOS

3.1 Tabla: departamento

Descripción: Tabla que almacenara los diferentes departamentos del país.

Campo	Tipo	Tamaño	Descripción	PK
id_departamento	INT	2	Clave primaria de Departamento.	*
nombre_departamento	VARCHAR	25	Nombre de Departamento.	

Relaciones: Ninguna.

Campos Clave Primaria: id_departamento.

Tabla 3.1 Tabla departamento de la base de datos.

3.2 Tabla: municipio

Descripción: Tabla que almacenara los diferentes municipios de cada departamento.

Campo	Tipo	Tamaño	Descripción	PK
id_municipio	INT	3	Clave primaria de Municipio.	*
id_departamento	INT	2	Clave de departamento a donde pertenece el municipio.	
nombre_municipio	VARCHAR	30	Nombre del Municipio.	

Relaciones: departamento: id_departamento (FK).

Campos Clave Primaria: id_municipio.

Tabla 3.3 Tabla municipio de la base de datos.

3.3 Tabla: cargo

Descripción: Tabla que almacenara los diferentes cargos desempeñados dentro del programa Oportunidades FGK.

Campo	Tipo	Tamaño	Descripción	PK
id_cargo	INT	2	Clave primaria del Cargo.	*
nombre_cargo	VARCHAR	200	Nombre del Cargo.	
descripcion	VARCHAR	500	Breve descripción del rol que representa el cargo.	

Relaciones: Ninguna

Campos Clave Primaria: id_cargo

Tabla 3.2 Tabla cargo de la base de datos.

3.4 **Tabla:** empleado

Descripción: Tabla que almacenara la información principal de cada Empleado.

Campo	Tipo	Tamaño	Descripción	PK
carnet_empleado	VARCHAR	15	Clave principal del Empleado.	*
id_cargo	INT	2	Clave del cargo que desempeñara.	
dui	VARCHAR	11	Documento Único de Identidad del Docente.	
nit	VARCHAR	17	Número de Identificación Tributaria del Docente.	
nombres	VARCHAR	45	Nombres del Docente.	
apellidos	VARCHAR	45	Apellidos del Docente.	
sexo	VARCHAR	1	Genero del Docente.	
fecha_nacimiento	DATE		Fecha de nacimiento del Docente.	
direccion	VARCHAR	150	Dirección de residencia del Docente.	
id_municipio	INT	2	Clave del municipio de residencia del Docente.	
telefono	VARCHAR	30	Teléfono del Docente.	
email	VARCHAR	75	Correo electrónico asignado al Empleado.	
estado	TINYINT	4	Booleano que indica si un empleado está activo o inactivo.	
observaciones	VARCHAR	250	Descripción u observaciones.	

Relaciones: municipio: id_municipio (FK) , cargo: id_cargp (FK).

Campos Clave Primaria: carnet_empleado.

Tabla 3.5 Tabla empleado de la base de datos.

3.5 **Tabla:** estado

Descripción: Tabla en la que se listan los diferentes estados que se pueden asignar a un alumno del programa.

Campo	Tipo	Tamaño	Descripción	PK
id_estado	INT	2	Clave primaria de Departamento.	*
estado	VARCHAR	45	Nombre de Departamento.	

Relaciones: Ninguna.

Campos Clave Primaria: id_estado.

Tabla 3.4 Tabla estado de la base de datos.

3.6 **Tabla:** centro_educativo

Descripción: Tabla que almacenara los diferentes centros educativos del territorio nacional.

Campo	Tipo	Tamaño	Descripción	PK
codigo_centro_educativo	INT	5	Clave primaria del centro educativo.	*
nombre_centro_educativo	VARCHAR	75	Nombre del centro educativo.	
direccion	VARCHAR	150	Dirección del centro educativo.	
telefono	VARCHAR	30	Listado de teléfonos del centro educativo.	
id_municipio	INT	2	Clave del municipio al que pertenece el centro educativo.	
descripcion	VARCHAR	250	Descripción u observaciones del centro educativo.	
sector	VARCHAR	10	Indica si el centro educativo pertenece al sector público o privado.	
zona	VARCHAR	10	Indica si el centro educativo está ubicado en zona rural o urbana.	
categoria	VARCAR	45	Especifica el tipo del centro educativo.	

Relaciones: municipio: id_municipio (fk).

Campos Clave Primaria: codigo_centro_educativo, id_municipio

Tabla 3.7 Tabla centro educativo de la base de datos.

3.7 **Tabla:** class

Descripción: Tabla que almacenara las diferentes Class (Promociones) que serán parte del programa oportunidades.

Campo	Tipo	Tamaño	Descripción	PK
id_class	VARCHAR	10	Clave primaria de Class.	*
anho_ingreso	YEAR	4	Año que iniciara la participación de la class.	
anho_egreso	YEAR	4	Año en el que culminará la participación de la class.	
descripcion	VARCHAR	255	Descripción u observaciones.	

Relaciones: Ninguna

Campos Clave Primaria: id_class

Tabla 3.6 Tabla class de la base de datos.

3.8 **Tabla:** alumno

Descripción: Tabla que almacenara la información principal de cada Alumno.

Campo	Tipo	Tamaño	Descripción	PK
carnet_alumno	VARCHAR	15	Clave primaria de Alumno.	*
id_class	VARCHAR	10	Clave de Class	
nombres	VARCHAR	45	Nombres del Alumno.	
apellidos	VARCHAR	45	Apellidos del Alumno.	
sexo	VARCHAR	1	Genero del Alumno.	
fecha_nacimiento	DATE		Fecha de nacimiento del Alumno.	
direccion	VARCHAR	150	Dirección de residencia del Alumno.	
foto	VARCHAR	100	Dirección URL de la foto del alumno	
id_municipio	INT	2	Clave del municipio al que pertenece la dirección de residencia del alumno.	
telefono	VARCHAR	10	Listado de números de contacto del Alumno.	
email	VARCHAR	75	Correo electrónico asignado al Alumno.	
codigo_centro_educativo	INT	5	Clave del centro educativo en el que el alumno está llevando sus estudios actuales.	
turno_educativo	VARCHAR	1	Turno escolar al que el alumno asiste al centro educativo.	
estado	INT	2	Estado en el que se encuentra el alumno (Activo o Inactivo)	
observacion	VARCHAR	250	Descripción y observaciones del Alumno.	

Relaciones: **Class:** id_class (fk), **municipio:** id_municipio (fk), **centro_educativo:** codigo_centro_educativo (fk).

Campos Clave Primaria: carnet_alumno

Tabla 3.8 Tabla alumno de la base de datos.

3.9 **Tabla:** ocupacion

Descripción: Tabla que almacenara las diferentes ocupaciones.

Campo	Tipo	Tamaño	Descripción	PK
id_ocupacion	INT	3	Clave primaria de ocupaciones.	*
nombre_ocupacion	VARCHAR	45	Nombre de la ocupación.	
descripcion	VARCHAR	250	Descripción de la ocupación.	

Relaciones: Ninguna

Campos Clave Primaria: id_ocupacion

Tabla 3.9 Tabla ocupacion de la base de datos.

3.10 **Tabla:** tipo_responsable

Descripción: Tabla que almacenara los diferentes roles de responsabilidad.

Campo	Tipo	Tamaño	Descripción	PK
id_tipo_responsable	INT	3	Clave primaria de tipo de responsable.	*
tipo_responsable	VARCHAR	45	Nombre de tipo responsable	

Relaciones: Ninguna

Campos Clave Primaria: id_tipo_responsable

Tabla 3.12 Tabla tipo de responsable de la base de datos.

3.11 **Tabla:** responsable

Descripción: Tabla que almacenara los responsables de cada alumno.

Campo	Tipo	Tamaño	Descripción	PK
Id_responsable	INT	11	Identificador único del responsable del alumno	*
dui	VARCHAR	10	Documento Único de Identidad (DUI)	
carnet_alumno	VARCHAR	15	Clave del alumno del cual será responsable.	
id_tipo_responsable	INT	3	Clave del tipo de responsable o rol.	
id_ocupacion	INT	3	Clave de la ocupación que ejerce.	
nombres	VARCHAR	45	Nombres del Responsable	
apellidos	VARCHAR	45	Apellidos del Responsable	
telefono	VARCHAR	30	Teléfono del Responsable	
direccion	VARCHAR	150	Dirección del Responsable	

Relaciones: alumno: carnet_alumno (fk), tipo_responsable: id_tipo_responsable (fk),
ocupación: id_ocupacion (fk).

Campos Clave Primaria: id_responsable

Tabla 3.11 Tabla responsable de la base de datos.

3.12 **Tabla:** seccion

Descripción: Tabla que almacenara las diferentes secciones que participarán por año dentro del programa oportunidades.

Campo	Tipo	Tamaño	Descripción	PK
id_seccion	VARCHAR	14	Clave primaria de Sección.	*
id_class	VARCHAR	10	Clave primaria de Class.	
anho	YEAR	4	Año de participación de la sección.	
descripcion	VARCHAR	255	Descripción u observaciones.	

Relaciones: class: id_class (fk).

Campos Clave Primaria: id_seccion.

Tabla 3.10 Tabla sección de la base de datos.

3.13 **Tabla:** alumno_seccion

Descripción: Tabla que almacenara la asignación de alumnos por sección.

Campo	Tipo	Tamaño	Descripción	PK
carnet_alumno	VARCHAR	15	Clave Primaria de Alumno	*
id_seccion	VARCHAR	14	Clave Primaria de Sección	*

Relaciones: alumno: carnet_alumno (fk), seccion: id_seccion (fk).

Campos Clave Primaria: carnet_alumno, id_seccion.

Tabla 3.15 Tabla alumno sección de la base de datos.

3.14 **Tabla:** bitacora_alumno_centro_educativo

Descripción: Tabla que almacenara la información de los cambios de centros educativos que el alumno realice.

Campo	Tipo	Tamaño	Descripción	PK
id_bitacora	INT	2	Clave primaria de la Bitácora de los centros educativos.	*
carnet_alumno	VARCHAR	15	Clave del alumno al que pertenece la bitácora.	
codigo_centro_educativo	INT	5	Clave del centro educativo al cual perteneció el alumno.	
Anho	YEAR	4	Año en que se realizó el cambio.	
grado_cursado	VARCHAR	25	Grado que curso el alumno en el centro educativo	
Observaciones	VARCHAR	250	Observaciones	

Relaciones: alumno: carnet_alumno (fk), centro_educativo: codigo_centro_educativo (fk).

Campos Clave Primaria: id_bitacora.

Tabla 3.14 Tabla bitácora alumno centro educativo de la base de datos.

3.15 **Tabla:** periodo

Descripción: Tabla que almacenara la descripción de cada periodo disponible en el año escolar.

Campo	Tipo	Tamaño	Descripción	PK
id_periodo	INT	2	Clave principal del periodo.	*
nombre_periodo	VARCHAR	45	Nombre del Periodo.	

Relaciones: Ninguna

Campos Clave Primaria: id_periodo

Tabla 3.13 Tabla periodo de la base de datos.

3.16 **Tabla:** materia

Descripción: Tabla que almacenara la información de cada materia.

Campo	Tipo	Tamaño	Descripción	PK
id_materia	INT	2	Clave principal del Materia.	*
nombre_materia	VARCHAR	45	Nombre de la Materia.	
codigo	VARCHAR	10	Código para identificar si la materia es del programa, del centro escolar o de ambas modalidades.	

Relaciones: Ninguna

Campos Clave Primaria: id_materia

Tabla 3.16 Tabla materia de la base de datos.

3.17 **Tabla:** grupo_materia

Descripción: Tabla que almacenara la información de cada grupo por materia del programa oportunidades.

Campo	Tipo	Tamaño	Descripción	PK
id_grupo_materia	VARCHAR	18	Clave principal del grupo de la materia.	*
id_seccion	VARCHAR	14	Clave de Sección.	
id_materia	INT	2	Clave de la materia.	
carnet_employado	VARCHAR	15	Clave del Empleado (Docente).	
Anho	YEAR	4	Año en el que fue habilitado el grupo por materia.	
Descripcion	VARCHAR	255	Descripción u observaciones	

Relaciones: empleado: carnet_employado (fk), materia: id_materia (fk), seccion: id_seccion (fk).

Campos Clave Primaria: id_grupo_materia.

Tabla 3.17 Tabla grupo materia de la base de datos.

3.18 **Tabla:** log_cambio_grupo_docente

Descripción: Tabla que almacenara los diferentes cambios de docente que se den en un grupo.

Campo	Tipo	Tamaño	Descripción	PK
carnet_employado	VARCHAR	15	Clave principal de Docente	*
id_grupo_materia	VARCHAR	18	Clave principal de Grupo de Materia	*
id_periodo	INT	2	Clave principal de Periodo	*
Fecha	DATE		Fecha en que se realizara el cambio.	
Observaciones	VARCHAR	250	Descripción del motivo del cambio.	

Relaciones: empleado: carnet_employado (fk), grupo_materia: id_grupo_materia (fk), periodo: id_periodo (fk).

Campos Clave Primaria: carnet_employado, id_grupo_materia, id_periodo

Tabla 3.18 Tabla cambio grupo docente de la base de datos.

3.19 **Tabla:** log_cambio_seccion_alumno

Descripción: Tabla que almacenara el historial de los cambios de seccion realizados por el alumno.

Campo	Tipo	Tamaño	Descripción
carnet_alumno	VARCHAR	15	Clave principal del alumno.
id_seccion	VARCHAR	14	Clave principal identificadora de la sección..
id_periodo	INT	11	Periodo en el que se realiza el registro de cambio de sección..
flotante	TINYINT	4	Especifica si el alumno es flotante o no.
observacion	VARCHAR	250	Observaciones a registrar respecto al cambio de sección.

Relaciones: alumno:carnet_alumno (Fk), seccion:id_seccion (Fk), Periodo:id_periodo (Fk).

Campos Clave: carnet_alumno, id_seccion, id_periodo.

Tabla 3.21 Tabla cambio sección alumno de la base de datos.

3.20 **Tabla:** evaluaciones

Descripción: Tabla que almacenara la descripción de cada evaluación realizada.

Campo	Tipo	Tamaño	Descripción	PK
id_evaluacion	INT	11	Clave principal de Evaluación	*
id_periodo	INT	2	Clave principal de Periodo	
id_materia	INT	2	Clave principal de Materia.	
id_class	INT	10	Clave primaria de Class	
nombre_evaluacion	VARCHAR	45	Nombre de la Evaluación	
Porcentaje	DECIMAL		Porcentaje equivalente de la Evaluación.	
año	YEAR	4	Año en que la evaluación es registrada para una materia determinada.	

Relaciones: periodo: id_periodo (fk), materia: id_materia (fk), class: id_class (fk).

Campos Clave Primaria: id_evaluacion.

Tabla 3.19 Tabla evaluaciones de la base de datos.

3.21 **Tabla:** notas_fgk

Descripción: Tabla que almacenara las notas obtenidas por el alumno en cada materia.

Campo	Tipo	Tamaño	Descripción	PK
carnet_alumno	VARCHAR	15	Clave principal de Alumno	*
id_evaluacion	INT	11	Clave principal de Evaluación	*
nota	DECIMAL	(4,2)	Nota Obtenida.	

Relaciones: alumno: carnet_alumno (fk), evaluacion: id_evaluacion (fk).

Campos Clave Primaria: carnet_alumno, id_evaluacion_grupo

Tabla 3.20 Tabla notas FGK de la base de datos.

3.22 **Tabla:** grado_mined

Descripción: Tabla que almacenara los grados cursados por un participan dentro del sistema educativo nacional.

Campo	Tipo	Tamaño	Descripción	PK
id_grado	INT	1	Clave principal del grado del sistema educativo nacional.	*
nombre_grado	VARCHAR	15	Nombre del Grado.	

Relaciones: Ninguna

Campos Clave Primaria: id_grado

Tabla 3.22 Tabla grado MINED de la base de datos.

3.23 **Tabla:** notas_mined

Descripción: Tabla que almacenara las notas obtenidas por cada materia base en el sistema educativo nacional.

Campo	Tipo	Tamaño	Descripción	PK
carnet_alumno	VARCHAR	15	Clave principal de Alumno.	*
id_materia	INT	2	Clave principal de Materia.	*
id_periodo	INT	2	Clave principal de Periodo.	*
id_grado	INT	1	Clave principal de Grado Mined.	*
Nota	DECIMAL		Nota Obtenida.	

Relaciones: alumno: carnet_alumno (fk), materia: id_materia (fk), periodo: id_periodo (fk), grado_mined: id_grado (fk).

Campos Clave Primaria: carnet_alumno, id_materia, id_periodo, id_grado.

Tabla 3.23 Tabla notas MINED de la base de datos.

3.24 **Tabla:** criterio

Descripción: Tabla que almacenara los diferentes criterios aptitudinales a evaluar.

Campo	Tipo	Tamaño	Descripción	PK
id_criterio	INT	2	Clave primaria del Criterio Actitudinal.	*
nombre_criterio	VARCHAR	200	Nombre del Criterio actitudinal de evaluación.	

Relaciones: Ninguna.

Campos Clave Primaria: id_criterio.

Tabla 3.24 Tabla criterio actitudinal de la base de datos.

3.25 **Tabla:** actitud

Descripción: Tabla que almacenara los diferentes Aptitudes evaluativas.

Campo	Tipo	Tamaño	Descripción	PK
id_actitud	INT	2	Clave primaria de la actitud.	*
id_criterio	INT	2	Clave primaria del Criterio Actitudinal al que corresponde.	
nombre_aptitud	VARCHAR	45	Nombre de la Actitud.	
Evaluar	TINYINT	4	Determina si actualmente se utiliza en las evaluaciones.	

Relaciones: criterio: id_criterio (fk).

Campos Clave Primaria: id_actitud.

Tabla 3.25 Tabla actitud de la base de datos.

3.26 **Tabla:** nota_actitudinal

Descripción: Tabla que almacenara las notas obtenidas por cada aspecto actitudinal durante un periodo de estudio.

Campo	Tipo	Tamaño	Descripción	PK
carnet_alumno	VARCHAR	15	Clave principal de Alumno.	*
Id_grupo_materia	VARCHAR	18	Clave principal del grupo de materia.	*
id_periodo	INT	2	Clave principal de Periodo.	*
id_actitud	INT	1	Clave principal de Actitud.	*
Nota	DECIMAL		Nota Obtenida.	

Relaciones: alumno: carnet_alumno (fk), grupo_materia: id_grupo_materia (fk), periodo: id_periodo (fk), actitud: id_actitud (fk).

Campos Clave: carnet_alumno, id_grupo_materia, id_periodo, id_actitud.

Tabla 3.26 Tabla nota actitudinal de la base de datos.

3.27 **Tabla:** users

Descripción: Tabla que almacena las credenciales de usuario de los empleados y alumnos.

Campo	Tipo	Tamaño	Descripción
id	BIGINT	20	Identificador único del registro
name	VARCHAR	255	Nombre del usuario
carnet	VARCHAR	255	Carnet identificador del usuario
email	VARCHAR	255	Correo electrónico
email_verified_at	TIMESTAMP		
is_admin	TINYINT	1	Boolean que determina si el usuario es administrador.
is_docente	TINYINT	1	Boolean que determina si el usuario es docente.
password	VARCHAR	255	Contraseña cifrada del usuario.
remember_token	VARCHAR	100	
created_at	TIMESTAMP		
updated_at	TIMESTAMP		
estado	TINYINT	4	

Campos Clave: id.

Tabla 3.27 Tabla usuarios de la base de datos.

3.2 DEFINICIÓN DE TECNOLOGIAS

3.2.1 HOSTING COMPARTIDO

Servicio de alojamiento en el que se almacenan múltiples sitios web (de diferentes clientes) en un solo servidor físico. Por lo tanto, todas las cuentas de los clientes comparten los recursos del servidor (Procesador, RAM, Ancho de banda, IP).

Se realizó un benchmark comparativo de proveedores de hosting, incluyendo el contratado por el director del programa. El documento de la comparativa realizada se encuentra en el ANEXO VI. A continuación, se presenta las características del plan de hosting seleccionado y que definirá la infraestructura del sistema EELO.

Características del servicio de hosting	
Proveedor	SiteGround
PLAN	Startup
Costo	\$11.95
Prestaciones de Cuenta	
Dominio	Gratis primer año - \$15.95/año
Soporte técnico	24/7 vía chat
Cuentas Email	Ilimitadas
Certificado SSL	✓
Copias de seguridad	Diaria
Cuentas FTP	Ilimitadas
Tráfico mensual	No medido
Número de visitas mensuales (óptimo)	10,000
Administrador de archivos - Cpanel	✓
Benchmark	9.7
Especificaciones Técnicas	
Almacenamiento SSD	✓
Espacio de alojamiento SSD	10 GB
Configuración del servidor personalizada	Contenedores Linux CentOS. Con tecnologías de servidor red apache y NGINX
Bases de datos MySQL	Ilimitadas
Versión php predeterminada	7.1
PHP personalizar múltiples versiones	✓
Acceso SSH y SFTP	✓
Recursos del Servidor	
Procesos del servidor simultáneas: 10	
Conexiones del servidor simultáneas de un IP: 10	
CPU segundos/programas y ejecuciones de scripts: 1000/hora, 10000/día, 300000/mes	
Promedio de tiempo de ejecución de procesos por día: 2 segundos	
Uso del CPU del servidor compartido: No más de 20% por un periodo de 10 segundos	
Memoria de servidor por cada proceso: 768 MB	
Inodes: 150,000	
Intervalos de Cron Jobs por el minimum: 30 minutos.	

Tabla 3.28 Plan Startup de SiteGround.

La razón por la que el hosting compartido delimita las tecnologías es porque estas han llegado a ser estandarizadas por el uso que se le da a dicho tipo de hosting; ya que, generalmente se utiliza para la implementación de sitios web como blogs personales, paginas informativas de empresas, sitios educativos, portafolios, foros, wikis y tiendas en línea que son construidos por medios de sistemas de gestión de contenido CMS como WordPress, Joomla, Drupal, Moodle, Magento y muchos más, los cuales permiten la creación de todos esos tipos de sitios web por medio de interfaces de alto nivel de abstracción.

La mayor parte de los CMS mencionados están creados por lenguaje de programación PHP y tienen soporte principalmente para el gestor de base de datos MySQL.

3.2.2 SERVIDOR WEB HTTP

A continuación, se describe la configuración de servidor que se adquiere con el proveedor de hosting seleccionado.

- **CONTENEDORES LINUX CXL**

El proveedor de hosting SiteGround utiliza la tecnología LXC para definir y limitar el uso de recursos del sistema por cada cliente. Esto permite tener un servidor físico con un sistema Linux anfitrión el cual contiene múltiples contenedores Linux que se van creando de acuerdo a las nuevas cuentas de clientes.

Los contenedores pueden ser definidos como máquinas creadas mediante virtualización ligera que son fácilmente replicables y que comparten kernel con el sistema anfitrión.

Utilizando las tecnologías cgroups y namespaces para administrar los procesos en el kernel anfitrión, se logra que los contenedores creados tengan una visión propia del sistema operativo, de los procesos del mismo, de la estructura de ficheros y de las interfaces de red, sin interferir en la maquina anfitriona o demás contenedores.

- **SERVIDOR APACHE COMO SERVIDOR WEB**

El servidor web es el encargado de procesar y servir todos los archivos que forman parte del sistema web, a todos los usuarios que realizan peticiones al interactuar con el mismo. El servidor Apache tiene una amplia aceptación en la red, y llegó a convertirse en el servidor web http más utilizado, siendo empleado en el 70% de los sitios web del mundo. Algunas de sus características:

- ❖ Servidor Web http de código abierto.

- ❖ Multiplataforma.
- ❖ Altamente configurable.
- ❖ La arquitectura del servidor Apache es muy modular. El servidor consta de una sección core y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web.
- ❖ Amplia extensión de documentación y soporte.

- **SERVIDOR PROXY NGINX**

La configuración implementada por el proveedor SiteGround también incluye un servidor NGINX con la función de proxy inverso. Esta implementación sirve de intermediario entre los clientes y el servidor web, al gestionar archivos estáticos en su memoria caché y realizando balanceo de carga para optimizar el trabajo de procesamiento del servidor web Apache. Además, define una red interna con el servidor web a la que los clientes que realizan las peticiones no tienen acceso, lo que añade una capa más de seguridad.

3.2.3 LENGUAJE DE PROGRAMACIÓN Y FRAMEWORK DE DESARROLLO

- **LENGUAJE PHP**

PHP (acrónimo recursivo de PHP: (Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en páginas HTML. Está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. (Autores contribuyentes de documentacion PHP, n.d.)

- ❖ Lenguaje totalmente libre y abierto.
- ❖ Los entornos de desarrollo son de rápida y fácil configuración.

- ❖ Fácil acceso e integración con las bases de datos.
- ❖ Posee una comunidad muy grande.
- ❖ Es el lenguaje con mayor usabilidad en el mundo.
- ❖ Es un lenguaje multiplataforma.
- ❖ Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados extensiones).

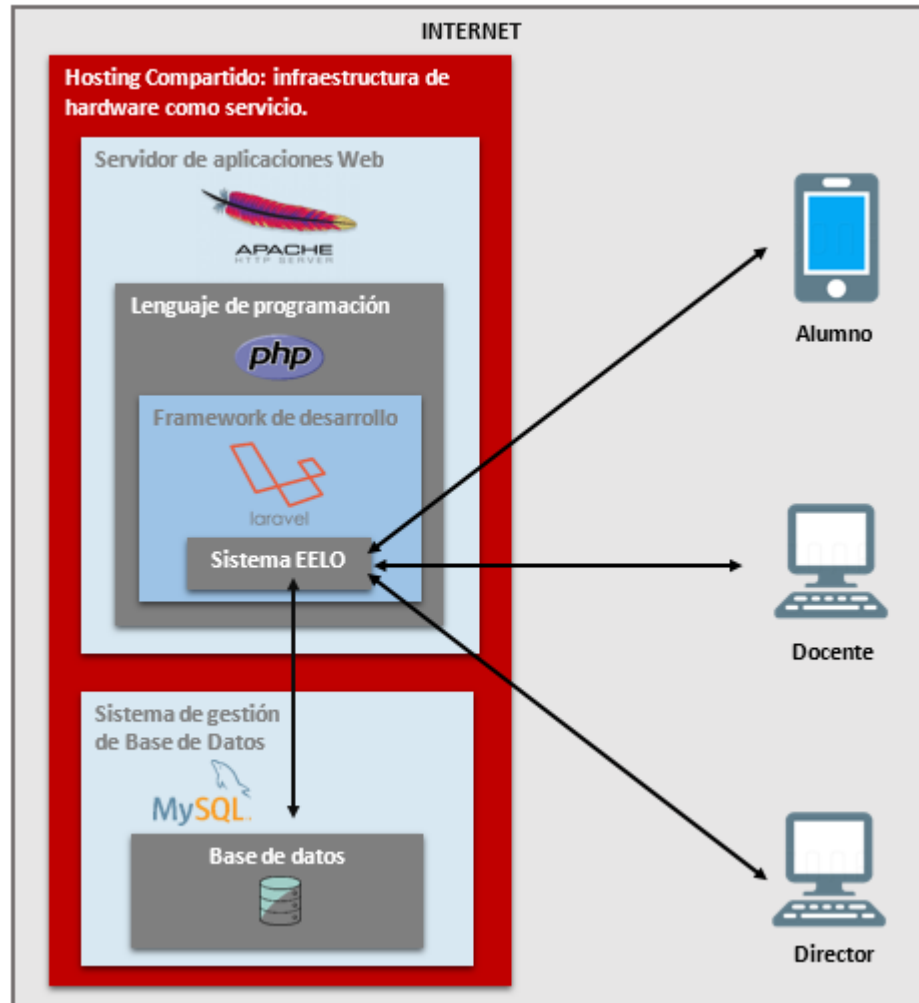


Figura 3.2 Estructura de Tecnologías del Sistema EELO.

- **FRAMEWORK LARAVEL**

Laravel es un framework para desarrollar aplicaciones web en PHP. Laravel tiene como objetivo ser un framework que permita el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y ofrecer multitud de funcionalidades. Algunas de las herramientas del framework son las siguientes:

- ❖ **Consola Artisan.** Artisan es una consola de comandos propia del framework y que provee una serie de comandos que asisten y agilizan la construcción de aplicaciones web.
- ❖ **Motor de plantillas Blade.** Blade permite crear vistas por medio de un sistema de plantillas en las que se pueden incrustar secciones HTML y variables y porciones de código PHP utilizando directivas.
- ❖ **ORM Eloquent.** Eloquent permite realizar un mapeo de los datos que se encuentran en la base de datos almacenados en un lenguaje de script SQL a objetos de PHP y viceversa. Eloquent brinda una capa de abstracción de la base de datos, ya que los datos son manejados por medio de los Modelos creados, sin utilizar lenguaje SQL.
- ❖ **Sistema de Ruteo.** En Laravel se define una tabla de rutas para acceder a los controladores, los cuales contienen la lógica de la aplicación. Los controladores procesan la información para generar y enviar las vistas Blade al usuario y también consultan y actualizan la base de datos por medio de los modelos Eloquent.



Figura 3.3 Patrón MVC con Laravel.

Todas las herramientas incluidas en Laravel están orientadas al desarrollo de código implementando el patrón MVC y ofrecer una sintaxis elegante y expresiva para escribir dicho código.

3.3 DISEÑO DE VISTAS DEL SISTEMA

Para crear el diseño de las interfaces gráficas de usuario del sistema se creó un prototipo de vistas utilizando la herramienta Photoshop. Dichos diseños fueron presentados en una reunión con el personal administrativo y los docentes del programa para validar si las interfaces gráficas de usuario se encuentran alineadas con la especificación de los requerimientos.

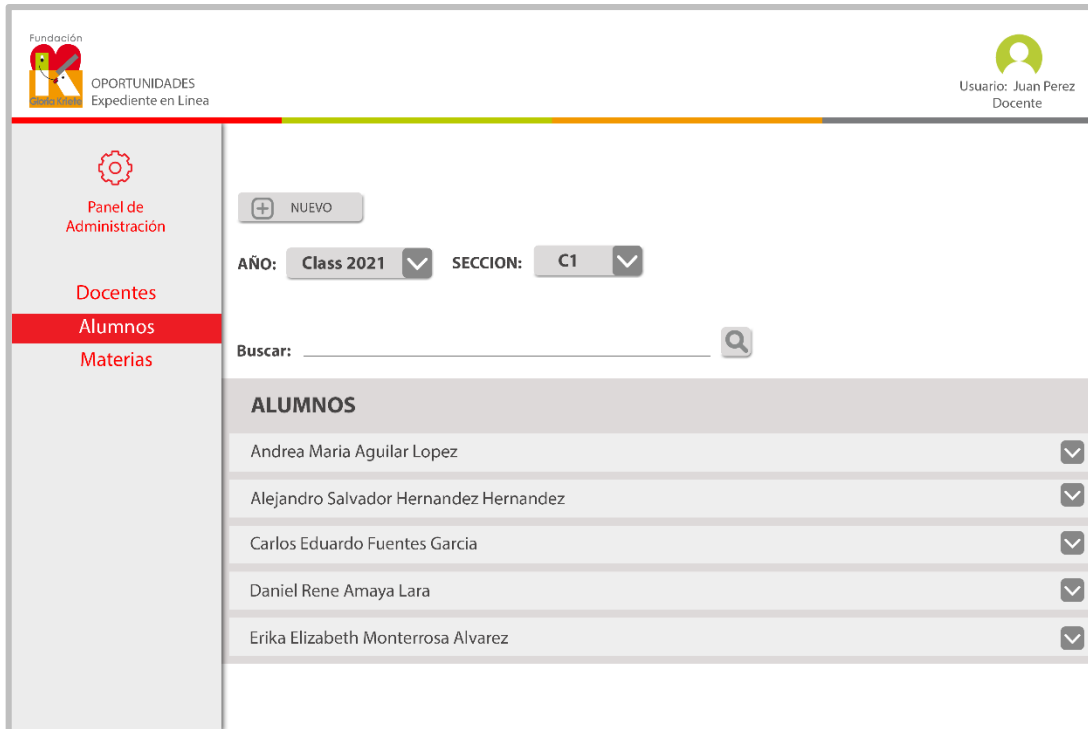


Figura 3.5 Prototipo de vista CRUD alumnos.

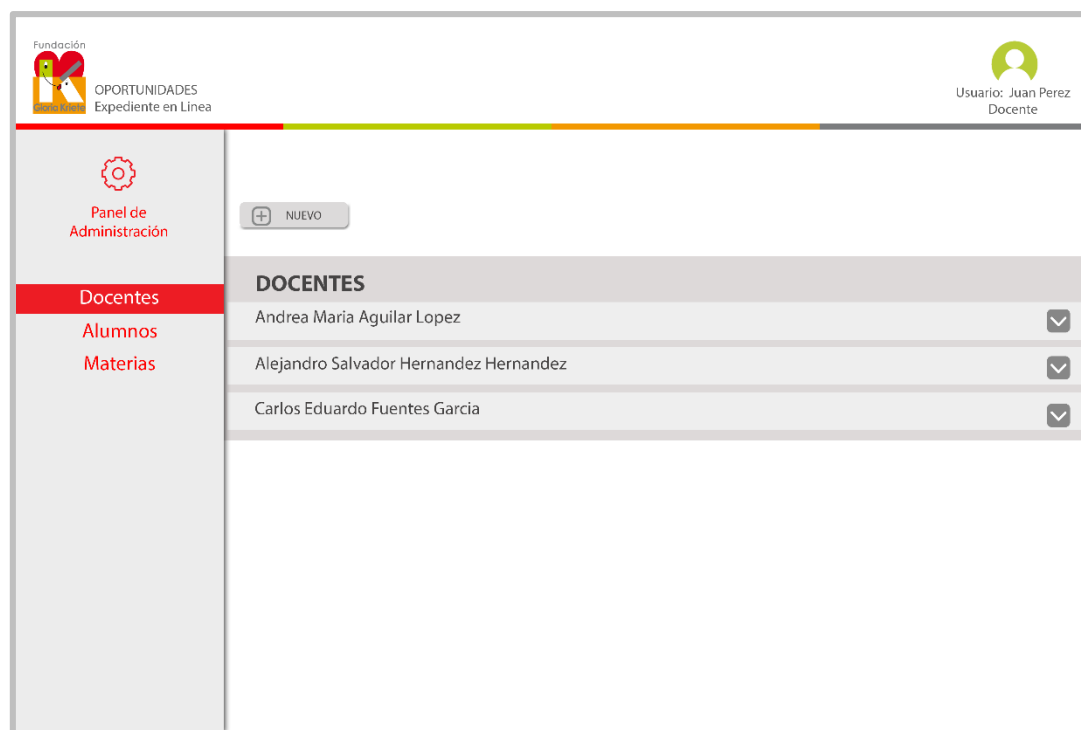


Figura 3.4 Prototipo de vista CRUD de docentes.

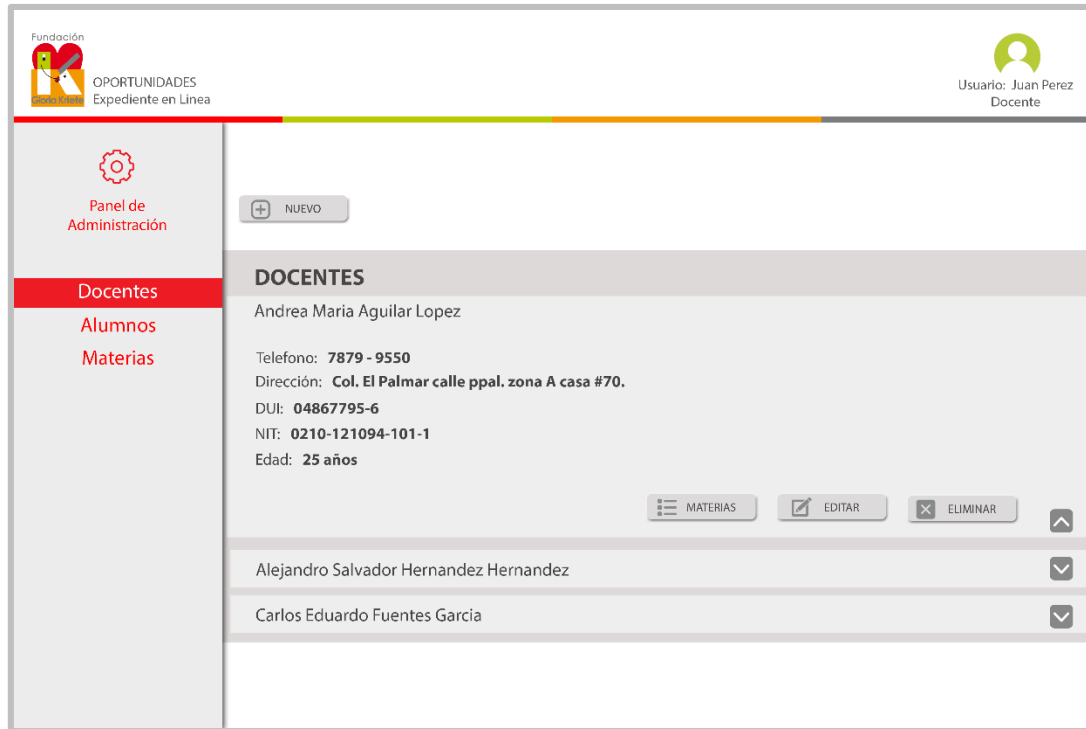


Figura 3.7 Prototipo vista detalle de docentes.

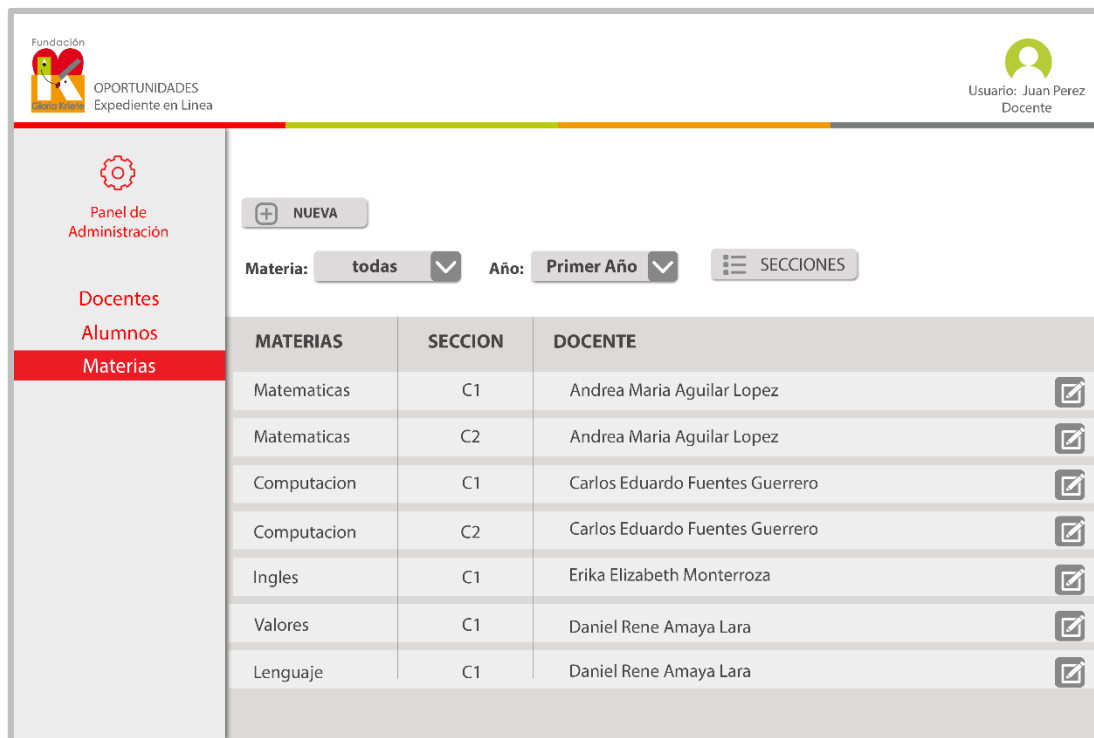


Figura 3.6 Prototipo vista de asignación de docentes a secciones.

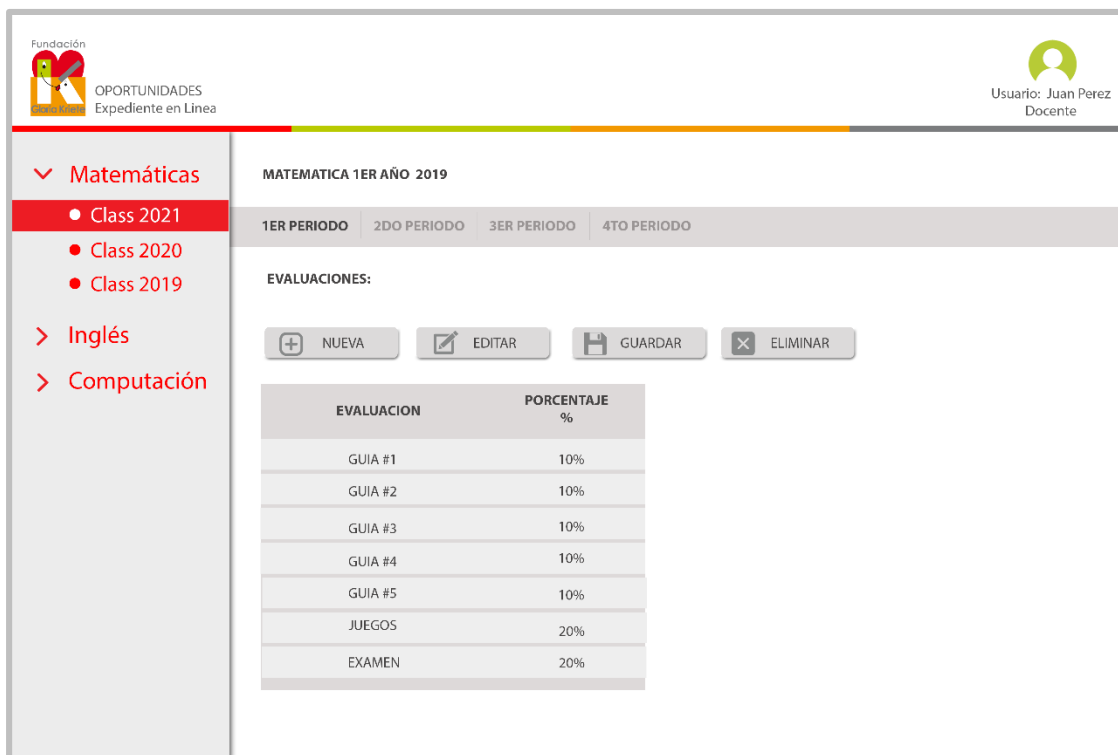


Figura 3.8 Prototipo de vista CRUD de evaluaciones.

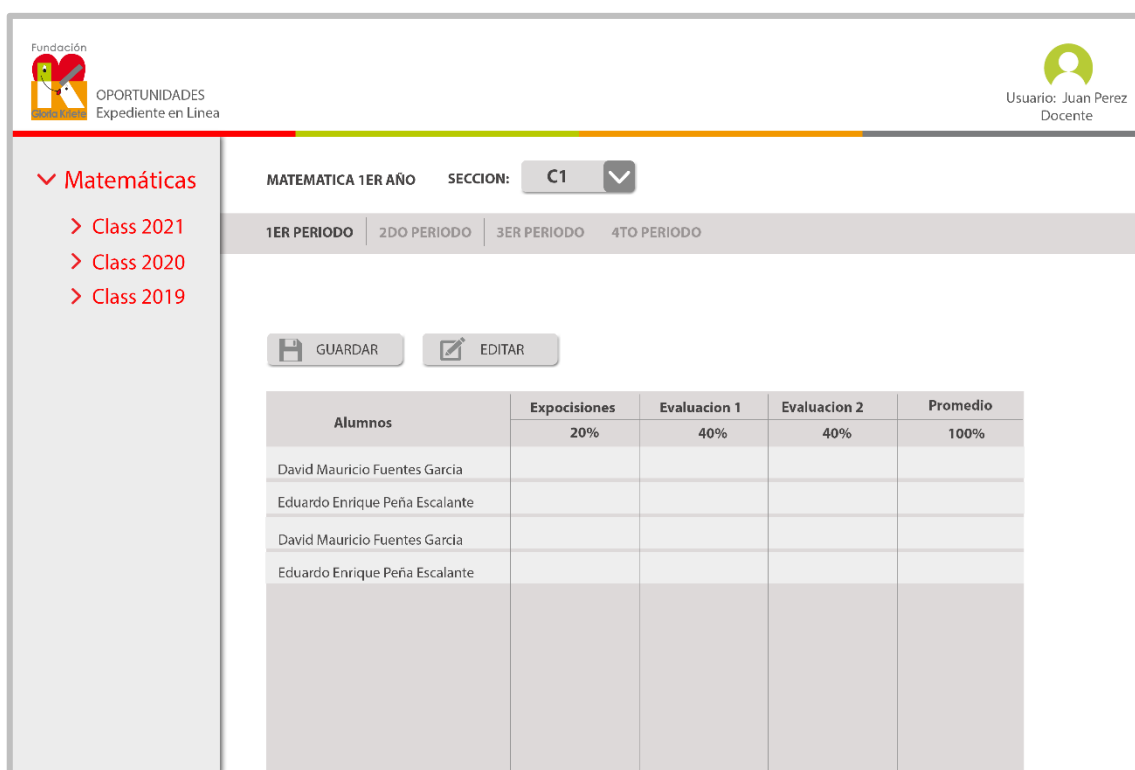


Figura 3.9 Prototipo de vista de registro de notas FGK.

- ❖ En primer lugar, las vistas propuestas cuentan con un menú de navegación en la parte lateral izquierda para todos los usuarios y que, dependiendo el rol de este, se cargan las opciones o ítems de navegación por cada una de las tareas definidas para el usuario.
- ❖ Cada una de las vistas está diseñada para realizar la gestión de los datos de las entidades en la base de datos. Por cada entidad, por ejemplo, el docente o alumno, la vista carga un listado de los datos registrados y botones para ejecutar las funciones CRUD con dichos registros.
- ❖ Las vistas propuestas fueron aceptadas por el personal del programa con la única observación de utilizar el manual de marca del Programa Oportunidades, ANEXO VII.

- **FRAMEWORK MATERIALIZE CSS**

Para la creación de las vistas del sistema EELO se incluirá la utilización de Materialize CSS, el cual es un framework front-end basado en los conceptos de material design de Google. El framework ofrece múltiples componentes CSS y JavaScript que permiten crear interfaces intuitivas y responsivas para que puedan ser accedidas desde un computador de escritorio o cualquier dispositivo móvil.

CAPITULO IV: DESARROLLO E IMPLEMENTACION DEL SISTEMA EELO

En este capítulo se ha documentado el desarrollo del código fuente de cada uno de los elementos del sistema en base al diseño y estructura establecida en el capítulo previo. Se presentan la funcionalidad de cada una de las vistas del sistema, así como también se describen los archivos que hacen posible dicha funcionalidad, siguiendo el patrón modelo vista controlador, implementado con el framework Laravel. Se realiza la descripción de actividades para llevar a cabo la implementación del sistema, así como la documentación de los resultados y comportamiento del sistema en el ambiente de producción.

4.1 ESTRUCTURA DEL SISTEMA

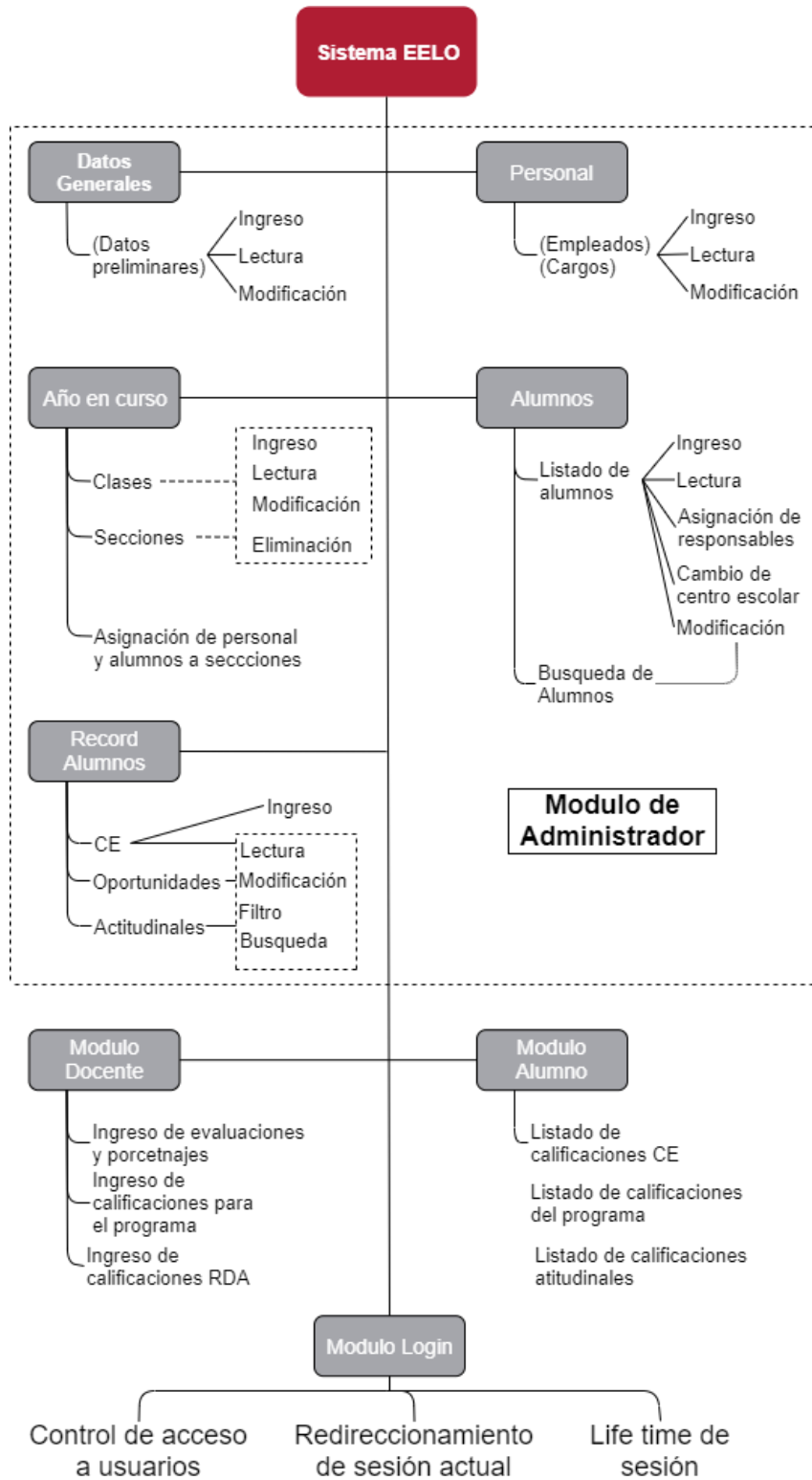


Figura 4.1 Esquema de los módulos y submódulos del sistema EELO

El sistema se compone de 3 módulos principales dedicados al rol que el usuario desempeña, permitiéndoles realizar las acciones y tareas que le corresponden (Administrador, Docente y Alumno), y un módulo dedicado a la seguridad (Modulo Login) el cual se encargara de redireccionar a los usuarios a sus páginas de inicio correspondientes según su rol.

Los módulos están compuestos por submódulos, los cuales agrupan las diferentes tareas que pueden realizar un usuario en específico, de acuerdo a su rol dentro del sistema. De esta forma se logra una mejor accesibilidad a la información y fácil navegación a nivel de interfaz de usuario. La mayor parte de las tareas corresponden a las operaciones fundamentales de persistencia de datos CRUD.

4.1.1 GENERALIDADES DEL SISTEMA

Todas las operaciones relacionadas con una misma entidad del modelo de la base de datos, han sido agrupadas en secciones o vistas, por medio de las cuales dichas operaciones son ejecutadas. Dichas vistas y su funcionalidad han sido desarrolladas bajo el patrón de arquitectura MVC (Modelo, Vista, Controlador) implementado con el framework Laravel con el propósito de mantener una estructura ordenada. El modelo es el archivo que representa la estructura de las entidades y sus atributos en la base de datos. El controlador es el archivo en el cual se crean los métodos que harán las peticiones a la base dando como resultado el registro, actualización y consulta de los datos, por medio de los modelos. Las vistas son los archivos que dan forma a la interfaz gráfica del sitio web y las operaciones dinámicas que se realizan inmediatamente del lado del cliente.

TABLA DE VERSIONES: HERRAMIENTAS DE SOFTWARE DE EL SISTEMA	
Gestor de paquetes Composer	V 1.8.4
Lenguaje de programación PHP	V 7.3.2
Laravel Framework	V 6.4.1
MySQL Community Server	V 8.0.13
Java	V 8
Visual Studio Code	V 1.49
Materialize CSS	V 1.0

En la tabla 4.1 se detallan las versiones del framework Laravel y demás herramientas de software utilizadas para establecer el entorno desarrollo y llevar a cabo la creación del código fuente del sistema EELO.

VISTAS DEL SISTEMA: VIEWS Y LAYOUTS

- Los archivos HTML en los cuales se encuentran definidas las estructuras de las diferentes vistas graficas de la aplicación, son llamadas **views** dentro del entorno Laravel.
- Todas las vistas se encuentran organizadas en subcarpetas en la ruta ***Eelofgkresourcesviews***.
- Dentro de los views de la aplicación se hace uso de Blade, el motor de creación de plantillas incluido en Laravel.
- **Directiva @yield:** Es una directiva de la sintaxis Blade la cual permite definir plantillas, llamadas **layouts**, e insertar contenido dinámico dentro de las mismas. De esta forma es posible utilizar un solo archivo como plantilla que contenga las secciones como la cabecera y menú de navegación, ya que son comunes a múltiples páginas, e ir definiendo de forma dinámica el contenido correspondiente a un view en específico. En la figura 4.2 se puede observar cómo se utiliza la directiva @yield dentro de las etiquetas **<main>** de la plantilla **admin.blade.php**.
- **Directiva @extends:** la directiva @extends de Blade permite aplicar una plantilla en otro archivo HTML. En la figura 4.7 se observa que en el view

departamento.blade.php se utiliza la plantilla *admin.blade.php* haciendo uso de la directiva **@extends** al inicio del archivo.

- **Directiva @section:** Es la directiva que permite insertar secciones de código, en los archivos que extienden una plantilla. Todo el contenido de etiquetas HTML que sea definido dentro de la directiva **@section('parametro')** es introducido en la plantilla, donde este definida la directiva **@yield('parametro')** con el mismo nombre como parámetro. En la figura 4.7 todo el código dentro de **@section('contenido')** es introducido en **@yield('contenido')** de plantilla *admin.blade.php*.

CONTROLADORES DEL SISTEMA

- **make:controller.** Utilizando la interfaz de línea de comando Artisan de Laravel se generan los controladores de la aplicación. El comando para la creación de los controladores es **php artisan make:controller Controlador --resource**, el cual recibe como parámetro el nombre del controlador a crear. Utilizando la opción **--resource** se generan dentro del controlador las funciones para el manejo de las peticiones Http, para realizar 7 operaciones básicas CRUD. En la imagen 4.9 se detallan los métodos creados en el controlador **DepartamentoController** por medio del comando, cada uno con el verbo Http correspondiente, la ruta y su nombre asignado.
- Los controladores son creados en la ruta **Eelofgk\app\Http\Controllers.**
- **Tabla de Rutas:** Todas las rutas que hacen referencia a los views y controladores de la aplicación Laravel se encuentran registradas en el archivo: **Eelofgk\routes\web.php.**
- **Route::resource (URI, controlador):** Este método es un controlador RESTful utilizado para generar la tabla de rutas requeridas para gestionar las operaciones básicas CRUD de una aplicación, utilizando un controlador para dar respuesta al manejo de dichas operaciones. El método toma dos argumentos, donde el primero es el URI de la solicitud entrante y el segundo es el nombre del controlador que se usa para

manejar la solicitud. De esta forma se define una serie de rutas URL para acceder a cada uno de los métodos creados en el controlador que es pasado como parámetro.

MODELOS DEL SISTEMA

- **make:model.** Con el comando *php artisan make:model NombreTabla*, se crean los modelos de la aplicación, los cuales se utilizan para modelar cada una de las entidades de la base de datos. Con el ORM Eloquent de Laravel, los modelos creados cuentan con una serie de métodos que facilitan la consulta, registro y actualización de los datos.
- Los modelos son creados en la ruta *Eelofgklapp*.

4.2 MÓDULO ADMINISTRADOR

El módulo de Administración contempla todo lo correspondiente con la parte inicial y preparación del sistema, tanto para la parte de ingreso de información básica y complementaria, como la asignación de los docentes y alumnos a las secciones que estarán asistiendo, al inicio de cada año escolar.

Además, el administrador puede consultar el avance académico de los alumnos por medio de los records, en los cuales se muestran con todo detalle las notas obtenidas por los alumnos a lo largo de permanencia en el programa Oportunidades.

VISTA DEL MÓDULO ADMINISTRADOR

La vista del home para el usuario administrador, está creada con el archivo *administrador.blade.php*, el cual es utilizado como layout para definir dos elementos principales, los cuales son comunes a todos los submódulos y secciones del módulo administrador.

- **Cabecera de admin.blade.php:** Dentro de la cabecera de la plantilla se definen las referencias a librerías de código que complementan y facilitan el desarrollo del código del sistema, figura 4.4:

- ❖ **Materialize CSS** utilizado como framework para la creación de elementos responsivos a nivel de interfaz de usuario, basado en Material Design.
- ❖ **Jquery**, librería JavaScript que facilita la manipulación de los elementos HTML y la definición de peticiones AJAX.
- ❖ **Google Icon Font**, librería de iconos de Material Design.
- ❖ **Maskedinput**, librería JavaScript utilizada para poner mascara de entrada de datos a campos de texto en los formularios de las vistas.

```

39 <body>
40
41 <!--BARRA SUPERIOR-->
42 > <nav id="nav_admin">...
67 </nav>
68
69 <div>
70 <!--PANEL LATERAL DEL MENU-->
71 > <ul id="slide-out" class="sidenav sidenav-fixed z-depth-3">...
179 </ul>
180
181 </div>
182
183 <!--CONTENIDO DINAMICO DE ACUERDO AL ELEMENTO SELECCIONADO EN EL MENU-->
184 <main>
185 <@yield('contenido')
186 <@include ('footer')
187 </main>
188
189
190 </body>

```

Figura 4.2 Uso de directiva @yield en plantilla admin.blade.php

- **Panel de Navegación de admin.blade.php:** El panel de navegación creado con el elemento **Collapsible** de Materialize Css, figura 4.3, permite agrupar los enlaces de todos los submodulos y sus secciones,

en un menú de navegación desplegable. Cada uno de los enlaces del menú, están registrados en la Tabla de Rutas.

```
91 <!--ELEMENTOS PRINCIPALES DEL MENU-->
92 <ul class="collapsible">
93
94     <!--DATOS GENERALES-->
95     <li><div class="collapsible-header waves-effect principal-item"><i
96         class="material-icons">settings</i>Datos Generales</div>
97 > <div class="collapsible-body">...
107 </div>
108 </li>
109
110     <!--PERSONAL-->
111     <li><div class="collapsible-header waves-effect principal-item"><i
112         class="material-icons">assignment_ind</i>Personal</div>
113 > <div class="collapsible-body">...
122 </div>
123 </li>
124
125     <!--AÑO EN CURSO-->
126     <li><div class="collapsible-header waves-effect principal-item"><i
127         class="material-icons">assignment</i>Año en Curso</div>
128 > <div class="collapsible-body">...
138 </div>
139 </li>
140
141     <!--ALUMNOS-->
142     <li><div class="collapsible-header waves-effect principal-item"><i
143         class="material-icons">school</i>Alumnos</div>
144 > <div class="collapsible-body">...
149 </div>
150 </li>
151
152     <!--RECORD DE ALUMNOS-->
153     <li><div class="collapsible-header waves-effect principal-item"><i
154         class="material-icons">assessment</i>Record de Alumnos</div>
155 > <div class="collapsible-body">...
168 </li>
169
170 </ul>
```

Figura 4.3 Panel de navegación del Layout admin.blade.php

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">

  <!-- JQuery -->
  <script src="https://code.jquery.com/jquery-3.3.1.min.js"
    integrity="sha256-FgpCb/KJQlLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8=" crossorigin="anonymous">
  </script>
  <!-- Compiled and minified CSS -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/materialize@1.0.0/css/materialize.min.css">
  <!-- Compiled and minified JavaScript -->
  <script src="https://cdn.jsdelivr.net/npm/materialize@1.0.0/js/materialize.min.js"></script>
  <!-- Import Google Icon Font -->
  <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
  <link rel="shortcut icon" type="image/x-icon" href="img/opportunidades_ico.ico" />
  <!--Alertify-->
  <link rel="stylesheet" href="{{ asset('js/alertifyJS/css/alertify.min.css') }}" />
  <link rel="stylesheet" href="{{ asset('js/alertifyJS/css/themes/semantic.min.css') }}" />
  <script src="{{ asset('js/alertifyJS/alertify.min.js') }}"></script>
  <!--Maskedinput-->
  <script src="{{ asset('includes/masked_input/dist/jquery.maskedinput.js') }}" type="text/javascript"></script>
  <!--DataTable-->
  <script type="text/javascript" charset="utf8" src="//cdn.datatables.net/1.10.11/js/jquery.dataTables.js"></script>
  <link rel="stylesheet" type="text/css" href="//cdn.datatables.net/1.10.11/css/jquery.dataTables.css">

  <!--CSS y JS propios del sistema-->
  <link rel="stylesheet" href="{{ asset('css/style.css') }}">
  <script src="{{ asset('js/init.js') }}"></script>
  <meta name="csrf-token" content="{{ csrf_token() }}">
  @yield('js')

  <title>{{ 'Home Expediente' }}</title>
</head>

```

Figura 4.4 Cabecera del Layout admin.blade.php

4.2.1 SUBMÓDULO: DATOS GENERALES

El submódulo de Datos Generales contempla las secciones que se dedican al registro de todos los datos complementarios y de configuración, los cuales son básicos y necesarios para el funcionamiento del sistema y registro de los demás datos. El registro de este tipo de datos se ha considerado, para que el sistema cumpla con el criterio de escalabilidad y pueda ser implementado en sedes de otras zonas del país.

En la imagen 4.5 se presentan todas las rutas definidas que hacen referencia a cada una de las secciones del submódulo Datos Generales, utilizando el método

resource. Y en la imagen 4.6 se muestra la forma en que dichas rutas son utilizadas en los enlaces del panel de navegación de la plantilla **admin.blade.php**.

```
//-----Admin routes-----//
Route::resource('/admin/departamentos', 'DepartamentoController');
Route::resource('/admin/municipios', 'MunicipioController');
Route::resource('/admin/school', 'CentroEducativoController');
Route::resource('/admin/clases', 'ClassController');
Route::resource('/admin/secciones', 'SeccionController');
Route::resource('/admin/materias', 'MateriaController');
Route::resource('/admin/criterios', 'CriterioController');
Route::resource('/admin/evaluacionesExtras', 'EvaluacionesExtrasController');
Route::resource('/admin/actitudes', 'ActitudController');
Route::get('/actitudes/lista', 'ActitudController@listaActitudesActivasJson');
```

Figura 4.5 Rutas del submódulo Datos Generales, en el archivo web web.php

```
<!--DATOS GENERALES-->
<li>
  <div class="collapsible-header waves-effect principal-item"><i
    class="material-icons">settings</i>Datos Generales</div>
  <div class="collapsible-body">
    <ul>
      <li><a href="/admin/departamentos" class="waves-effect secondary-item">Departamentos</a></li>
      <li><a href="/admin/municipios" class="waves-effect secondary-item">Municipios</a></li>
      <li><a href="/admin/school" class="waves-effect secondary-item">Centros Educativos</a></li>
      <li><a href="/admin/materias" class="waves-effect secondary-item">Materias</a></li>
      <li><a href="/admin/criterios" class="waves-effect secondary-item">Criterios Actitudinales</a></li>
      <li><a href="/admin/actitudes" class="waves-effect secondary-item">Actitudes</a></li>
      <li><a href="/admin/evaluacionesExtras" class="waves-effect secondary-item">Otras Evaluaciones</a></li>
    </ul>
  </div>
</li>
```

Figura 4.6 Enlaces del módulo Datos Generales en el menú de navegación

4.2.1.1 SECCIÓN: DEPARTAMENTO

En la sección de Departamentos se registran los departamentos del país a los cuales los usuarios y centros educativos pertenecen

VISTA DE LA SECCIÓN DEPARTAMENTO

- El view **Departamento.blade.php** hace uso de la directiva Blade **@extends('layout.admin')**, para hacer uso de la plantilla

admin.blade.php. Con la directiva **@section** se define el contenido dentro del archivo que corresponde a la sección de departamento. Todo el código dentro de la directiva **@section('contenido')** es introducido en **@yield('contenido')** de plantilla ***admin.blade.php***.

- La vista de la sección Departamento cuenta con un listado de los registros guardados hasta el momento. Para realizar un nuevo registro cuenta con un campo de texto para ingresar el nombre del departamento y un botón para guardar el registro. Para realizar una edición, cada registro del listado cuenta con un botón de edición, el cual activa una ventana modal para realizar la edición del nombre del departamento y guardar dicha edición.
- La vista de la sección Departamento es acompañada por un archivo JavaScript ***departamento.js*** con el cual se agregan las funcionalidades al view ***Departamento.blade.php*** y que permiten interactuar con la misma de forma dinámica.

```
3 @extends('layouts.admin')
4
5 @section('js')
6 <script src="{{ asset('js/departamento.js') }}"></script>
7 @endsection
8
9 @section('contenido')
10 <div class="container">
11
12     <!--encabezado-->
13 > <header>...
15 </header>
16
17 > <div class="cuerpo">...
67 </div>
68
69 </div>
70
71 <!--Modal para editar los registros-->
72 > <div id="ModalEdit" class="modal">...
94 </div>
95
96 @endsection
97
```

Figura 4.8 Uso de las directivas @extends y @section en View Departamento.

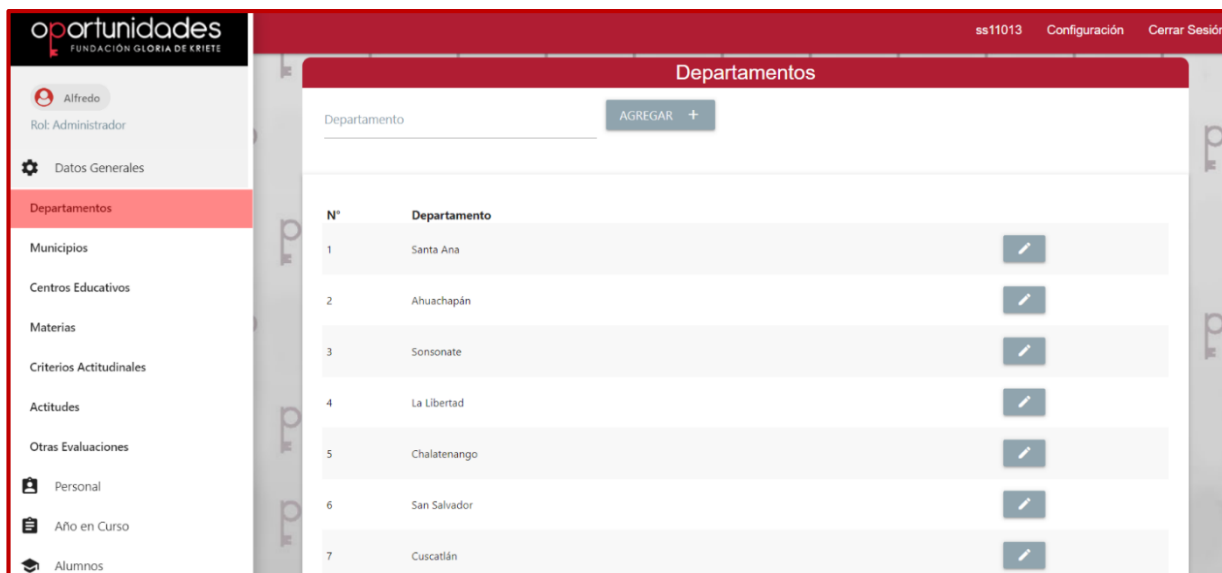


Figura 4.7 View Departamento.

CONTROLADOR DE LA SECCIÓN DEPARTAMENTO

- En la figura 4.9 se detallan los métodos creados por defecto en el controlador ***DepartamentoController***, cada uno con el verbo http correspondiente, la ruta y su nombre asignado.
- **departamentos.index:** función del controlador ***DepartamentoController*** que corresponde a una petición Http GET hecha a la ruta ***/admin/departamento*** y devuelve la vista ***departamento.blade.php*** junto con una colección de los departamentos registrados en la base de datos.
- **departamentos.store:** función de ***DepartamentoController***, ejecutada con una petición Http POST a la ruta ***/admin/departamento***. La función toma los datos obtenidos de la vista, para registrar un nuevo departamento en la base de datos. La petición del lado del cliente, es ejecutada por medio de Ajax, con la función ***agregarRegistro()*** del archivo ***departamento.js***.
- **departamentos.show:** La función show del controlador, se ejecuta cuando realiza una petición Http de tipo GET con la ruta ***/admin/departamento/{id_departamento}***, siendo el último elemento un parámetro que se envía desde el lado del cliente. La función recibe el parámetro y realiza una consulta a la base de datos utilizando el parámetro como identificador del departamento a buscar, la función devuelve el resultado de la consulta a la vista del departamento. Dicha función es invocada desde una petición Ajax en la vista ***departamento.blade.php***, activada con el botón de edición de registro en la vista.

- **departamentos.update:** La función update del controlador, se ejecuta cuando realiza una petición Http de tipo PUT con la ruta */admin/departamento/{id_departamento}*. Con esta función realiza una sentencia update en la base de datos, con los registros enviados por medio de Ajax en la vista de departamento. El método Ajax se ejecuta al confirmar la actualización de datos en el modal de edición.
- **departamentos.destroy:** En la función **destroy** se ejecuta una sentencia delete en la base de datos, para eliminar el registro de un departamento cuyo llave primaria es pasada como parámetro en la ruta */admin/departamento/{id_departamento}*. Para ejecutar la función, la petición Http que se hace a dicha ruta es de tipo DELETE. La petición se realiza con Ajax en la vista de departamento.

Metodo	URI	Name	Accion
GET HEAD	admin/departamentos/index	departamentos.index	App\Http\Controllers\DepartamentoController@index
POST	admin/departamentos	departamentos.store	App\Http\Controllers\DepartamentoController@store
GET HEAD	admin/departamentos/create	departamentos.create	App\Http\Controllers\DepartamentoController@create
PUT PATCH	admin/departamentos/{departamento}	departamentos.update	App\Http\Controllers\DepartamentoController@update
DELETE	admin/departamentos/{departamento}	departamentos.destroy	App\Http\Controllers\DepartamentoController@destroy
GET HEAD	admin/departamentos/{departamento}	departamentos.show	App\Http\Controllers\DepartamentoController@show
GET HEAD	admin/departamentos/{departamento}/edit	departamentos.edit	App\Http\Controllers\DepartamentoController@edit

Figura 4.9 Rutas y métodos del controlador DepartamentoController.

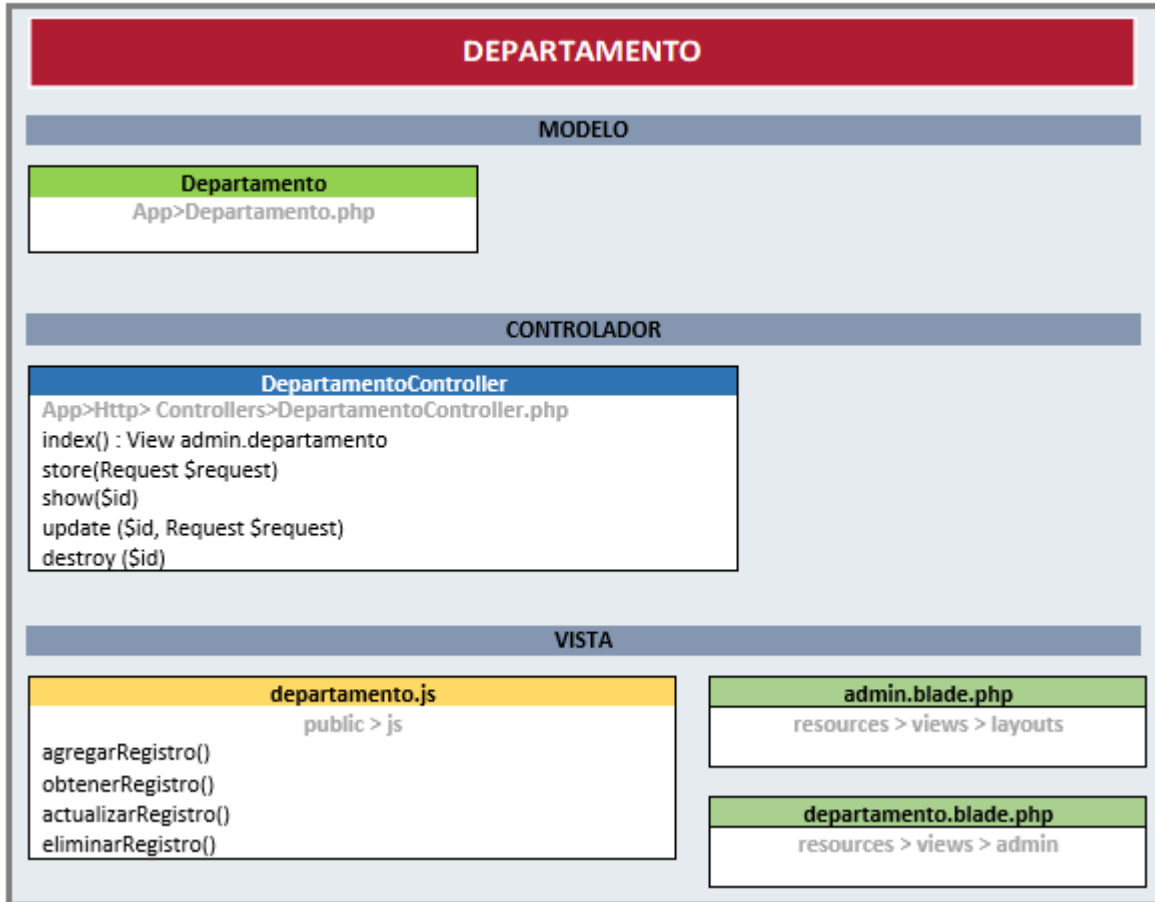


Figura 4.10 MVC de Sección Departamento

En la figura 4.10 se detalla la estructura de las clases y archivos utilizados para implementar el patrón Modelo Vista Controlador (MVC) y desarrollar la sección Departamento con sus operaciones CRUD. La implementación del patrón MVC es definida por las clases y herramientas proporcionadas por el framework Laravel.

La misma forma de aplicar el patrón Modelo Vista Controlador (MVC) es utilizada como base para la creación de los demás módulos, submódulos y secciones del sistema. A continuación, se detallan los demás submódulos y secciones del sistema, detallando las generalidades de cada una.

4.2.1.2 SECCIÓN MUNICIPIO

En la sección de Municipios se muestran los municipios por departamento del país, al igual que la sección Departamento se cuenta con una vista en la cual se desencadenan las operaciones CRUD de forma dinámica, de acuerdo a la interacción del administrador con los elementos de la vista.

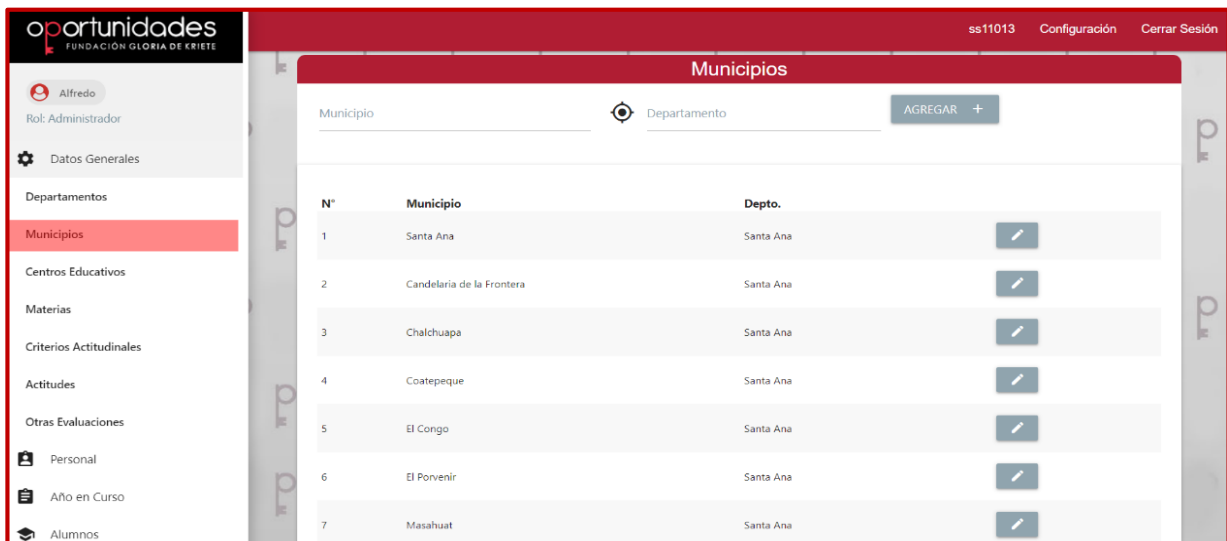


Figura 4.11 Vista de la sección Municipio.

Tabla 4.2 Funcionalidad de Sección Municipio.

FUNCIONALIDAD DE SECCIÓN MUNICIPIO	
Cargar vista Municipio	La vista se carga desde el menú de navegación por medio de una petición GET a la ruta <code>/admin/municipio/</code> la cual es gestionada por el método <code>index()</code> del controlador <code>MunicipioController</code> . El método devuelve la vista <code>Municipio.blade.php</code> y una colección de objetos del modelo <code>Municipio</code> .
Registrar nuevo Municipio	La vista cuenta con un formulario en la parte superior para realizar un nuevo registro. El botón guardar realiza una petición Ajax de tipo POST para enviar la información al método <code>store(Request \$request)</code> del controlador <code>MunicipioController</code> el cual ejecuta una sentencia INSERT en la base de datos, con el nuevo registro.
Editar Municipio	El botón de edición de cada registro, realiza una petición por medio de Ajax, la cual es manejada con el método <code>show(\$id)</code> del controlador para obtener los datos del registro a modificar y enviarlos como respuesta. Los datos son cargados en el formulario de un modal.
Guardar edición	Al guardar la edición de los datos en el modal, se realiza la petición PUT para ejecutar el método <code>update()</code> del controlador el cual recibe como parámetro el id y los datos a modificar de un municipio específicos para construir y ejecutar la sentencia UPDATE en la base de datos.

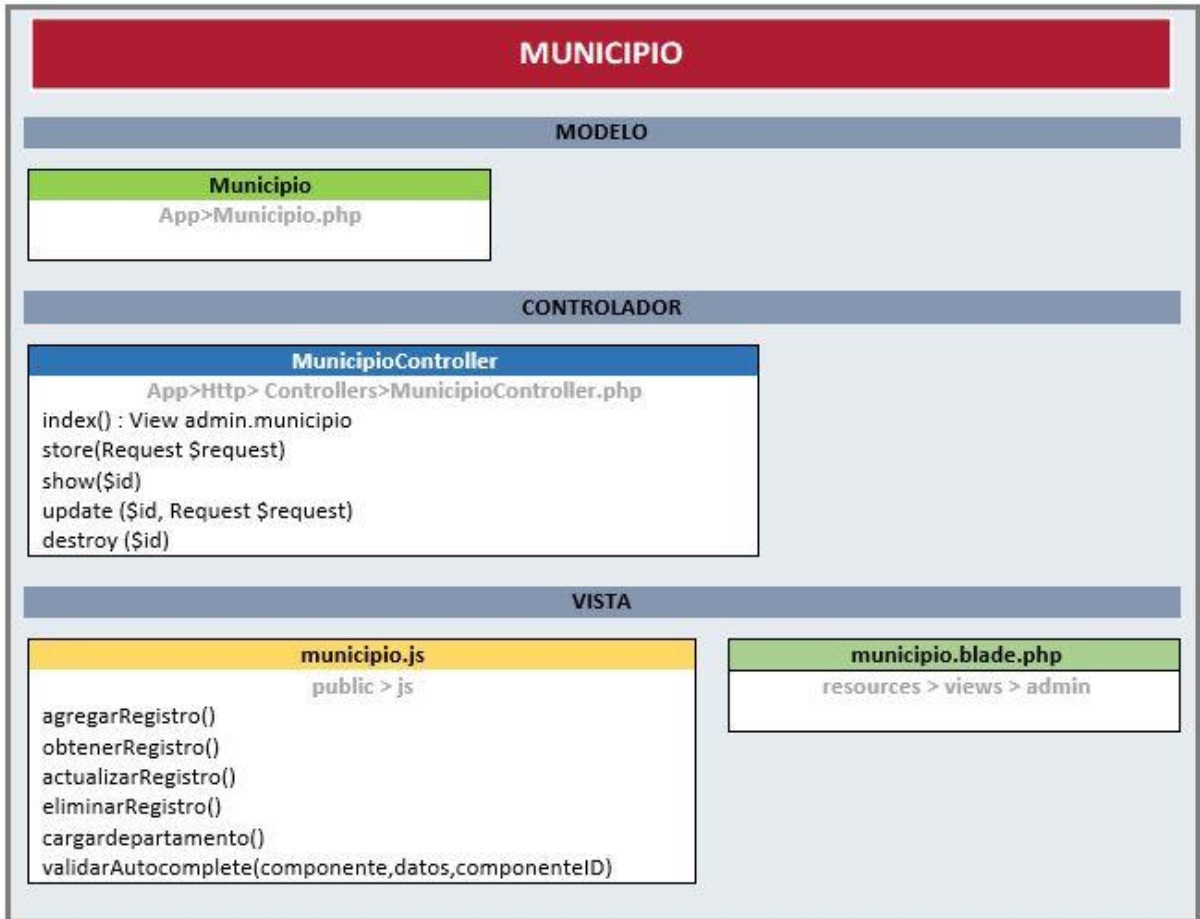


Figura 4.12 MVC de la sección Municipio.

4.2.1.3 SECCIÓN CENTRO EDUCATIVO

La sección de Centros Educativos se utiliza para registrar toda la información referente a los centros educativos públicos o privados donde los alumnos cursan noveno grado y bachillerato.

Código	Nombre	Teléfono	Dirección
10220	COMPLEJO EDUCATIVO CAPITAN GENERAL GERARDO BARRIOS	7471-6031	FINAL TERCERA CALLE ORIENTE PASAJE LOS ALMENDROS BARRIO EL CALVARIO
10221	COMPLEJO EDUCATIVO ANSELMA SANCHEZ DE MANCIA	7613-4398	AVENIDA 14 DE DICIEMBRE Y CALLE PRICIPAL, CLONIA SAN MANUEL, BARRIO SAN ANTONIO
10245	CENTRO ESCOLAR CANTON SAN SEBASTIAN	7987-6819	CALLE AL COCO, DESVIO A 70 METROS DESPUES DEL RIO
10249	CENTRO ESCOLAR COLONIA SANTA CRUZ	2411-4574	COLONIA CUSCACHAPA, CALLE PRINCIPAL
10268	CENTRO ESCOLAR FRANCISCO IGNACIO CORDERO	2414-7163	FINAL CALLE GENERAL RAMON FLORES ORIENTE
10269	CENTRO ESCOLAR GUADALUPE RETANA HERRERA	2444-0249	SEXTA AVENIDA NORTE Y SEXTA CALLE PONIENTE NUMERO VEINTIDOSBARRIO SANTA CRUZ
10270	CENTRO ESCOLAR SIMON BOLIVAR	2444-0659	CALLE RAMON FLORES PONIENTE Y SEXTA AVENIDA SUR

Figura 4.13 Vista de la Sección Centro Educativo.

Form fields for registration:

- Código: 00000
- Municipio
- Teléfono: 0000-0000
- Nombre
- Sector: Público
- Categoría: Centro Escolar
- Zona: Urbano
- Dirección
- Descripción - Campo Opcional

AGREGAR +

Figura 4.14 Desplegable para registro de Centro Educativo.

FUNCIONALIDAD DE SECCIÓN CENTRO EDUCATIVO	
Cargar vista Centro Educativo	La vista se carga desde el menú de navegación por medio de una petición GET a la ruta <code>/admin/school</code> la cual es gestionada por el método <code>index()</code> del controlador <code>CentroEducativoController</code> . El método devuelve la vista <code>school.blade.php</code> , figura 4.13, y una colección de objetos del modelo <code>CentroEducativo</code> y <code>Municipio</code> , utilizados para cargar el listado de centros educativos registrados en la vista.
Registrar nuevo Centro Educativo	La vista cuenta con un formulario desplegable en la parte superior, figura 4.14, para realizar un nuevo registro. El botón guardar realiza una petición Ajax de tipo POST para enviar la información al método <code>store(Request \$request)</code> del controlador <code>CentroEducativoController</code> el cual ejecuta una sentencia INSERT en la base de datos, con el método <code>save()</code> del modelo <code>CentroEducativo</code> .
Editar Centro Educativo	El botón de edición de cada registro, realiza una petición por medio de Ajax, la cual es manejada con el método <code>show(\$id)</code> del controlador para obtener los datos del registro a modificar y enviarlos como respuesta. Los datos son cargados en el formulario de un modal.
Guardar edición	Al guardar la edición de los datos en el modal, se realiza la petición PUT para ejecutar el método <code>update()</code> del controlador el cual recibe como parámetro el id y los datos a modificar de un Centro educativo específico para construir y ejecutar la sentencia UPDATE en la base de datos, con el método <code>update()</code> del modelo <code>CentroEducativo</code> .

Tabla 4.3 Funcionalidad de Sección Centro Educativo.

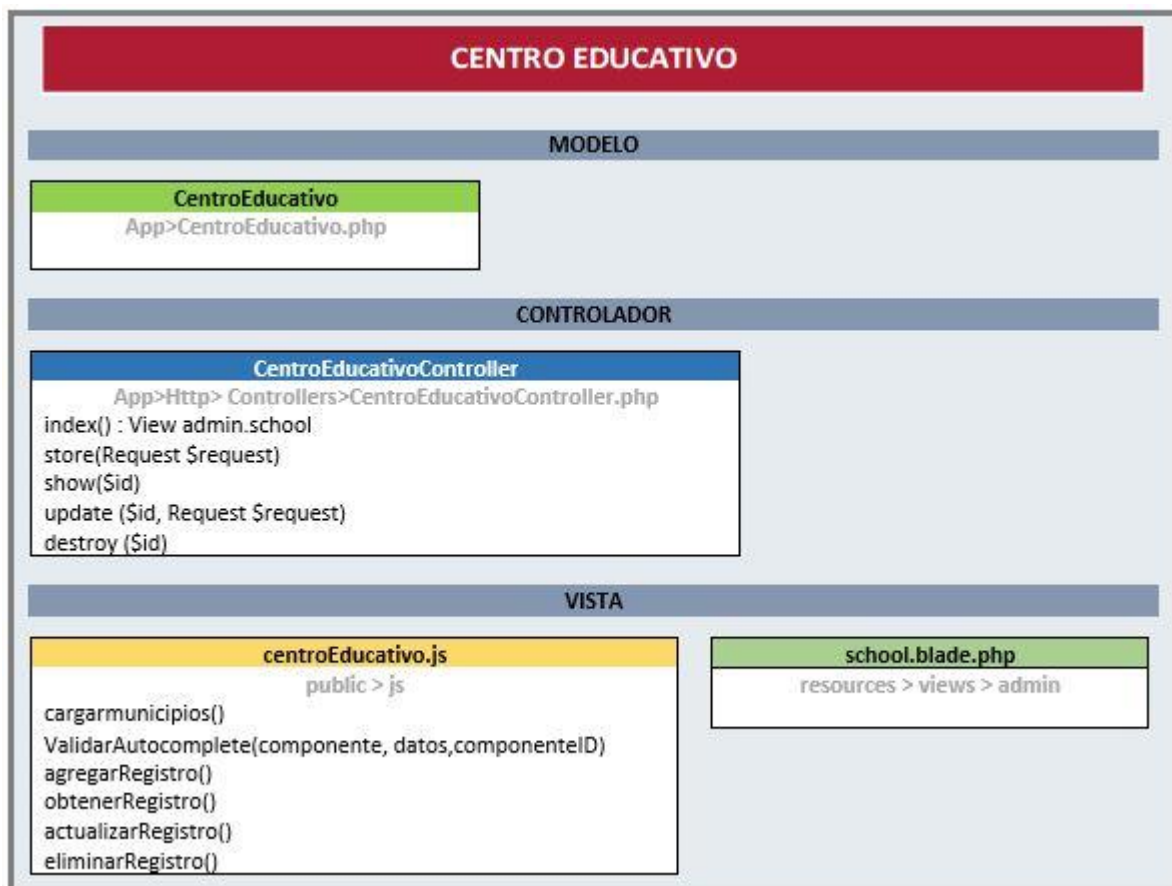


Figura 4.15 MVC de Centro Educativo.

4.2.1.4 SECCIÓN MATERIAS

La sección Materias se encarga de registrar las materias en las cuales los alumnos tendrán registros de notas, siendo estas las impartidas en el programa de educación del MINED como las que están recibiendo en sus centros educativos.

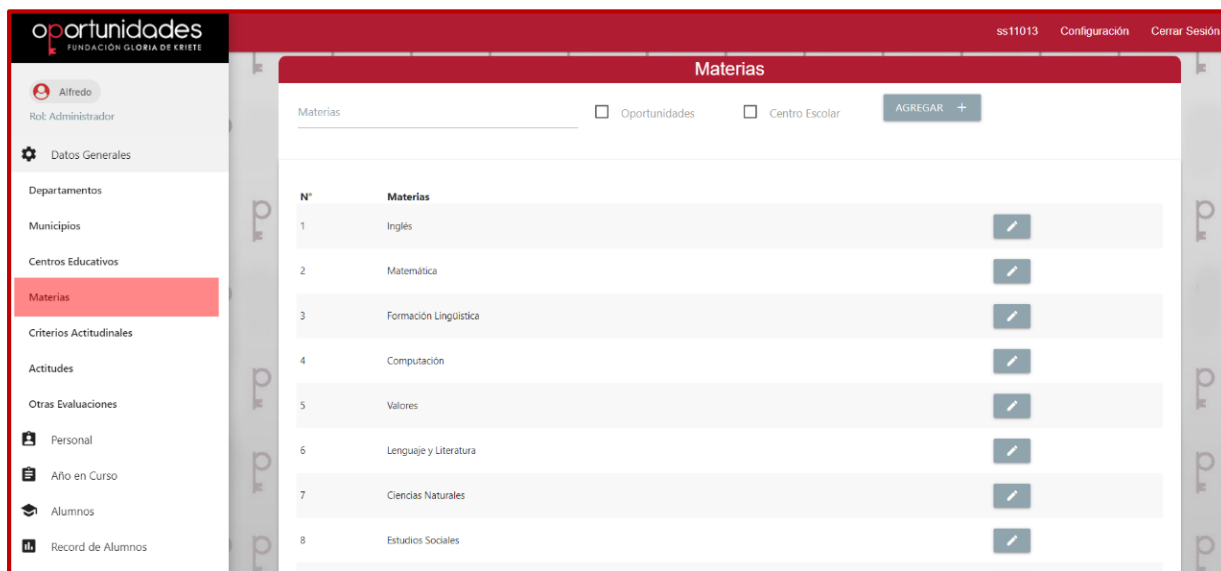


Figura 4.16 Vista de Sección Materia.

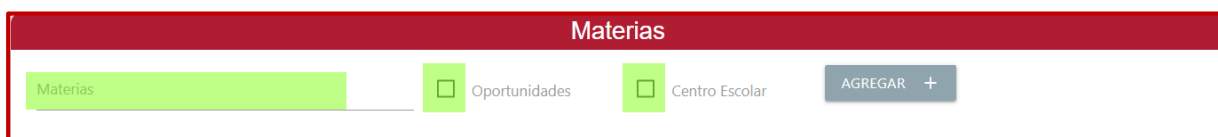


Figura 4.17 Formulario de registro de Materia.

FUNCIONALIDAD DE SECCIÓN MATERIA	
Cargar vista Materia	La vista se carga desde el menú de navegación por medio de una petición GET a la ruta <code>/admin/materias</code> la cual es gestionada por el método <code>index()</code> del controlador <code>MateriaController</code> . El método devuelve la vista <code>materia.blade.php</code> y una colección de objetos del modelo <code>Materia</code> , utilizada para cargar el listado de registros en la vista, figura 4.16.
Registrar nueva Materia	La vista cuenta con un formulario en la parte superior para realizar un nuevo registro, en el cual se debe especificar el nombre y a qué programa pertenece la materia, figura 4.17. El botón guardar realiza una petición Ajax de tipo POST para enviar la información al método <code>store(Request \$request)</code> del controlador <code>MateriaController</code> el cual ejecuta una sentencia INSERT en la base de datos, con el método <code>save()</code> del modelo <code>Materia</code> .
Editar Materia	El botón de edición de cada registro en el listado de materias, realiza una petición por medio de Ajax, la cual es manejada con el método <code>show(\$id)</code> del controlador para obtener los datos del registro a modificar y enviarlos como respuesta. Los datos son cargados en el formulario de un modal.
Guardar edición	Al guardar la edición de los datos en el modal, se realiza la petición PUT para ejecutar el método <code>update()</code> del controlador el cual recibe como parámetro el id y los datos a modificar de una materia específica para construir y ejecutar la sentencia UPDATE en la base de datos, con el método <code>update()</code> del modelo <code>Materia</code> .

Tabla 4.4 Funcionalidad de Sección Materia.

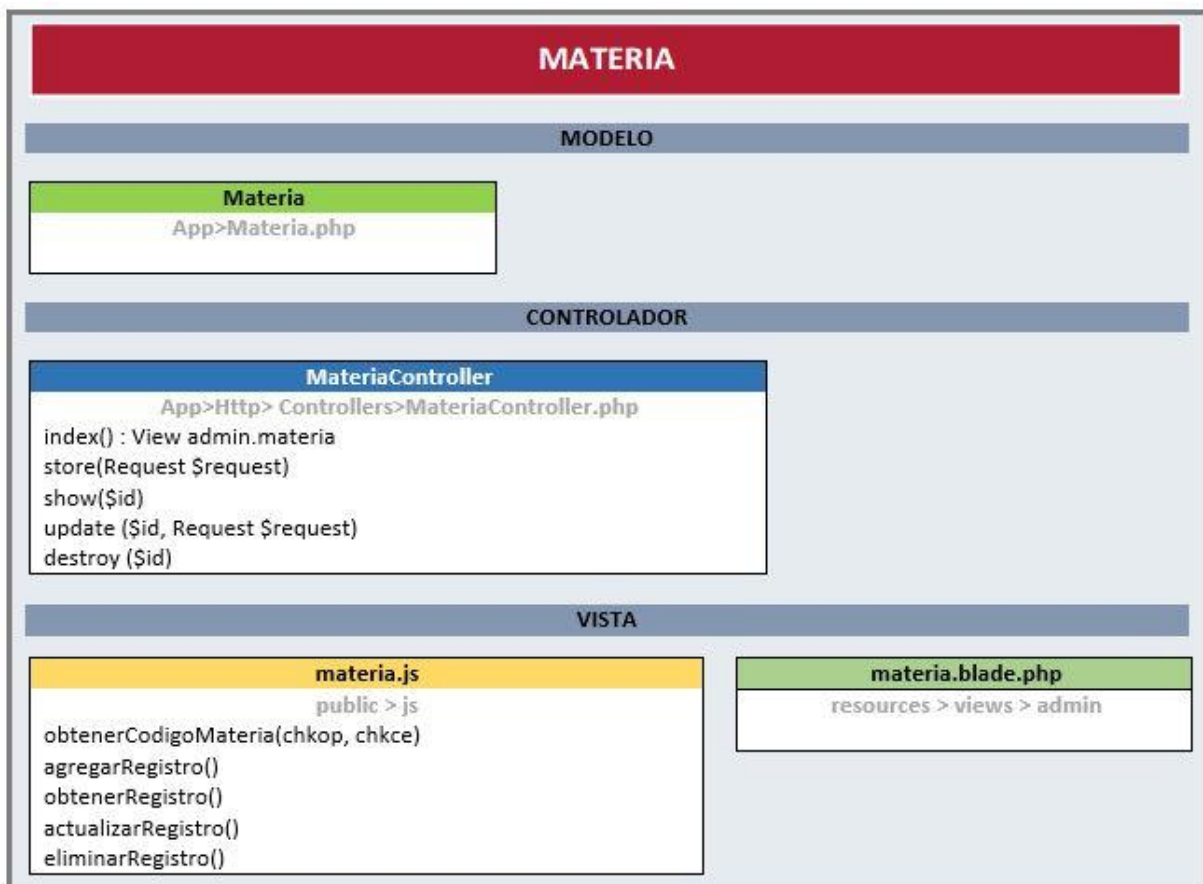
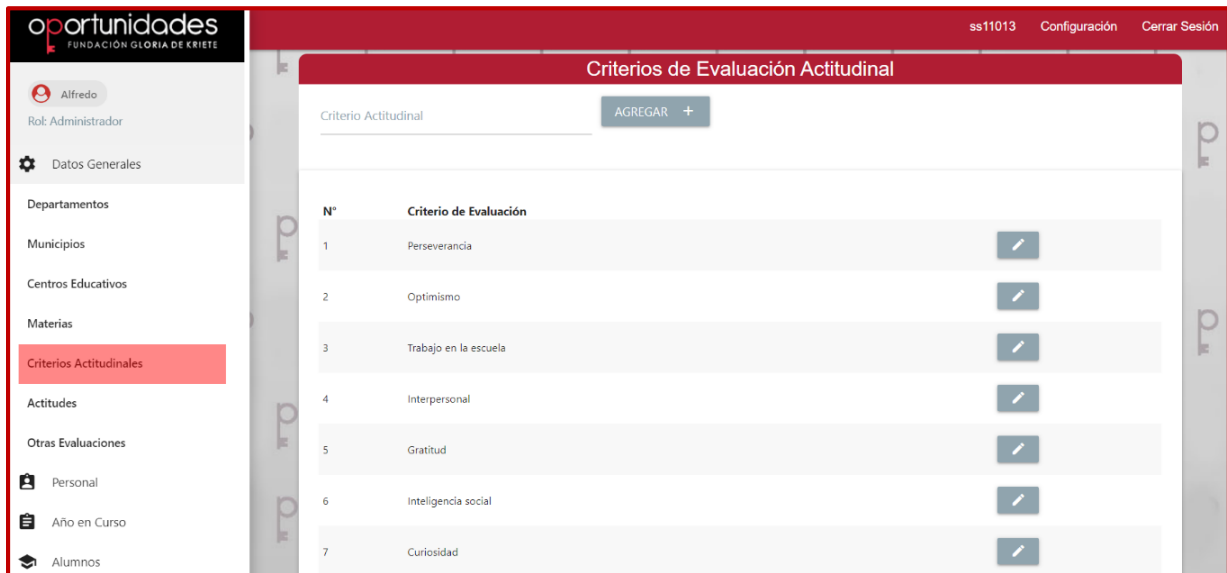


Figura 4.18 MVC de Sección Materia.

4.2.1.5 SECCIÓN CRITERIO DE EVALUACIÓN ACTITUDINAL Y ACTITUDES

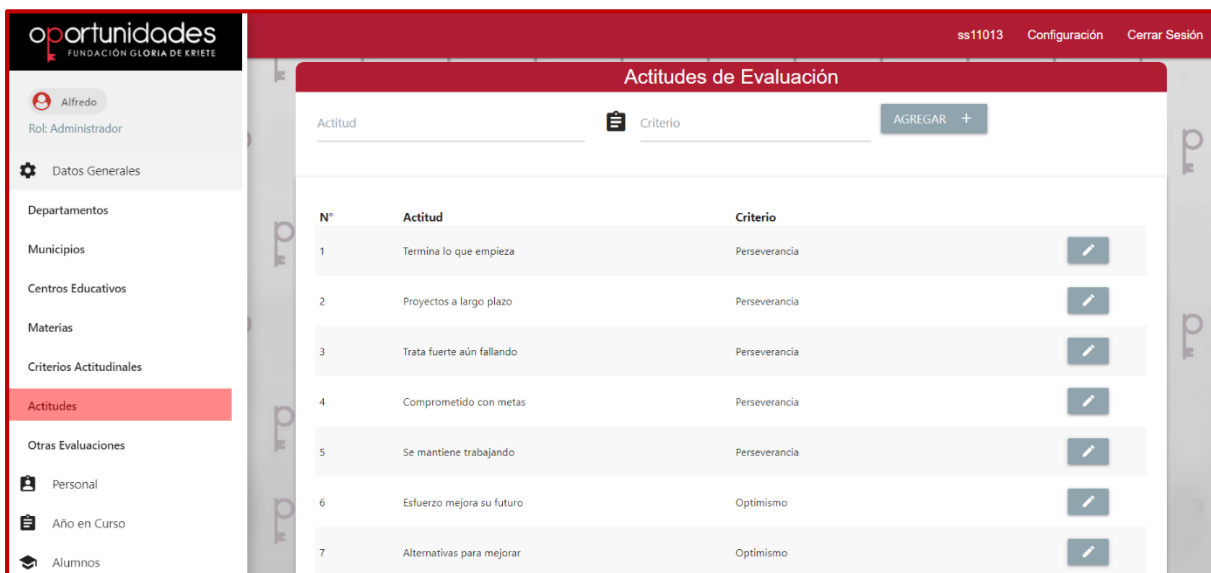
La sección criterios actitudinales registra y actualiza las categorías que son utilizadas para organizar de forma puntual, todas las actitudes para la evaluación del comportamiento y carácter de los alumnos. El registro y edición de las actitudes también cuenta con su vista definida, en la cual se especifica el criterio al que pertenece.



The screenshot displays the 'Criterios de Evaluación Actitudinal' section. The sidebar on the left includes the user profile 'Alfredo' (Administrador) and navigation options: Datos Generales, Departamentos, Municipios, Centros Educativos, Materias, Criterios Actitudinales (highlighted), Actitudes, and Otras Evaluaciones. The main content area features a search bar for 'Criterio Actitudinal' with an 'AGREGAR +' button. Below is a table with 7 rows, each representing a criterion.

N°	Criterio de Evaluación	
1	Perseverancia	
2	Optimismo	
3	Trabajo en la escuela	
4	Interpersonal	
5	Gratitud	
6	Inteligencia social	
7	Curiosidad	

Figura 4.19 Vista de sección Criterio actitudinal de evaluación.



The screenshot displays the 'Actitudes de Evaluación' section. The sidebar is identical to the previous screenshot, with 'Actitudes' highlighted. The main content area features a search bar for 'Actitud' with a 'Criterio' dropdown menu and an 'AGREGAR +' button. Below is a table with 7 rows, each representing an attitude and its associated criterion.

N°	Actitud	Criterio	
1	Termina lo que empieza	Perseverancia	
2	Proyectos a largo plazo	Perseverancia	
3	Trata fuerte aún fallando	Perseverancia	
4	Comprometido con metas	Perseverancia	
5	Se mantiene trabajando	Perseverancia	
6	Esfuerzo mejora su futuro	Optimismo	
7	Alternativas para mejorar	Optimismo	

Figura 4.20 Vista de sección Actitud.

FUNCIONALIDAD DE SECCIÓN CRITERIO ACTITUDINAL	
Cargar vista Criterio actitudinal	La vista se carga desde el menú de navegación por medio de una petición GET a la ruta /admin/criterios la cual es gestionada por el método index() del controlador CriterioController . El método devuelve la vista criterio.blade.php y una colección de objetos del modelo Criterio , utilizada para cargar el listado de registros en la vista, figura 4.19.
Registrar nuevo Criterio actitudinal	La vista cuenta con un formulario en la parte superior para realizar un nuevo registro, en el cual se debe especificar el nombre del criterio. El botón guardar realiza una petición Ajax de tipo POST para enviar la información al método store(Request \$request) del controlador CriterioController el cual ejecuta una sentencia INSERT en la base de datos, con el método save() del modelo Criterio .
Editar Criterio actitudinal	El botón de edición de cada registro en el listado de criterios actitudinales, realiza una petición por medio de Ajax, la cual es manejada con el método show(\$id) del controlador para obtener los datos del registro a modificar y enviarlos como respuesta. Dichos datos son cargados en el formulario de un modal.
Guardar edición de Criterio actitudinal	Al guardar la edición de los datos en el modal, se realiza la petición PUT para ejecutar el método update() del controlador el cual recibe como parámetro el id y los datos a modificar de un criterio específico para construir y ejecutar la sentencia UPDATE en la base de datos, con el método update() del modelo Criterio .

Tabla 4.5 Funcionalidad de Sección Criterio Actitudinal.

FUNCIONALIDAD DE SECCIÓN ACTITUD	
Cargar vista Actitud	La vista se carga desde el menú de navegación por medio de una petición GET a la ruta /admin/actitudes la cual es gestionada por el método index() del controlador ActitudController . El método devuelve la vista actitud.blade.php y una colección de objetos del modelo Actitud y Criterio , utilizada para cargar el listado de registros en la vista, figura 4.20.
Registrar nueva Actitud	La vista cuenta con un formulario en la parte superior para realizar un nuevo registro, en el cual se debe especificar el nombre y a qué criterio pertenece la actitud. El botón guardar realiza una petición Ajax de tipo POST para enviar la información al método store(Request \$request) del controlador ActitudController el cual ejecuta una sentencia INSERT en la base de datos, con el método save() del modelo Actitud .
Editar Actitud	El botón de edición de cada registro en el listado de actitudes, realiza una petición por medio de Ajax, la cual es manejada con el método show(\$id) del controlador para obtener los datos del registro a modificar y enviarlos como respuesta. Los datos son cargados en el formulario de un modal y mostrados en pantalla.
Guardar edición Actitud	Al guardar la edición de los datos en el modal, se realiza la petición PUT para ejecutar el método update() del controlador el cual recibe como parámetro el id y los datos a modificar de una actitud específica para construir y ejecutar la sentencia UPDATE en la base de datos, con el método update() del modelo Actitud .

Tabla 4.6 Funcionalidad de Sección Actitud.

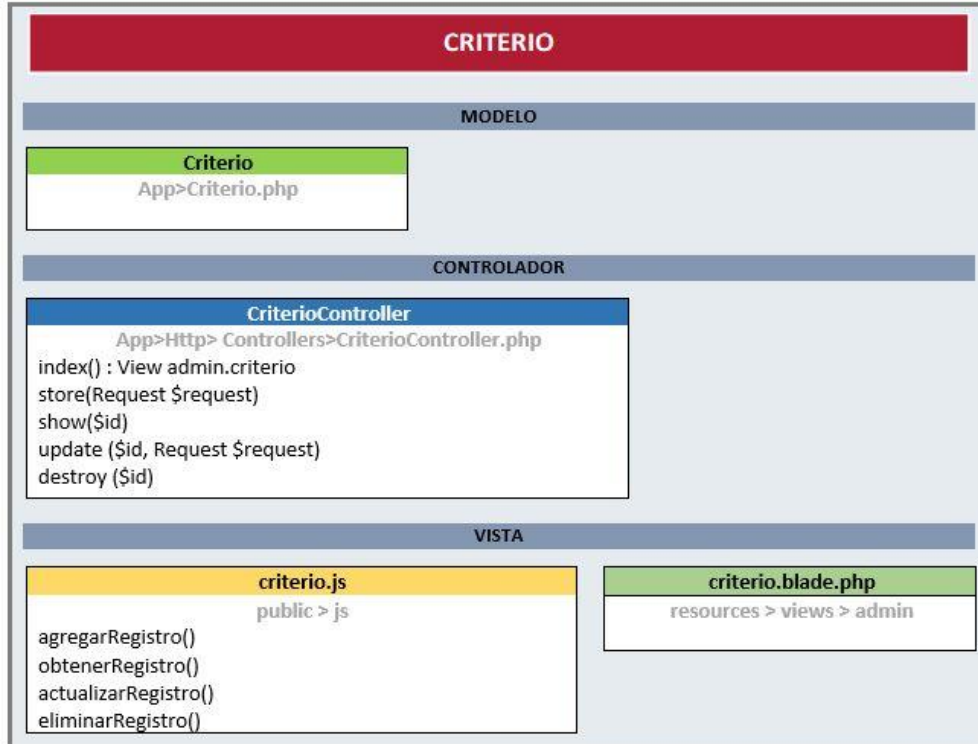


Figura 4.22 MVC de la sección Criterio Actitudinal.

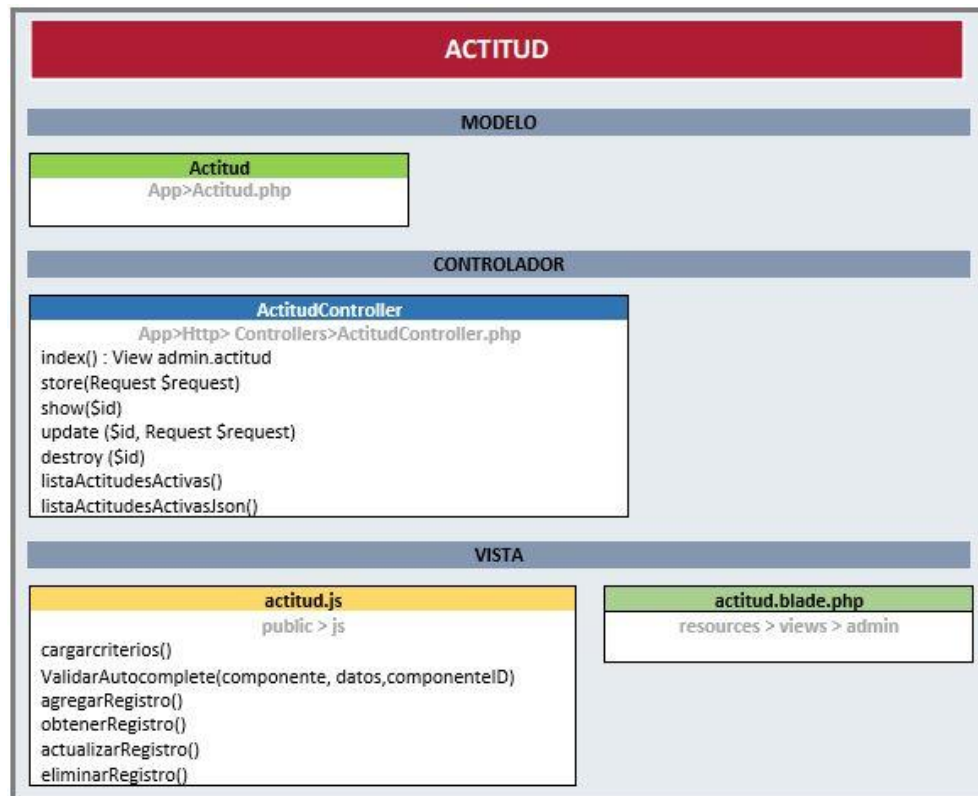


Figura 4.21 MVC de la sección Actitud.

Archivo	función	Descripción
municipio.js	cargarDepartamentos()	Utiliza la lista de Departamentos registrados, para devolver un objeto de dicha lista con el formato adecuado para cargarla en un elemento autocomplete de materialize css.
municipio.js	validarAutocomplete(componente, datos, componenteID)	Obtiene el valor ingresado del autocomplete componente , y lo compara con los valores en el arreglo Datos . El método asigna el ID del valor que coincide con el valor del autocomplete a la variable componenteID .
init.js	llenarAutocomplete(componente, datos, id)	Carga el autocomplete componente con los valores del arreglo datos . Establece la opción onAutocomplete de componente para que se guarde el id del valor autocompletado en la variable ID .
centroeducativo.js	cargarmunicipios()	Utiliza la lista de Municipios registrados, para devolver un objeto de dicha lista con el formato adecuado para cargarla en un elemento autocomplete de materialize css.
materia.js	obtenerCodigoMateria(chkce, chkop)	Verifica los checkbox del tipo de materia del formulario de nuevo registro, para devolver el código correspondiente a la materia, que indica si la materia pertenece al programa oportunidades, a los centros escolares o a ambos.
actitud.js	cargarCriterios()	Utiliza la lista de Criterios registrados, para devolver un objeto de dicha lista con el formato adecuado para cargarla en un elemento autocomplete de materialize css.
ActitudController.php	ListaActitudesActivas()	Devuelve una colección de actitudes cuyo estado es activo.
ActitudController.php	ListaActitudesActivasJson()	Devuelve un objeto Json de actitudes cuyo estado es activo.

Tabla 4.7 Métodos y funciones de las clases del submódulo Datos Generales.

En la tabla 4.7 se presentan los métodos y funciones complementarias de las clases pertenecientes a todas las secciones del submódulo Datos generales. La tabla se utiliza como referente para identificar de forma fácil y rápida en que consiste cada una de las porciones de código incluidas en cada una de las operaciones que se ejecutan al llevar una tarea específica dentro del sistema.

4.2.2 SUBMÓDULO: PERSONAL

El submódulo de Personal, corresponde al registro y actualización de la información del personal perteneciente al programa oportunidades y que hace uso del sistema. Toda la información personal de sus empleados estará registrada en este módulo, también se administrarán los roles que cada miembro del personal desempeña.

Las secciones del submódulo Personal, extienden de igual forma la plantilla Blade ***admin.blade.php***. En la imagen 4.23 presentan las rutas pertenecientes a los controladores del submódulo, y en la imagen 4.24 se presenta la porción del menú de navegación en la plantilla que contiene los enlaces de las secciones del submódulo.

```
Route::resource('/admin/cargos', 'CargoController');
Route::resource('/admin/empleados', 'EmpleadoController');
```

Figura 4.23 Rutas del módulo Personal.

```
<!--PERSONAL-->
<li>
  <div class="collapsible-header waves-effect principal-item"><i
    class="material-icons">assignment_ind</i>Personal</div>
  <div class="collapsible-body">
    <ul>
      <li><a href="/admin/cargos" class="waves-effect secondary-item">Cargos</a></li>
      @if(Auth::user()->is_admin == 1 && Auth::user()->is_docente == 0)
      <li><a href="/admin/empleados" class="waves-effect secondary-item">Empleados</a></li>
      @elseif(Auth::user()->is_admin == 1 && Auth::user()->is_docente == 1)
      <li><a href="/admin/empleados" class="waves-effect secondary-item" style="display:none">Empleados</a></li>
      @endif
    </ul>
  </div>
</li>
```

Figura 4.24 Enlaces del módulo personal en el menú de navegación.

4.2.2.1 SECCIÓN CARGOS

La sección Cargos está destinada a registrar todos los cargos que desempeña el personal del programa Oportunidades, cuya información será registrada en el sistema. Por defecto el sistema posee el registro de los cargos cuyos roles están definidos para los usuarios del sistema: Administrador, asistente administrativo y docente.

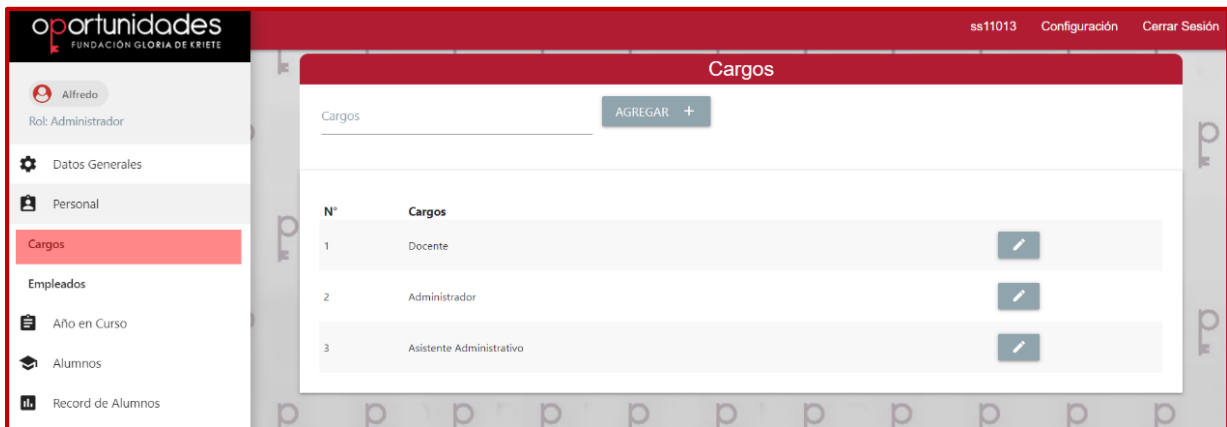


Figura 4.25 Vista de la sección Cargo.

FUNCIONALIDAD DE SECCIÓN CARGO	
Cargar vista Cargo	La vista se carga desde el menú de navegación por medio de una petición GET a la ruta <code>/admin/cargos</code> la cual es gestionada por el método <code>index()</code> del controlador <code>CargoController</code> . El método devuelve la vista <code>cargo.blade.php</code> y una colección de objetos del modelo <code>Cargo</code> , utilizada para cargar el listado de registros en la vista, figura 4.25.
Registrar nuevo Cargo	La vista cuenta con un pequeño formulario en la parte superior para realizar un nuevo registro, en el cual se debe especificar el nombre del cargo a registrar. El botón guardar realiza una petición Ajax de tipo POST para enviar la información al método <code>store(Request \$request)</code> del controlador <code>CargoController</code> el cual ejecuta una sentencia INSERT en la base de datos, con el método <code>save()</code> del modelo <code>Cargo</code> .
Editar Cargo	El botón de edición de cada registro en el listado de cargos, realiza una petición por medio de Ajax, la cual es manejada con el método <code>show(\$id)</code> del controlador para obtener los datos del registro a modificar por medio de su ID y enviarlos como respuesta. Los datos son cargados en el formulario de un modal y mostrados en pantalla.
Guardar edición Cargo	Al guardar la edición de los datos en el modal, se realiza la petición PUT para ejecutar el método <code>update()</code> del controlador el cual recibe como parámetro el id y los datos a modificar de un cargo específico para construir y ejecutar la sentencia UPDATE en la base de datos, con el método <code>update()</code> del modelo <code>Cargo</code> .

Tabla 4.8 Funcionalidad de Sección Cargo.

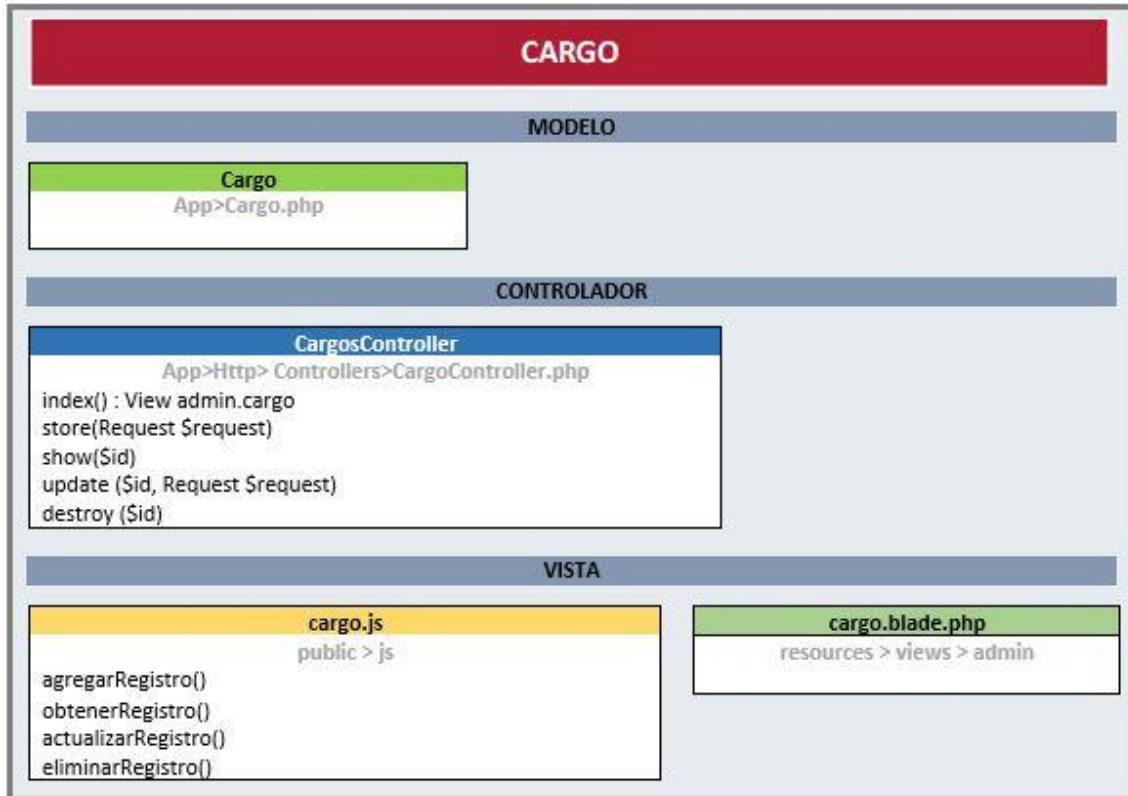


Figura 4.26 MVC de la sección Cargo.

4.2.2.2 SECCIÓN EMPLEADOS

La sección de Empleados implementa las operaciones básicas CRUD de listar, registrar, modificar y eliminar los registros de los empleados del programa Oportunidades. Además se encarga de crear el usuario para los empleados cuyo cargo asignado sea parte de los roles definidos para el sistema: Administrador, Asistente administrativo y Docente.

oportunidades
FUNDACIÓN GLORIA DE KRITTE

ss11013 Configuración Cerrar Sesión

Alfredo
Rol: Administrador

Datos Generales

Personal

Cargos

Empleados

Año en Curso

Alumnos

Record de Alumnos

Mostrar 10 registros

Carnet	Cargo	Nombres	Apellidos	Telefono	
GG001	Administrador	Doris Jeannette	Granados Granados	7255-4395	
GP007	Docente	Elmer Israel	Galdámez Pérez	7751-9276	
MC006	Docente	Mónica Saraí	Menjívar de Chávez	7015-9973	
MG002	Administrador	Silvia Lorena	Munguía Granados	7741-9929	
MichellePlatero	Administrador	Jocelyn Michelle	Castaneda Platero	7768-6144	
PL005	Docente	Iris Violeta	Peraza Linares	6020-5099	
RA008	Docente	Iliana Jeannette	Renderos Arrué	6112-3201	
RM004	Docente	Lorena Beatriz	Ramírez Martínez	7352-7462	
SE009	Docente	Nanci Rebeca	Salas Eguizabal	7031-3137	

Mostrando página 1 de 1

Anterior 1 Siguiente

Figura 4.28 Vista de la sección Empleado.

Empleados

+ Agregar Empleado

Nombre

Apellidos Sexo F M

Fecha Nacimiento:DD-MM-YYYY

DUI:0000000-0

NIT:0000-000000-000-0

Carnet

Cargo

Teléfono

Municipio

Dirección

Email

Estado

Figura 4.27 Formulario para registro de Empleado.

FUNCIONALIDAD DESECCIÓN EMPLEADO	
Cargar vista Empleado	La vista se carga desde el menú de navegación por medio de una petición GET a la ruta <code>/admin/empleados</code> la cual es gestionada por el método <code>index()</code> del controlador <code>EmpleadoController</code> . El método devuelve la vista <code>empleado.blade.php</code> y una colección de objetos del modelo <code>Empleado</code> , utilizada para cargar el listado de registros en la vista, figura 4.25.
Registrar nuevo Empleado	La vista cuenta con un pequeño formulario en la parte superior para realizar un nuevo registro, en el cual se debe especificar el nombre del cargo a registrar. El botón guardar realiza una petición Ajax de tipo POST para enviar la información al método <code>store(Request \$request)</code> del controlador <code>CargoController</code> el cual ejecuta una sentencia INSERT en la base de datos, utilizando el método <code>save()</code> del modelo <code>Empleado</code> . Además de registrar los datos del empleado, también crea un registro en la tabla <code>Usuario</code> de la base de datos siempre y cuando el cargo del empleado registrado corresponda a alguno de los roles definidos para el sistema. Si el empleado registrado ocupa otro cargo, solamente se creará el registro en la tabla <code>Empleado</code> .
Editar Empleado	El botón de edición de cada registro en el listado de cargos, realiza una petición por medio de Ajax, la cual es manejada con el método <code>show(\$id)</code> del controlador para obtener los datos del registro a modificar por medio de su ID y enviarlos como respuesta. Los datos son cargados en el formulario de un modal y mostrados en pantalla.
Guardar edición Empleado	Al guardar la edición de los datos en el modal, se realiza la petición PUT para ejecutar el método <code>update()</code> del controlador el cual recibe como parámetro el id y los datos a modificar de un cargo específico para construir y ejecutar la sentencia UPDATE en la base de datos, con el método <code>update()</code> del modelo <code>Cargo</code> .

Tabla 4.9 Funcionalidad de Sección Empleado.

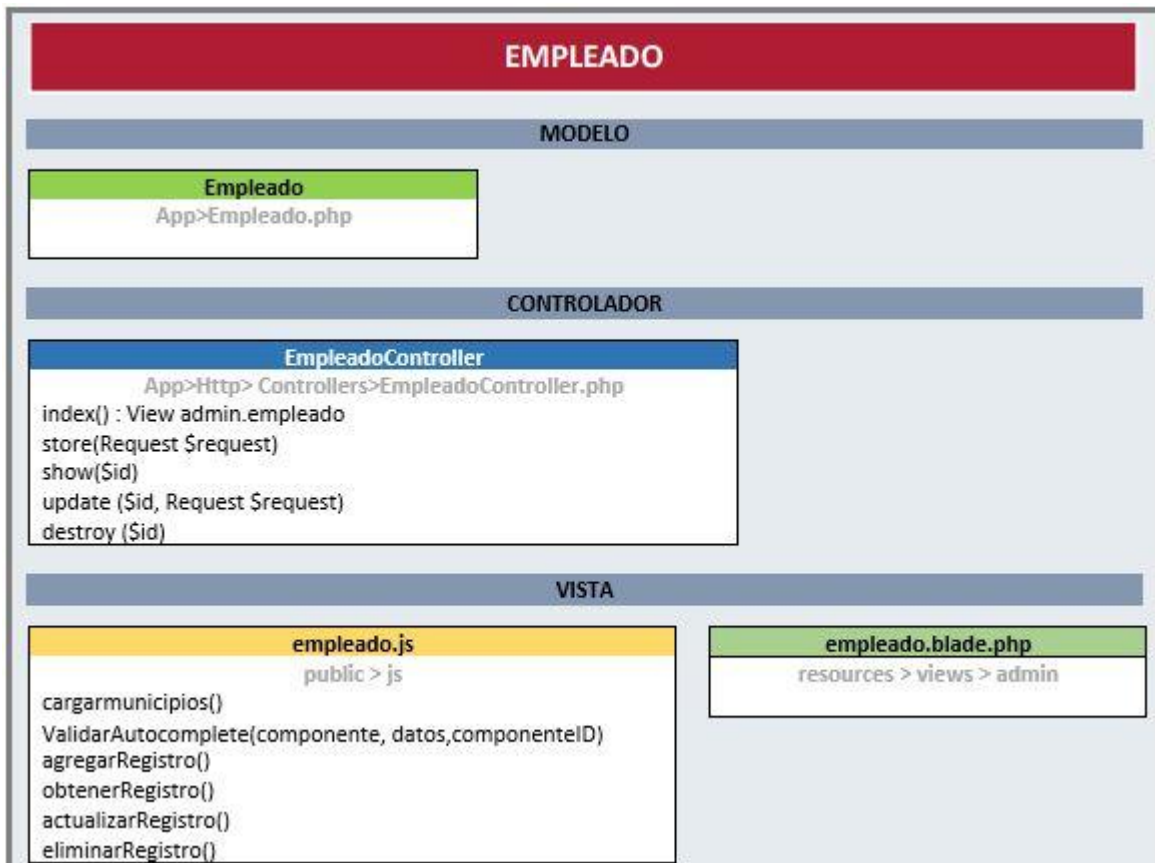


Figura 4.29 MVC de sección Empleado.

4.2.3 SUBMODULO: ALUMNOS

EL submódulo de Alumnos está diseñado para realizar las tareas de registro y edición referente a toda la información de los alumnos, como lo son los datos de contacto del alumno y de los adultos responsables del mismo y los datos del centro educativo en el que cursan el programa escolar del MINED.

```
Route::resource('/admin/alumnos', 'AlumnoController');
Route::get('nuevo-alumno', 'AlumnoController@create');

Route::post('/admin/alumnos/actualizar/{parametro}', 'AlumnoController@actualizar');
Route::resource('/admin/tiporesponsable', 'TipoResponsableController');
Route::resource('/admin/responsables', 'ResponsableController');
Route::resource('/admin/ocupaciones', 'OcupacionController');
Route::post('/admin/bitacorás', 'BitacoraCEA@store');

Route::get('/admin/buscador-alumno/', 'AlumnoController@buscar');
Route::get('/admin/buscador-alumno/{parametro}', 'AlumnoController@buscar');
```

Figura 4.30 Rutas del submódulo Alumnos.

```
<!--ALUMNOS-->
<li>
  <div class="collapsible-header waves-effect principal-item"><i
    class="material-icons">school</i>Alumnos</div>
  <div class="collapsible-body">
    <ul>
      <li><a href="/admin/alumnos" class="waves-effect secondary-item">Listado de Alumnos</a></li>
      <li><a href="/admin/buscador-alumno" class="waves-effect secondary-item">Buscador de Alumnos</a></li>
    </ul>
  </div>
</li>
```

Figura 4.31 Enlaces del submódulo Alumno en el menú de navegación.

oportunidades FUNDACIÓN GLORIA DE KRIETE

ss11013 Configuración Cerrar Sesión

Alfredo Rol. Administrador

Datos Generales

Personal

Año en Curso

Alumnos

Listado de Alumnos

Buscador de Alumnos

Record de Alumnos

Detalle de Alumnos

Mostrar 25 registros

Buscar:

Carnet	Nombres	Apellidos	Clase	Turno
2018-SA-FT-0001	Michelle Abigail	Aguilar Figueroa	CLASS-2020	Matutino
2018-SA-FT-0003	Evelyn Karina	Alarcón Ramirez	CLASS-2020	Matutino
2018-SA-FT-0004	Tatiana Alejandra	Arce Tejada	CLASS-2020	Matutino
2018-SA-FT-0005	Daniela Elizabeth	Arias Molina	CLASS-2020	Matutino
2018-SA-FT-0007	Kenia Abigail	Calderón Flores	CLASS-2020	Vespertino
2018-SA-FT-0008	Jeansy Stephannie	Calderón Hernández	CLASS-2020	Matutino
2018-SA-FT-0010	José Rodrigo	Castellanos Calles	CLASS-2020	Matutino
2018-SA-FT-0011	Madelin Rocio	Chavarría García	CLASS-2020	Matutino
2018-SA-FT-0012	Allison Iveth	Chávez Pérez	CLASS-2020	Matutino
2018-SA-FT-0014	Elizabeth Estefanía	Colindres Corea	CLASS-2020	Vespertino
2018-SA-FT-0015	Alejandra Lizbeth	Córdova Pacheco	CLASS-2020	Matutino
2018-SA-FT-0017	Wilma Rocío	Cuéllar Silva	CLASS-2020	Matutino

Figura 4.32 Vista del submódulo Alumnos.

Nuevo Alumno

CARNET

FOTO

Nombres

Apellidos

Fecha Nacimiento: DD-MM-YYYY SEXO: F M

Dirección

Municipio

Email

Número de Teléfono

Centro Escolar

Turno educativo Seleccionar

CLASS Seleccionar

Estado Seleccionar

AGREGAR +

Figura 4.33 Vista de registro de nuevo alumno.

CARNET
2018-SA-FT-0001

Nombres Apellidos

Fecha Nacimiento: DD-MM-YYYY
08-12-2002 SEXO: F M

Dirección Municipio
1-Santa Ana

Email Número de Teléfono Centro Escolar
88018-CENTRO ESCOLAR CATOLICO SAN VICENTE DE PAUL

Turno educativo CLASS Estado
Matutino CLASS-2020 Activo

Observaciones

ACTUALIZAR INFORMACIÓN +

Responsables de Alumno

+ AGREGAR RESPONSABLE + AGREGAR TIPO DE RESPONSABLE + AGREGAR OCUPACIÓN

DUI	Nombres	Apellidos	Teléfono	Parentesco	Ocupación
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Madre	Empleada

Centros Escolares

+ CAMBIO DE CENTRO ESCOLAR

Centro Escolar	Grado Académico	Año
<input type="text"/>	<input type="text"/>	<input type="text"/>

Figura 4.34 Vista actualización de un alumno.

FUNCIONALIDAD DE SECCIÓN ALUMNO	
Cargar Listado de alumnos	El listado de alumnos se carga desde el menú de navegación por medio de una petición GET a la ruta /admin/alumnos la cual es gestionada por el método index() del controlador AlumnoController . El método devuelve la vista listado-alumnos.blade.php y una colección de objetos del modelo Alumno , con el listado completo de los alumnos activos. Dicho listado es cargado en la vista, en un datatable , figura 4.32.
Registrar nuevo Alumno	El listado de alumnos cuenta con un botón en la esquina inferior derecha, para agregar un nuevo alumno. Dicho botón realiza una petición Ajax tipo GET a la ruta /nuevo-alumno . De esta forma se ejecuta la función create() del controlador AlumnoController , el cual devuelve la vista alumno.blade.php y los datos necesarios para cargar el formulario de nuevo registro la vista. Figura 4.33. Al guardar los datos del nuevo alumno, dichos datos son enviados al controlador y manejados por la función store() con la cual se realiza el registro correspondiente. Además de registrar el nuevo alumno en la base de datos con el método save() del modelo Alumno . Además la función store() crea el registro correspondiente del alumno en la tabla Usuario , utilizando el modelo correspondiente. Una vez realizados los registros, la función devuelve la vista alumnodetalle.blade.php con los datos del alumno registrado.
Editar Alumno	Para realizar la edición de la información de un alumno, en la vista de listado de alumnos el carnet de cada alumno cuenta con un enlace que realiza una petición Http la cual es manejada con el método show(\$id) del controlador AlumnoController . El método obtiene el carnet del alumno como parámetro y obtiene sus datos los cuales son enviados como respuesta, junto con la vista alumnodetalle.blade.php
Guardar edición de alumno	Al presionar el botón actualizar información se envían los datos editados al controlador, por medio de una petición que es manejada con la función actualizar() la cual ejecuta la sentencia UPDATE para registrar la actualización de datos del alumno en la base de datos.
Registro de Responsables de alumno	Para poder realizar el registro de los responsables desde la vista alumno-detalle.blade.php se utilizan los modelos Responsable , TipoResponsable y Ocupacion , cada uno de los cuales cuenta con su controlador, implementado de la forma que se ha manejado hasta el momento. La vista cuenta con un modal para realizar el registro de la información de contacto de los responsables del alumno. La sección cuenta con la opción de registrar nuevos tipos de responsable y ocupaciones si no se encuentran registradas. Cada entidad es registrada por medio de la función store() de su controlador correspondiente.
Registro cambio de Centro Educativo	Para registrar el historial de un cambio de centro educativo del alumno, en la vista alumno-detalle.blade.php se utilizar el modelo BitacoraCEA.php y su respectivo controlador, para poder registrar los cambios de centro escolar que realiza el alumno durante el curso del programa Oportunidades.
Buscador de Alumno	Para realizar la búsqueda de un alumno se utiliza la función buscar(\$id) del controlador AlumnoController . Devuelve la vista buscador-alumno.blade.php , creada para poder realizar la búsqueda de un alumno, especificando su carnet. El buscador muestra los resultados independientemente del estado del alumno. La función es accedida por medio de la ruta admin/buscador-alumno/{carnet} . Si se especifica el carnet, devuelve la vista con los resultados de la búsqueda; si no, solamente devuelve la vista, para poder realizar una nueva búsqueda.

Tabla 4.10 Funcionalidad de submódulo Alumno.



Figura 4.35 MVC del submódulo Alumno.

4.2.4 SUBMODULO: AÑO EN CURSO

Este submódulo está dedicado a encapsular todas las tareas que se encargan de estar registrando y actualizando cada año, los datos necesarios para continuar con las tareas de un nuevo periodo escolar. En dicho submódulo, se especifica como estarán organizados los alumnos en las secciones, así como también que materias y a que secciones impartirán clases cada uno de los docentes. Establece las asignaciones para que el expediente del alumno continúe actualizándose en el transcurso de un nuevo periodo escolar.

```
//AÑO EN CURSO
Route::resource('/admin/clases', 'ClassController');
Route::resource('/admin/secciones', 'SeccionController');
Route::resource('/admin/asignacion-docentes', 'GrupoMateriaController');
Route::resource('/admin/asignacion-docentes-bitacora', 'BitacoraAsignacionDocenteController');
Route::get('/admin/asignacion-docentes/busqueda/{parametro}', 'GrupoMateriaController@buscarxanio');
Route::resource('/admin/asignacion-alumnos', 'AlumnoSeccionController');
```

Figura 4.36 Rutas de las secciones del submódulo Año en curso.

```
<!--AÑO EN CURSO-->
<li>
  <div class="collapsible-header waves-effect principal-item"><i
    class="material-icons">assignment</i>Año en Curso</div>
  <div class="collapsible-body">
    <ul>
      <li><a href="/admin/clases" class="waves-effect secondary-item">Clases</a></li>
      <li><a href="/admin/secciones" class="waves-effect secondary-item">Secciones</a></li>
      <li><a href="/admin/asignacion-alumnos" class="waves-effect secondary-item">Asignación Secciones-Alumnos</a></li>
      <li><a href="/admin/asignacion-docentes" class="waves-effect secondary-item">Asignación Secciones-Docentes</a></li>
      <li><a href="/admin/importar-notas" class="waves-effect secondary-item">Registro de Notas</a></li>
      <li><a href="/admin/notas-extras" class="waves-effect secondary-item">Registro Certificaciones</a></li>
      <li><a href="/admin/boletas-notas" class="waves-effect secondary-item">Generar Boletas de Notas</a></li>
    </ul>
  </div>
</li>
```

Figura 4.37 Enlaces en el menú del submódulo Año en curso.

4.2.4.1 SECCIÓN CLASE

En la sección de Clase se definen las clases o grupos de alumnos que acuden a la fundación por año. Cada año se define una nueva clase para los nuevos alumnos que ingresan.

La vista **class.blade.php** muestra las clases que se encuentran activas en el año en curso. Cuenta con una sección desplegable para realizar el nuevo registro de una Clase. Cada registro en la vista cuenta con los botones para realizar las operaciones de edición y eliminación.

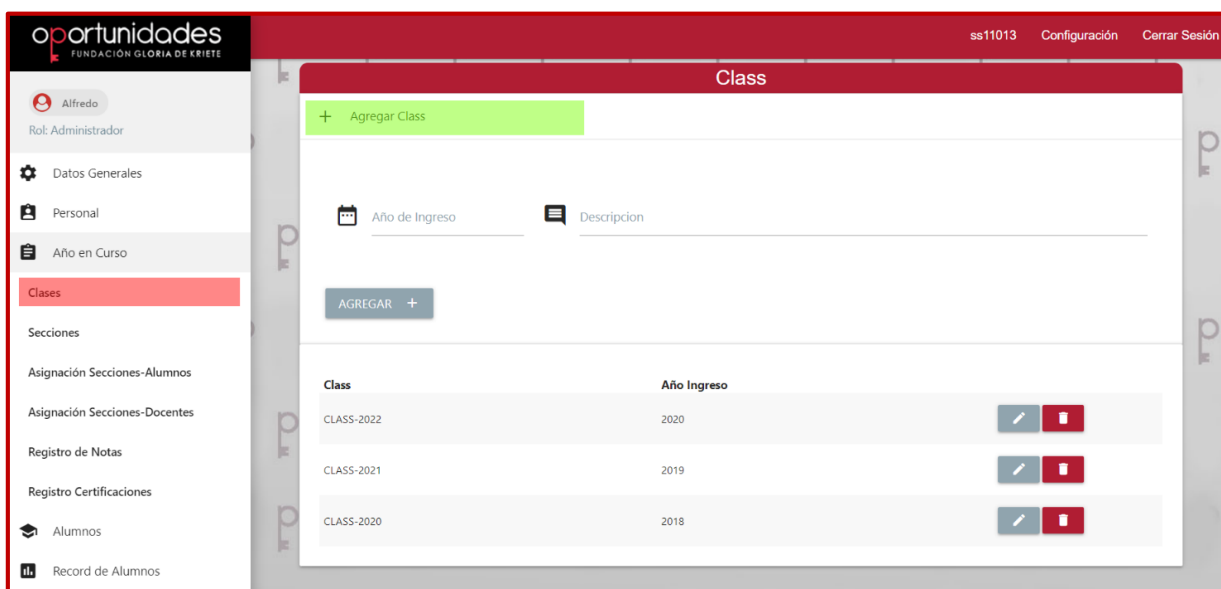


Figura 4.38 Vista de la sección Class.

FUNCIONALIDAD DE SECCIÓN CLASE	
Cargar vista Clase	La vista se carga desde el menú de navegación por medio de una petición GET a la ruta <code>/admin/clases</code> la cual es gestionada por el método <code>index()</code> del controlador <code>ClassController</code> . El método devuelve la vista <code>class.blade.php</code> y una colección de objetos del modelo <code>Clase</code> , utilizada para cargar las clases del año en curso, figura 4.38.
Registrar nuevo Clase	La vista cuenta con un pequeño formulario desplegable en la parte superior para realizar un nuevo registro. El botón guardar realiza una petición Ajax de tipo POST para enviar la información al método <code>store(Request \$request)</code> del controlador <code>ClassController</code> el cual ejecuta una sentencia INSERT en la base de datos, utilizando el método <code>save()</code> del modelo <code>Clase</code> .
Editar Clase	El botón de edición de cada registro en el listado de clases, realiza una petición por medio de Ajax, la cual es manejada con el método <code>show(\$id)</code> del controlador para obtener los datos del registro a modificar por medio de su ID y enviarlos como respuesta. Los datos son cargados en el formulario de un modal y mostrados en pantalla.
Guardar edición de Clase	Al guardar la edición de los datos en el modal, se realiza la petición PUT para ejecutar el método <code>update()</code> del controlador el cual recibe como parámetro el id y los datos a modificar de una clase específica para construir y ejecutar la sentencia UPDATE en la base de datos, con el método <code>update()</code> del modelo <code>Clase</code> .
Eliminar Clase	El botón de eliminar de cada registro en el listado de clases, realiza una petición de tipo DELETE por medio de Ajax, la cual es manejada con la función <code>destroy()</code> del controlador <code>ClassController</code> .

Tabla 4.11 Funcionalidad de Sección Clase.

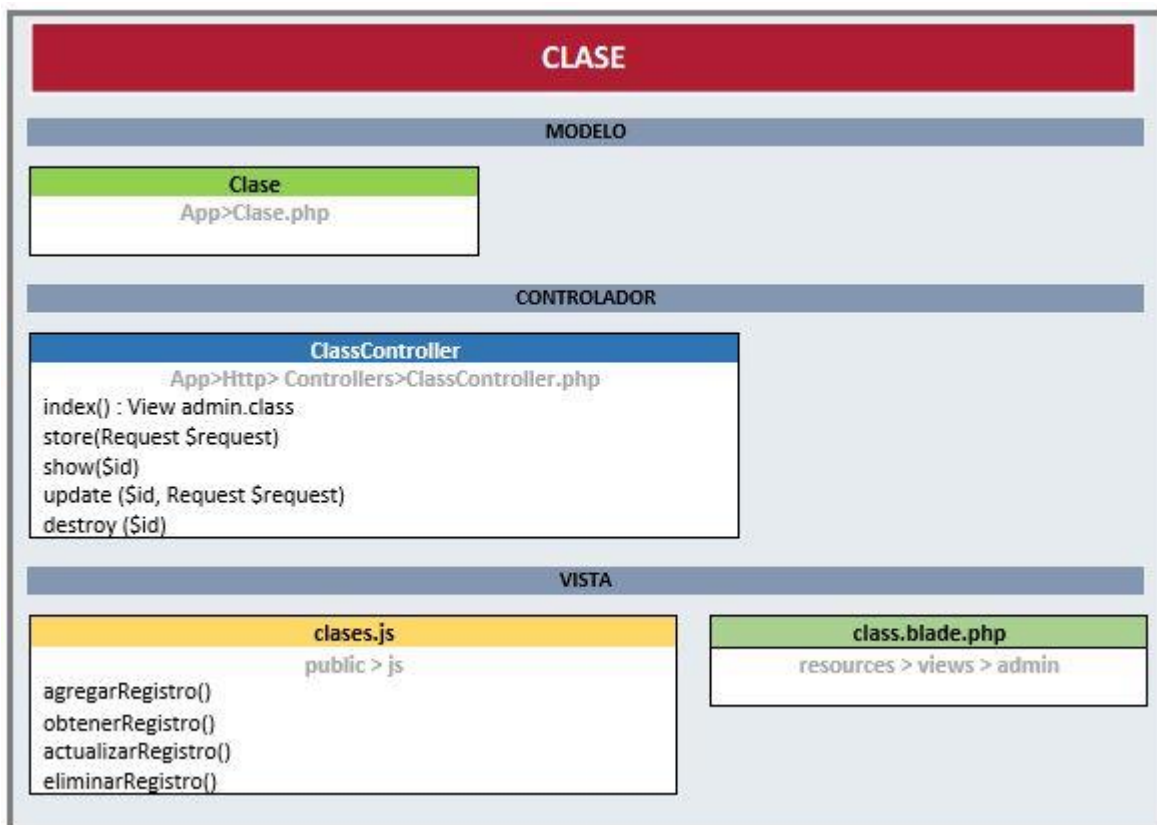


Figura 4.39 MVC de la sección Clase.

4.2.4.2 SECCIÓN: SECCIONES DEL PROGRAMA

Cada año se realiza el registro de las secciones en que estarán organizados los alumnos de cada Clase (año escolar). El número de secciones creadas por clase, dependerá de la cantidad de alumnos activos de la misma.

La vista muestra la lista de secciones de las clases activas en el año en curso. Cuenta con una sección desplegable para realizar el nuevo registro de una nueva sección. Cada registro en la vista cuenta con los botones para realizar las operaciones de edición y eliminación.

The screenshot displays the 'Sección' management interface. The top header includes the user name 'Alfredo', role 'Rol: Administrador', and session information 'ss11013 Configuración Cerrar Sesión'. The main content area features a table with the following data:

Sección	Clase	Año	Operaciones
A1-CLASS-2020	CLASS-2020	2020	[Edit] [Delete]
A2-CLASS-2020	CLASS-2020	2020	[Edit] [Delete]
B1-CLASS-2021	CLASS-2021	2020	[Edit] [Delete]
B2-CLASS-2021	CLASS-2021	2020	[Edit] [Delete]
B3-CLASS-2021	CLASS-2021	2020	[Edit] [Delete]
C1-CLASS-2022	CLASS-2022	2020	[Edit] [Delete]
C2-CLASS-2022	CLASS-2022	2020	[Edit] [Delete]

Figura 4.40 Vista secciones por clase.

FUNCIONALIDAD DE SECCIONES	
Cargar vista Sección	La vista se carga desde el menú de navegación por medio de una petición GET a la ruta <code>/admin/secciones</code> la cual es gestionada por el método <code>index()</code> del controlador <code>SeccionController</code> . El método devuelve la vista <code>seccion.blade.php</code> y una colección de objetos del modelo <code>Seccion</code> , utilizada para cargar las secciones del año en curso, figura 4.40.
Registrar nuevo Sección	La vista cuenta con un formulario desplegable en la parte superior para realizar un nuevo registro. El botón guardar realiza una petición Ajax de tipo POST para enviar la información al método <code>store(Request \$request)</code> del controlador <code>SeccionController</code> el cual ejecuta una sentencia INSERT en la base de datos, utilizando el método <code>save()</code> del modelo <code>Seccion</code> .
Editar Sección	El botón de edición de cada registro en el listado de secciones, realiza una petición por medio de Ajax, la cual es manejada con el método <code>show(\$id)</code> del controlador para obtener los datos del registro a modificar por medio de su ID y enviarlos como respuesta. Los datos son cargados en el formulario de un modal y mostrados en pantalla.
Guardar edición Sección	Al guardar la edición de los datos en el modal, se realiza la petición PUT para ejecutar el método <code>update()</code> del controlador el cual recibe como parámetro el id y los datos a modificar de una sección específica para construir y ejecutar la sentencia UPDATE en la base de datos, con el método <code>update()</code> del modelo <code>Seccion</code> .
Eliminar Sección	El botón de eliminar de cada registro en el listado de secciones, realiza una petición de tipo DELETE por medio de Ajax, la cual es manejada con la función <code>destroy()</code> del controlador <code>SeccionController</code> .

Tabla 4.12 Funcionalidad de Secciones.

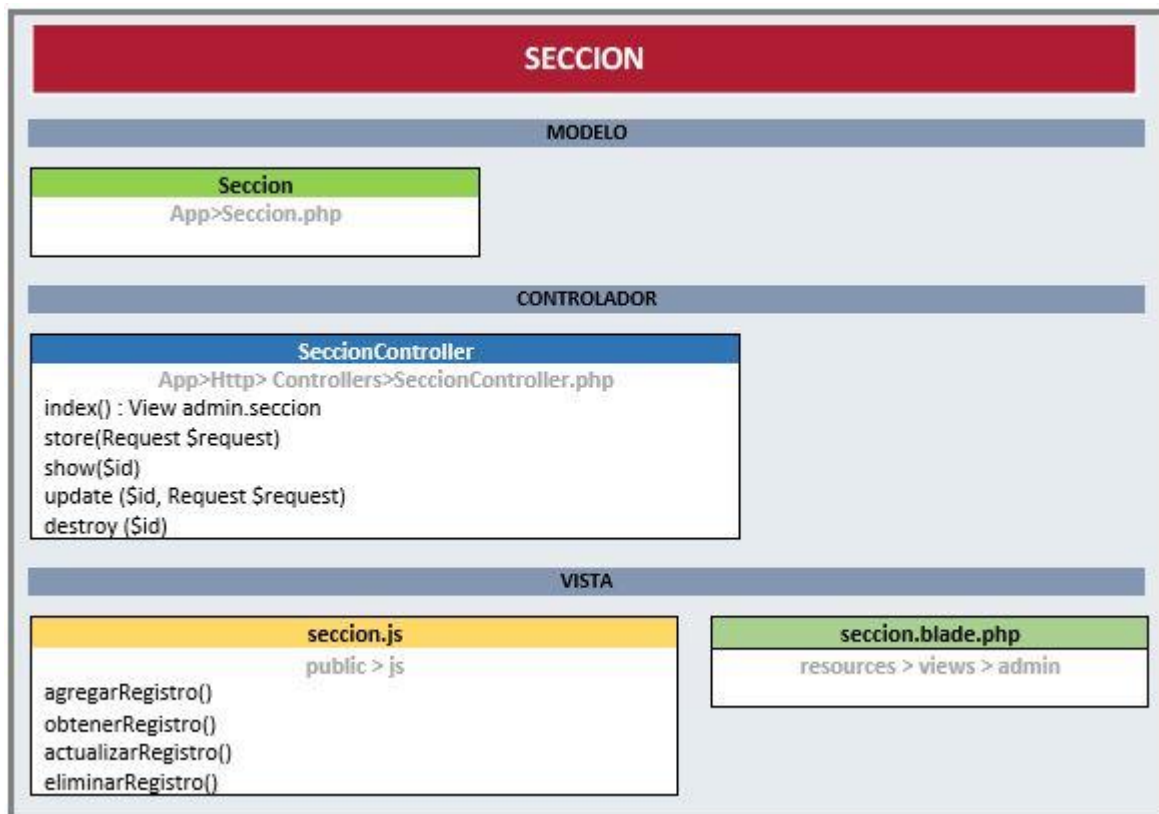


Figura 4.41 MVC de sección.

4.2.4.3 SECCION ASIGNACIÓN ALUMNO-SECCIÓN

La sección Asignación Secciones-alumno es la que permite realizar el registro de la organización de los alumnos en las secciones creadas para el año escolar en curso. La asignación se realiza de forma arbitraria de acuerdo a la disponibilidad de los alumnos con los horarios de clase asignados a cada sección.

Asignación de Alumnos a Secciones

Elegir CLASS

Secciones

ALUMNO

Figura 4.42 Selector de Clase en vista Asignación de alumnos a secciones.

oportunidades FUNDACIÓN GLORIA DE KRIETE

ss11013 Configuración Cerrar Sesión

Alfredo Rol: Administrador

Datos Generales

Personal

Año en Curso

Clases

Secciones

Asignación Secciones-Alumnos

Asignación Secciones-Docentes

Registro de Notas

Registro Certificaciones

Alumnos

Record de Alumnos

Asignación de Alumnos a Secciones

CLASS-2022

Secciones

ALUMNO	C1	C2	Flotante		
Argueta Martinez Erick Abid	<input checked="" type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	<input type="button" value="✎"/>	<input type="button" value="🔒"/>
Avalos Sanchez Natalia Michelle	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>	<input type="button" value="✎"/>	<input type="button" value="🔒"/>
Berrios Alvarenga Daniel Osvaldo	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>	<input type="button" value="✎"/>	<input type="button" value="🔒"/>
Bolaños Menjivar Nancy María	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>	<input type="button" value="✎"/>	<input type="button" value="🔒"/>
Canesa Aguilar Javier Ernesto	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>	<input type="button" value="✎"/>	<input type="button" value="🔒"/>
Cardona Méndez Carlos Rafael	<input checked="" type="radio"/>	<input type="radio"/>	<input type="checkbox"/>	<input type="button" value="✎"/>	<input type="button" value="🔒"/>
Chacón Silva Elizabeth Rachel	<input type="radio"/>	<input checked="" type="radio"/>	<input type="checkbox"/>	<input type="button" value="✎"/>	<input type="button" value="🔒"/>

Figura 4.43 Vista Asignación de alumnos a secciones.

Figura 4.44 Modal de edición de asignaciones alumno-sección.

FUNCIONALIDAD DE ASIGNACIÓN DE ALUMNOS A SECCIÓN	
Cargar vista	La vista se carga desde el menú de navegación por medio de una petición GET a la ruta <code>/admin/asignacion-alumnos</code> la cual es gestionada por el método <code>index()</code> del controlador <code>AlumnoSeccionController</code> . El método devuelve la vista <code>asignacion-alumnos.blade.php</code> y una lista de Clases del año en curso para llenar el selector, figura 4.42.
Seleccionar Clase	Al seleccionar una clase, se ejecuta una petición Ajax con la cual se llama a la función <code>show(\$clase)</code> del controlador <code>AlumnoSeccionController</code> . La función devuelve el listado de los alumnos pertenecientes a la clase especificada, además devuelve las secciones y las asignaciones que ya han sido registradas. Los datos obtenidos con la petición son cargados en la vista, por medio de las funciones: <code>cargarSecciones(sec)</code> , <code>cargarAlumnos(alumnos, sec)</code> , <code>cargarAsignados(asig, alumnos)</code> del archivo <code>alumnoseccion.js</code>
Guardar Asignación de alumnos	Para realizar las asignaciones, en la vista <code>asignacion-alumnos.blade.php</code> se marca el botón de radio correspondiente a la sección en que cada alumno será asignado. Adicionalmente se marca el checkbox de la columna flotante si el alumno contará con flexibilidad de horarios, figura 4.43. Una vez marcadas todas las asignaciones, con el botón Guardar se ejecuta la función <code>guardar_asignaciones()</code> del archivo <code>alumnoseccion.js</code> , la cual construye un arreglo de datos con todas las asignaciones hechas y lo envía a la función <code>store(Request \$request)</code> del controlador, el cual realiza los registros correspondientes en la base de datos.
Editar asignación de alumno	Cada registro en la lista de asignaciones realizadas, cuenta con el botón para realizar la modificación de la asignación del alumno correspondiente. Al iniciar una edición se abre el modal en el cual se especifica el cambio que se hará, y en qué periodo del año escolar pertenece el cambio figura 4.44. Al guardar los cambios, se ejecuta la función <code>actualizar_asignacion()</code> del archivo <code>alumnoseccion.js</code> , con la cual se realiza la petición Https para ejecutar la función <code>update(\$id, Request \$request)</code> . Además de ejecutar la sentencia UPDATE correspondiente en la base de datos, la función también realiza un registro nuevo en la tabla <code>log_cambio_seccion_alumno</code> con el modelo correspondiente.

Tabla 4.13 Funcionalidad de Asignación de alumnos a sección.

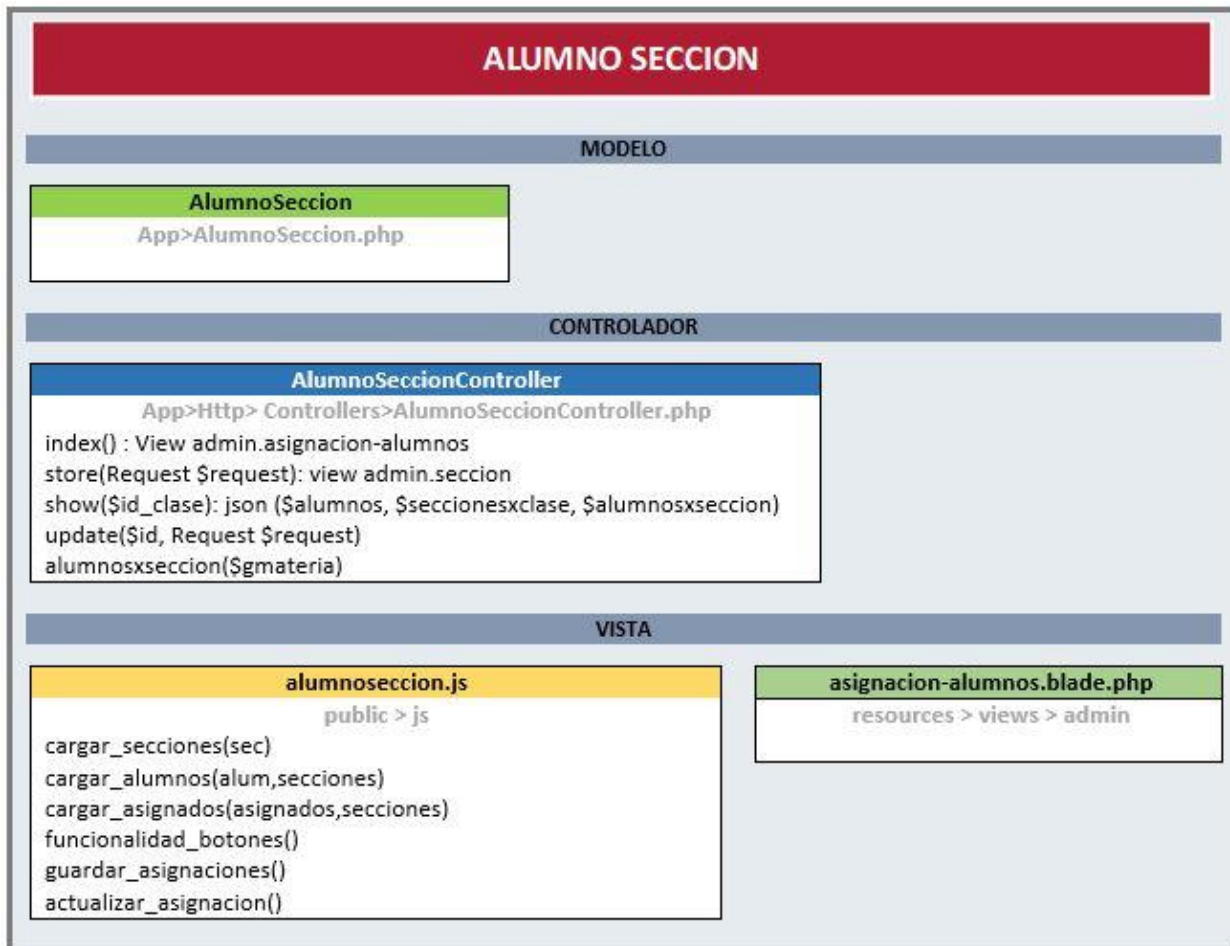


Figura 4.45 MVC de sección asignación alumnos a sección.

4.2.4.4 SECCION: ASIGNACIÓN SECCIONES-DOCENTE

Esta sección permite al administrador realizar la asignación a cada docente de las materias que impartirá y a que secciones, en el año en curso. La vista de la asignación de secciones a docentes permite realizar las operaciones de registro, modificaciones y también realizar consultas sobre historial de cambios de asignación realizados y además permite realizar consultas de años anteriores.

oportunidades
FUNDACIÓN GLORIA DE KRIETE

ss11013 Configuración Cerrar Sesión

Alfredo
Rol: Administrador

- Datos Generales
- Personal
- Año en Curso
- Clases
- Secciones
- Asignación Secciones-Alumnos
- Asignación Secciones-Docentes**
- Registro de Notas
- Registro Certificaciones
- Alumnos

Asignación de Secciones a Docentes - año 2020

+ Realizar Asignación

BUSCAR Año

Mostrar 10 registros

Sección	Materia	Docente
A1-CLASS-2020	Computación	Iris Violeta Peraza Linares
A1-CLASS-2020	Emprendimiento	Iris Violeta Peraza Linares
A1-CLASS-2020	Formación Lingüística	Iliana Jeannette Renderos Arrué
A1-CLASS-2020	Inglés	Lorena Beatriz Ramírez Martínez
A1-CLASS-2020	Matemática	Elmer Israel Galdámez Pérez

Figura 4.46 Vista de asignaciones de secciones a docentes.

Asignación de Secciones a Docentes - año 2020

+ Realizar Asignación

Sección: Seleccionar

Materia: Elegir materia

Docente: Elegir Docente

Año: 2020

Descripción:

AGREGAR +

Figura 4.47 Formulario para registrar asignación de sección a docente.

The image shows a web form titled "Modificar" with a red header. It contains three dropdown menus: "Sección" with the value "A1-CLASS-2020", "Materia" with "Computación", and "Docente" with "Iris Violeta Peraza Linares". Below these are two more fields: "Fecha: DD-MM-YYYY" with a calendar icon and "Elegir Periodo" with a dropdown arrow. There is a text area labeled "Observación" and a "GUARDAR" button with a right-pointing arrow.

Figura 4.48 Formulario para editar la asignación de sección a docente.

FUNCIONALIDAD DE ASIGNACIÓN DE SECCION A DOCENTES	
Cargar vista Asignación de secciones a docente	La vista se carga desde el menú de navegación por medio de una petición GET a la ruta <code>/admin/asignacion-docentes</code> la cual es gestionada por el método <code>index()</code> del controlador <code>GrupoMateriaController</code> . El método devuelve la vista <code>asignacion-docente.blade.php</code> y una lista de las asignaciones del año en curso para llenar la tabla de la vista, figura 4.46.
Registrar una nueva asignación	La vista cuenta con un formulario desplegable en la parte superior para realizar una nueva asignación de sección a docente. El botón guardar del formulario realiza una petición Ajax de tipo POST para enviar la información al método <code>store(Request \$request)</code> del controlador <code>GrupoMateriaController</code> el cual ejecuta una sentencia INSERT en la base de datos, utilizando el método <code>save()</code> del modelo <code>GrupoMateria</code> .
Editar asignación de sección a docente	Cada asignación de la lista cuenta con sus botones correspondientes para realizar una modificación en el registro o para consultar el historial de cambios realizados para un registro en específico. Las funciones <code>show(\$id)</code> y <code>update()</code> del controlador <code>GrupoMateriaController</code> son utilizados para cargar el modal de edición de un registro específico y guardar la edición realizada, respectivamente. Adicionalmente, realiza un registro nuevo en la tabla <code>log_cambio_seccion_docente</code> para poder tener un historial de que docente estuvo a cargo de impartir una materia a una sección determinada, en caso de que se hayan dado cambios. El historial de cambios de un grupo de materia puede ser consultado en la vista, por medio de la función <code>obtenerBitacoraCambios()</code> del archivo <code>grupomateria.js</code>
Realizar búsqueda de años anteriores	La vista cuenta con un buscador, en el cual se especifica un año escolar anterior al año en curso. El buscador utiliza la función <code>buscarxanio(\$anio)</code> del controlador <code>GrupoMateriaController()</code> , la cual devuelve la lista de asignaciones del año especificado.

Tabla 4.14 Funcionalidad de Asignación de secciones a docente.



Figura 4.50 MVC de asignación de secciones a docente.

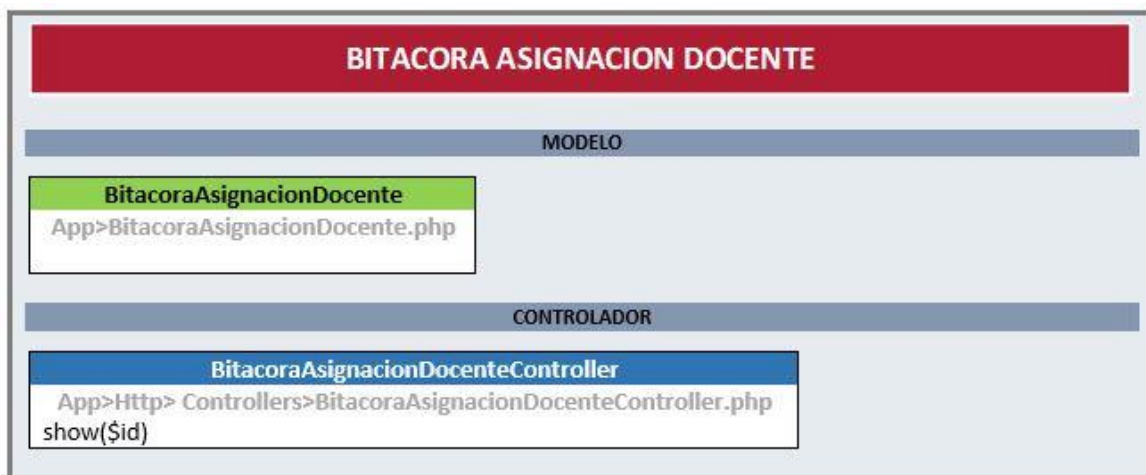


Figura 4.49 MVC de Bitácora de Asignación de secciones a docentes.

Archivo	función	Descripción
AlumnoSeccionController.php	Alumnosxseccion (gmateria)	Recibe como parámetro el Id de una asignación de materia a docente. Devuelve el listado de alumnos pertenecientes a la sección que corresponde a la asignación recibida.
alumnoseccion.js	cargar_secciones (sec)	Recibe un listado de secciones pertenecientes a una clase específica. Construye el bloque HTML correspondiente por cada sección del listado. De esta forma se agrega una columna, con la sección como encabezado a la tabla de la vista asignación-alumnos.
alumnoseccion.js	cargar_alumnos (alum,secciones)	Recibe un listado de alumnos y secciones pertenecientes a una clase específica. Construye el bloque HTML con el cual se crea una fila por cada alumno en la tabla de la lista asignación-alumnos. Cada fila contiene el nombre del alumno y los botones necesarios para realizar la asignación y edición del alumno a una sección.
alumnoseccion.js	cargar_asignados (asignados,secciones)	Recibe un listado de alumnos que ya se encuentran asignados a una sección pertenecientes a una clase específica. Recorre el listado de alumnos asignados para actualizar los botones de asignación e inhabilitarlos, en la vista asignación-alumnos.
alumnoseccion.js	funcionalidad_botones()	<p><i>Tabla 4.15 Métodos y funciones de las clases del submódulo Año en Curso.</i></p> Agrega la funcionalidad de los botones con los atributos <code>data-clase="editar"</code> y <code>data-clase="guardar-edición"</code> definidos. La funcionalidad del botón con el atributo <code>data-clase="editar"</code> consiste en habilitar los campos para poder editar la asignación de un alumno. La funcionalidad del botón con el atributo <code>data-clase="guardar-edición"</code> abre el modal para especificar los datos necesarios y guardar la edición de una asignación de alumno a una sección.
alumnoseccion.js	guardar_asignaciones()	Recorre la tabla de la vista asignación-alumnos y crea una lista con las nuevas asignaciones realizadas. Dicha lista es enviada al controlador ejecutando una petición Ajax dentro de la función. Muestra el mensaje correspondiente a la respuesta recibida en la petición.
alumnoseccion.js	actualizar_asignacion()	Obtiene los valores del formulario del modal de edición de una asignación, y los envía por medio de una petición Ajax al controlador. Muestra el mensaje correspondiente a la respuesta recibida en la petición.
GrupoMateriaController.php	buscarxanio(\$anio)	Recibe como parámetro un año anterior al año en curso. Devuelve la lista de asignaciones de secciones a docentes por materia, del año recibido.
GrupoMateriaController.php	idmateria(\$strgrupo)	Recibe como parámetro el Id de una asignación de materia a docente. Devuelve el id de la materia perteneciente a la asignación recibida.

GrupoMateriaController.php	materia(\$strgrupo)	Recibe como parámetro el Id de una asignación de materia a docente. Devuelve el registro de la materia perteneciente a la asignación recibida.
GrupoMateriaController.php	idclass(\$strgrupo)	Recibe como parámetro el Id de una asignación de materia a docente. Devuelve el id de la clase perteneciente a la asignación recibida.
GrupoMateriaController.php	idseccion(\$strgrupo)	Recibe como parámetro el Id de una asignación de materia a docente. Devuelve el id de la sección perteneciente a la asignación recibida.
GrupoMateriaController.php	gruposxdocente(\$id)	Recibe como parámetro el carnet del docente y devuelve el listado de secciones asignadas por materia a dicho docente en el año en curso.
GrupoMateriaController.php	seccionexdocente(\$id)	Recibe como parámetro el id de un empleado con el cargo de docente. Devuelve el listado de secciones por materia, asignados a dicho docente en el año en curso.
grupomateria.js	obtenerBitacoraCambios ()	Obtiene el id de una asignación de sección a docente por materia y ejecuta una petición Ajax al controlador BitacoraAsignacionDocenteController, con la cual recibe como respuesta el historial de cambios de la asignación y los muestra en pantalla en un modal.
grupomateria.js	busqueda()	Obtiene el año específico del campo de texto de búsqueda en la vista, y por medio de una petición Ajax lo envía a la función buscarxanio() del controlador.

4.2.5 SUBMÓDULO: RECORD DE ALUMNOS

El submódulo Record se encarga de mostrar el rendimiento de los alumnos en el transcurso de los 3 años en cada una de las clases de evaluación, las cuales son las calificaciones FGK del programa, evaluación actitudinal y las calificaciones obtenidas en su centro escolar correspondiente.

```
//RECORD DE ALUMNOS
Route::resource('/admin/record_escolar', 'RecordEscolarController');
Route::resource('/admin/record_fgk', 'RecordFGKController');
Route::resource('/admin/record_actitudinal', 'RecordActitudinalController');
Route::get('/admin/detalle-notas-fgk/{carnet}', 'RecordFGKController@notasPeriodoAlumno');
Route::put('/admin/actualizar-notas-fgk/{carnet}', 'RecordFGKController@actualizar');
```

Figura 4.51 Rutas de submódulo Record de alumnos.

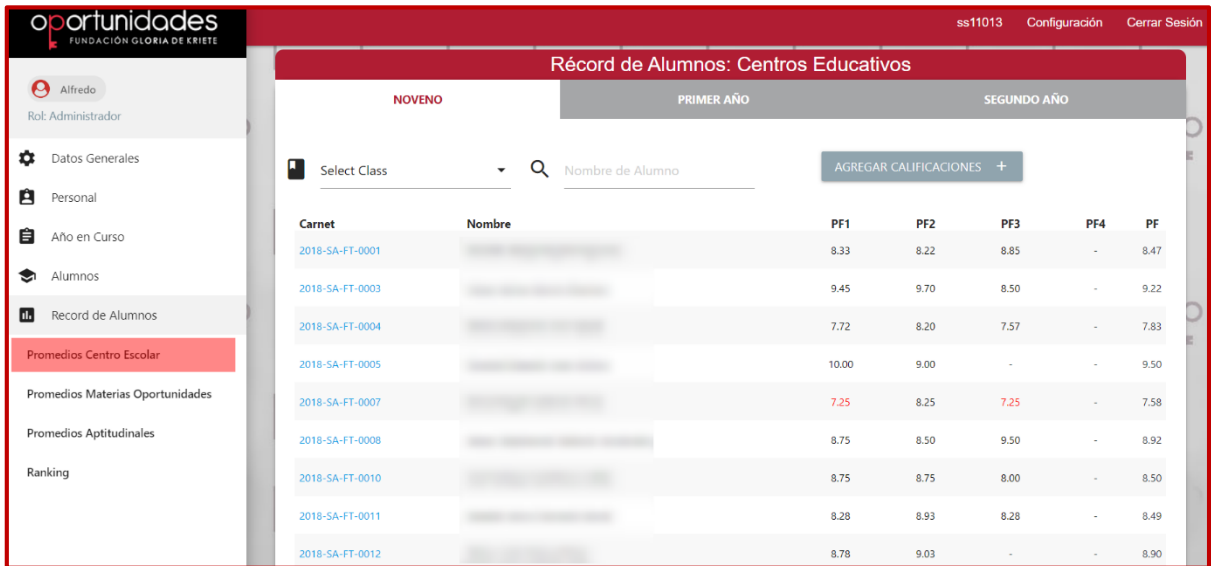
```
<!--RECORD DE ALUMNOS-->
<li>
<div class="collapsible-header waves-effect principal-item">
  class="material-icons"assessment</i>Record de Alumnos</div>
  <div class="collapsible-body">
    <ul>
      @if(Auth::user()->is_admin == 1 && Auth::user()->is_docente == 0)
      <li><a href="/admin/record_escolar" class="waves-effect secondary-item">Promedios Centro Escolar</a></li>
      <li><a href="/admin/record_fgk" class="waves-effect secondary-item">Promedios Materias Oportunidades</a></li>
      <li><a href="/admin/record_actitudinal" class="waves-effect secondary-item">Promedios Aptitudinales</a></li>
      <li><a href="/admin/RankingFGK" class="waves-effect secondary-item">Ranking</a></li>

      @elseif(Auth::user()->is_admin == 1 && Auth::user()->is_docente == 1)
      <li><a href="/admin/record_escolar" class="waves-effect secondary-item">Promedios Centro Escolar</a></li>
      @endif
    </ul>
  </div>
</li>
```

Figura 4.52 Enlaces en el menú del submódulo Record de Alumnos.

4.2.5.1 SECCIÓN PROMEDIOS DE CENTRO ESCOLAR

La sección Promedios de centro escolar muestra una lista completa de todos los alumnos con sus respectivas calificaciones promedio obtenidas por periodo en sus centros escolares, las cuales han sido registradas y organizadas por año escolar por medio de pestañas.



Carnet	Nombre	PF1	PF2	PF3	PF4	PF
2018-SA-FT-0001		8.33	8.22	8.85	-	8.47
2018-SA-FT-0003		9.45	9.70	8.50	-	9.22
2018-SA-FT-0004		7.72	8.20	7.57	-	7.83
2018-SA-FT-0005		10.00	9.00	-	-	9.50
2018-SA-FT-0007		7.25	8.25	7.25	-	7.58
2018-SA-FT-0008		8.75	8.50	9.50	-	8.92
2018-SA-FT-0010		8.75	8.75	8.00	-	8.50
2018-SA-FT-0011		8.28	8.93	8.28	-	8.49
2018-SA-FT-0012		8.78	9.03	-	-	8.90

Figura 4.53 Vista Promedios Centro Escolar.



Figura 4.54 Formulario de ingreso de promedios por alumno.

Promedios de Período					
CARNET:	2018-SA-FT-0003				
NOMBRE:	Evelyn Karina Alarcón Ramirez				
PERIODO	Matemática	Lenguaje y Literatura	Ciencias Naturales	Estudios Sociales	PP
1	9.70	9.50	8.80	9.80	9.45
2	9.50	9.40	9.90	10.00	9.7
3	7.80	7.80	9.30	9.10	8.5
<input type="button" value="EXIT"/>					

Figura 4.55 Detalle de promedios por materias y periodos.

FUNCIONALIDAD PROMEDIOS DE CENTRO ESCOLAR	
Cargar vista promedios Centro Escolar	<p>La vista se carga por medio de una petición GET a la ruta <code>/admin/record_escolar</code> la cual es gestionada por el método <code>index()</code> del controlador <code>RecordEscolarController</code>. El método devuelve la vista <code>record_escolar.blade.php</code> y una lista de alumnos y sus promedios, por cada clase del año en curso, figura 4.53.</p> <p>Cada lista de alumnos es cargada en una tabla correspondiente al año escolar, por medio de la función <code>llenarBaselInicial(lista)</code> del archivo <code>record.js</code>. Las tablas se encuentran organizadas en la vista, utilizando pestañas de navegación, cada una cuenta con un filtro de clases y buscador por nombre de alumno. La funcionalidad de dichos elementos es cargada con el archivo <code>record.js</code>.</p>
Guardar promedios de alumno por periodo	<p>El botón agregar calificaciones, abre una ventana emergente con un formulario para realizar el registro de notas de un alumno específico, figura 4.54. En primer lugar, se realiza la búsqueda del alumno en las listas obtenidas al cargar la vista, registrando el carnet.</p> <p>Al encontrar el alumno se registra el periodo, grado y los promedios del alumno.</p> <p>El botón guardar realiza una petición de tipo POST para enviar la información al método <code>store(Request \$request)</code> del controlador <code>RecordEscolarController</code>, el cual ejecuta una sentencia INSERT en la base de datos, utilizando el método <code>save()</code> del modelo <code>Record</code>.</p>
Ver detalle de promedios por materia	<p>La consulta del detalle de promedios por periodo y materia de un alumno específico, se realiza por medio de la columna carnet del listado de alumnos. Al hacer clic sobre el carnet del alumno, se ejecuta una petición Ajax, la cual es gestionada por el método <code>show(\$id)</code> del controlador <code>RecordEscolarController</code>, en el cual se realiza una consulta a la base de datos para obtener los promedios del alumno en el año escolar correspondiente.</p> <p>Se construye la tabla de promedios por materia y periodo dentro de un modal y es mostrado en pantalla, figura 4.55</p>
Editar promedios por periodo	<p>Para realizar la edición de los registros de promedios, se abre una ventana emergente al hacer clic sobre el promedio final del alumno y periodo a editar, en el listado de alumnos de la vista. Al abrir el modal se obtienen los datos y promedios del alumno por medio de la función <code>edit(\$id)</code> del controlador <code>RecordEscolarController</code>. Al guardar las ediciones realizadas, se ejecuta la función <code>update()</code> del controlador para construir y ejecutar la sentencia UPDATE en la base de datos, con el método <code>update()</code> del modelo <code>Record</code>.</p>

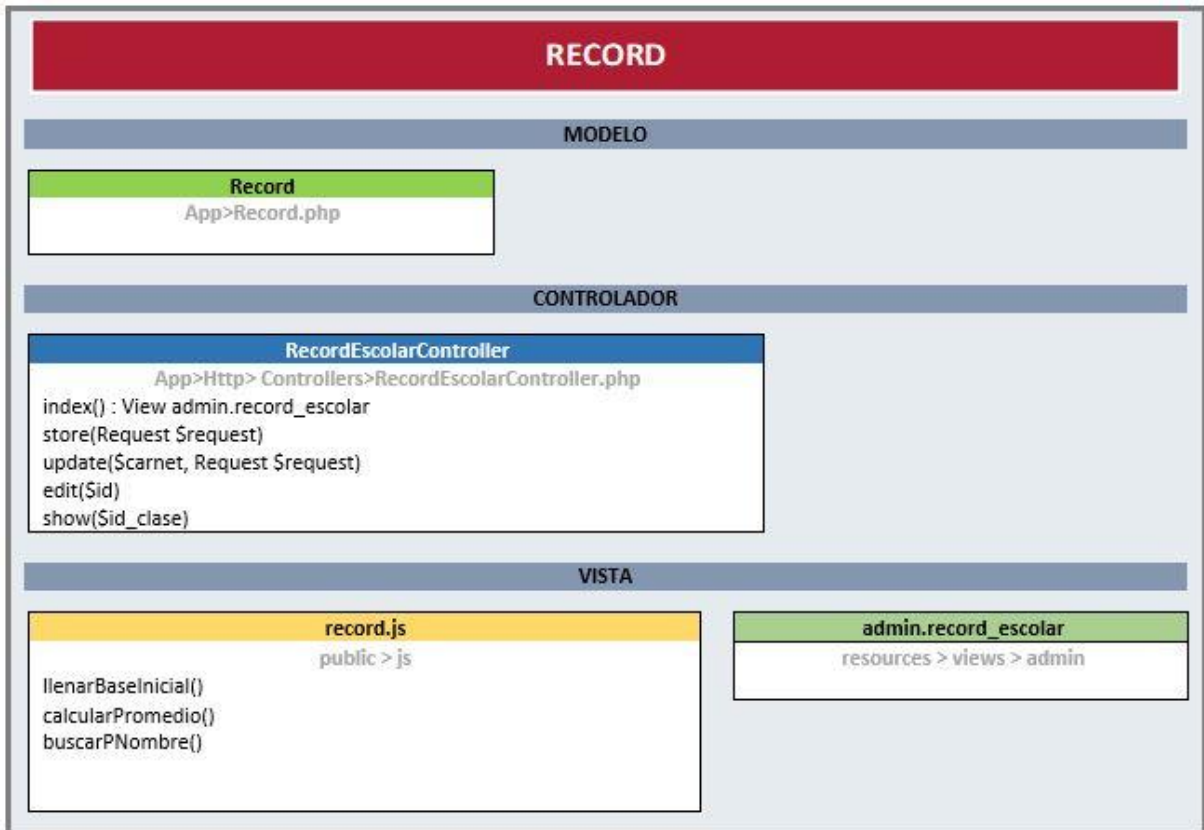


Figura 4.56 MVC de la Sección Record Escolar.

4.2.5.2 SECCIÓN PROMEDIOS DE MATERIAS OPORTUNIDADES

La sección de Promedios de Materias Oportunidades muestra una gran similitud con la sección de Promedios de Centro Escolar ya que cumplen con la misma función, pero en áreas diferentes. Esta sección se encarga de mostrar las calificaciones y promedios que los alumnos han adquirido en las materias que se imparten dentro del programa Oportunidades, durante su estancia en el mismo.

oportunidades FUNDACIÓN GLORIA DE KRIETE

ss11013 Configuración Cerrar Sesión

Récord de Alumnos: Oportunidades

PRIMER AÑO SEGUNDO AÑO TERCER AÑO

Select Class Nombre de Alumno

Carnet	Nombre	PF1	PF2	PF3	PF4	PF
2020-SA-FT-0062	Luis Alexander Villalobos Calderón	7.62	8.54	-	-	8.08
2020-SA-FT-0061	Magaly Astrid Veliz Vásquez	8.76	9.21	-	-	8.98
2020-SA-FT-0059	Leslie Johanna Servellón Vásquez	8.51	8.20	-	-	8.35
2020-SA-FT-0058	Kattie Patricia Servellón Vásquez	8.50	8.31	-	-	8.40
2020-SA-FT-0057	Reina Alejandra Sandoval Reyes	8.95	9.31	-	-	9.13
2020-SA-FT-0056	Melvin Daniel Sandoval Alvarado	8.26	8.51	-	-	8.39
2020-SA-FT-0055	Javier Alexander Sagastume Umaña	8.30	8.88	-	-	8.59

Figura 4.57 Vista de Sección Promedios de Materias Oportunidades.

Notas de Período

CARNET: 2020-SA-FT-0062

NOMBRE:

PERIODO	Inglés	Matemática	Formación Lingüística	Computación	Valores	Orientación Vocacional	PP	Honor Roll
1	7.46	7.70	7.46	7.44	7.50	8.12	7.62	X
2	6.67	9.50	9.59	8.90	8.96	7.65	8.54	X

EXIT

Figura 4.59 Detalle de Promedios por Materias Oportunidades.

Edición de Notas

Carnet: 2020-SA-FT-0062

Nombre:

Clase: CLASS-2022 Período: 1 Año: 2020

Matemática	-	-	-	-	-	-	
	10%	10%	10%	10%	15%	25%	20%
	8.00	8.00	7.50	10.00	1.00	9.28	9.41

Figura 4.58 Edición de notas de Materias Oportunidades.

FUNCIONALIDAD DE RECORD DE MATERIAS OPORTUNIDADES	
Cargar vista de promedios de materias oportunidades	La vista se carga por medio de una petición GET a la ruta <code>/admin/record_fgk</code> la cual es gestionada por el método <code>index()</code> del controlador <code>RecordFGKController</code> . El método devuelve la vista <code>record_fgk.blade.php</code> y una lista de alumnos y sus promedios en las materias del programa oportunidades, por cada grado del año en curso, figura 4.57. Cada lista de alumnos es cargada en una tabla correspondiente al año escolar, por medio de la función <code>llenarBaseInicial(lista)</code> del archivo <code>record_fundacion.js</code> . Las tablas se encuentran organizadas en la vista, utilizando pestañas de navegación; cada una cuenta con un filtro de clases y buscador por nombre de alumno. La funcionalidad de dichos elementos es cargada con el archivo <code>record_fundacion.js</code> .
Consulta del detalle de promedios por materia y periodo de un alumno	La consulta del detalle de promedios por periodo y materia de un alumno específico, se realiza por medio de la columna carnet del listado de alumnos. Al hacer clic sobre el carnet del alumno, se ejecuta una petición Ajax, la cual es gestionada por el método <code>show(\$id)</code> del controlador <code>RecordFGKController</code> , en el cual se realiza una consulta a la base de datos para obtener los promedios por periodo y materia, del alumno en el año escolar correspondiente. Se construye la tabla de promedios por materia y periodo dentro de un modal y es mostrado en pantalla, figura 4.58
Realizar la edición de notas de un alumno	Para realizar la edición de los registros de promedios, se abre una ventana emergente al hacer click sobre el promedio final del alumno, en la columna del periodo a editar; en el listado de alumnos de la vista, figura 4.59. Al abrir el modal se obtienen los datos y promedios del alumno por medio de la función <code>notasPeriodoAlumno(\$id, Request \$req)</code> del controlador <code>RecordFGKController</code> . Se hace clic sobre el botón editar de la materia correspondiente, lo cual habilita los campos de texto para realizar la edición. Al guardar las ediciones realizadas, se ejecuta la función <code>actualizar()</code> del controlador para construir y ejecutar la sentencia UPDATE en la base de datos por cada nota de la materia y periodo especificado, utilizando el método <code>update()</code> del modelo <code>RecordFGK</code> .

Tabla 4.17 Funcionalidad del Record de materias oportunidades.

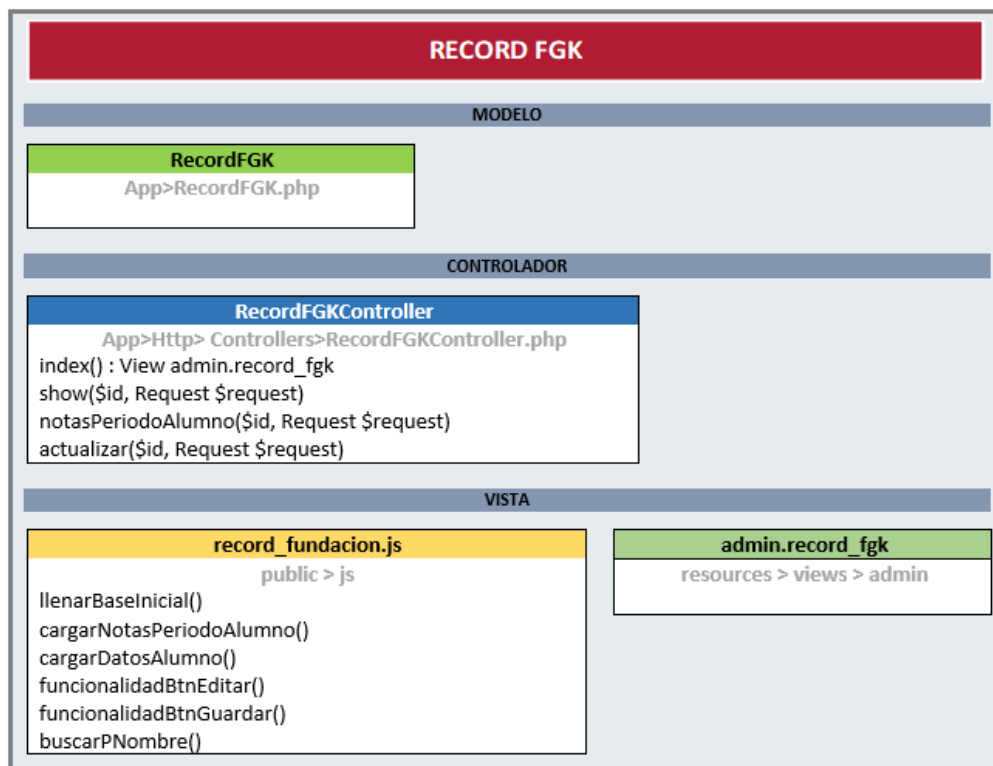


Figura 4.60 MVC de Promedios de Materias Oportunidades.

4.2.5.3 SECCION PROMEDIOS ACTITUDINALES

El programa Oportunidades además de registrar tanto calificaciones de materias de centro escolares como las de su propio programa también se preocupan por el lado actitudinal de los alumnos como la conducta o el interés que ellos tienen en las materias del programa. La sección de Record Actitudinal se encarga de mostrar al administrador todas las calificaciones actitudinales con las que los docentes han evaluado a los alumnos, dependiendo de cómo estos se desenvuelven en las materias que están cursando.

Carnet	Nombre	PF1	PF2	PF3	PF4	PF
2020-SA-FT-0062	Luis Alexander Villalobos Calderón	4.50	-	-	-	4.50
2020-SA-FT-0061	Magaly Astrid Veliz Vásquez	4.83	-	-	-	4.83
2020-SA-FT-0059	Leslie Johanna Servellón Vásquez	4.67	-	-	-	4.67
2020-SA-FT-0058	Kattie Patricia Servellón Vásquez	4.83	-	-	-	4.83
2020-SA-FT-0057	Reina Alejandra Sandoval Reyes	4.80	-	-	-	4.80
2020-SA-FT-0056	Melvin Daniel Sandoval Alvarado	4.95	-	-	-	4.95
2020-SA-FT-0055	Javier Alexander Sagastume Umaña	4.75	-	-	-	4.75

Figura 4.62 Vista de promedios actitudinales.

PERIODO	Perseverancia	Optimismo	Trabajo en la escuela	Interpersonal	Gratitud	Inteligencia social	Curiosidad	Ánimo
1	4.60	4.13	4.25	5.00	5.00	4.67	4.17	4.00

Figura 4.61 Detalle de promedios actitudinales por periodo.

FUNCIONALIDAD DE PROMEDIOS ACTITUDINALES	
Cargar vista de promedios actitudinales	<p>La vista se carga por medio de una petición GET a la ruta <code>/admin/record_actitudinal</code> la cual es gestionada por el método <code>index()</code> del controlador <code>RecordActitudinalController</code>. El método devuelve la vista <code>record_actitudinales.blade.php</code> y una lista de alumnos y los promedios finales por periodo de sus evaluaciones actitudinales. Se devuelve una lista por cada grado del año en curso, figura 4.61.</p> <p>Cada lista de alumnos es cargada en una tabla correspondiente al año escolar, por medio de la función <code>llenarBaseInicial(lista)</code> del archivo <code>record_fundacion.js</code>. Las tablas se encuentran organizadas en la vista, utilizando pestañas de navegación; cada una cuenta con un filtro de clases y buscador por nombre de alumno. La funcionalidad de dichos elementos es cargada con el archivo <code>record_actitudes.js</code>.</p>
Consultar detalle de promedios actitudinales por periodo de un alumno	<p>La consulta del detalle de promedios por periodo y criterio actitudinal de un alumno específico, se realiza por medio de la columna carnet del listado de alumnos. Al hacer clic sobre el carnet del alumno, se ejecuta una petición Ajax, la cual es gestionada por el método <code>show(\$id)</code> del controlador <code>RecordActitudinalController</code>, en el cual se realiza una consulta a la base de datos para obtener los promedios por periodo y criterio actitudinal del alumno en el año escolar correspondiente.</p> <p>Se construye la tabla de promedios finales por criterio actitudinal y periodo dentro de un modal y es mostrado en pantalla, figura 4.62</p>

Tabla 4.18 Funcionalidad de Record de promedios actitudinales.

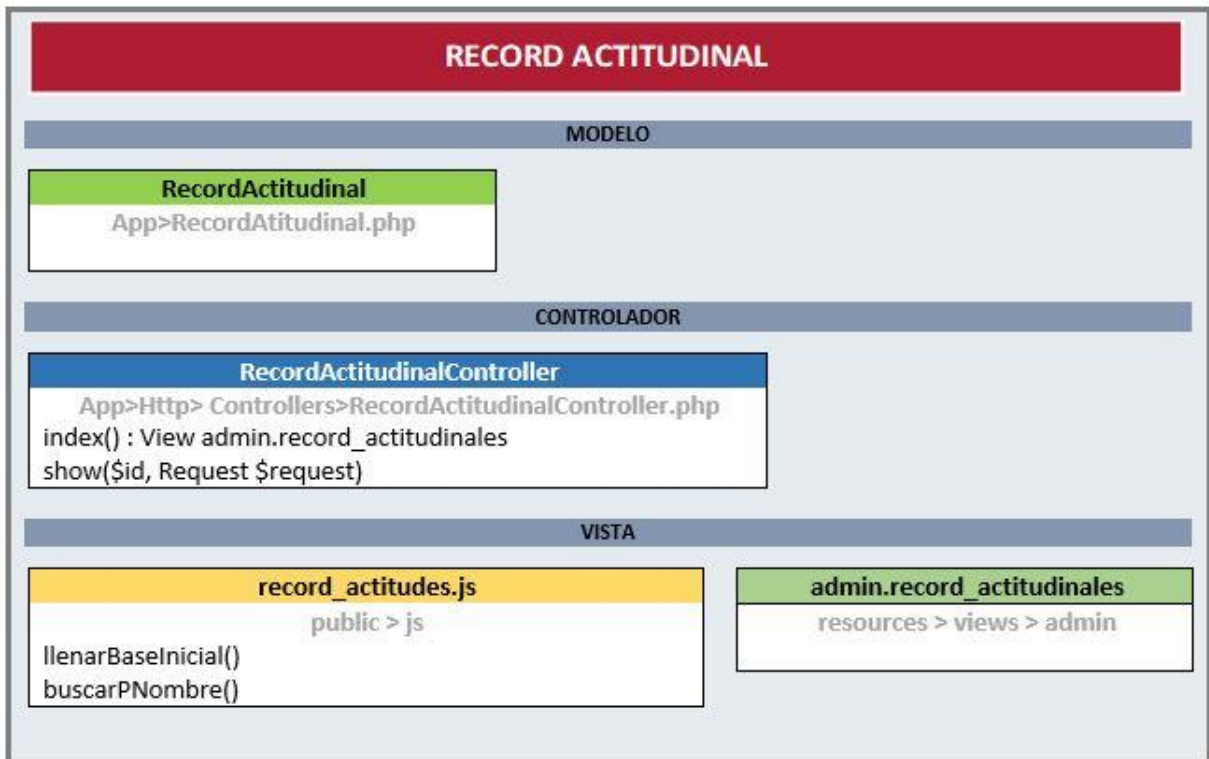


Figura 4.63 MVC de Record de Promedios Actitudinales.

4.2.5.4 SECCION RANKING

En la sección de Ranking se recopila todos los promedios obtenidos de los centros escolares y los promedios del programa para tener un promedio global calculado y la diferencia entre ellos con la idea de mostrar una diferencia del desempeño del alumno en ambos lugares.

Carnet	Nombre	CCGK	Mined	CCGK&Mined	Diferencia
2018-SA-FT-0030		9.49	9.21	9.349	0.286
2018-SA-FT-0070		9.40	9.41	9.401	-0.010
2018-SA-FT-0069		9.39	9.58	9.485	-0.186
2018-SA-FT-0054		9.38	8.89	9.135	0.494
2018-SA-FT-0033		9.38	9.43	9.406	-0.044
2018-SA-FT-0064		9.38	8.33	8.856	1.051
2018-SA-FT-0031		9.37	8.49	8.930	0.881
2018-SA-FT-0041		9.34	8.67	9.004	0.671
2018-SA-FT-0023		9.31	9.36	9.337	-0.046
2018-SA-FT-0058		9.29	7.48	8.386	1.811

Figura 4.64 Vista del Ranking de promedios globales.

FUNCIONALIDAD DE SECCION RANKING	
Cargar vista Ranking	<ul style="list-style-type: none"> • La vista se carga por medio de una petición GET a la ruta <code>/admin/ranking/{\$clase}</code> la cual es gestionada por el método <code>RankingFGK(\$clase)</code> del controlador <code>ExpedienteController</code>. • El método devuelve la vista <code>RankingFGK.blade.php</code>. • En el método se realiza una consulta a la base de datos con la cual se obtienen los promedios globales del programa Oportunidades y del centro escolar por cada alumno y la diferencia entre ambos promedios. El resultado de la consulta es enviado y mostrado en el Datatable de la vista <code>RankingFGK.blade.php</code>, Figura 4.64.

Tabla 4.19 Funcionalidad de Sección Ranking.

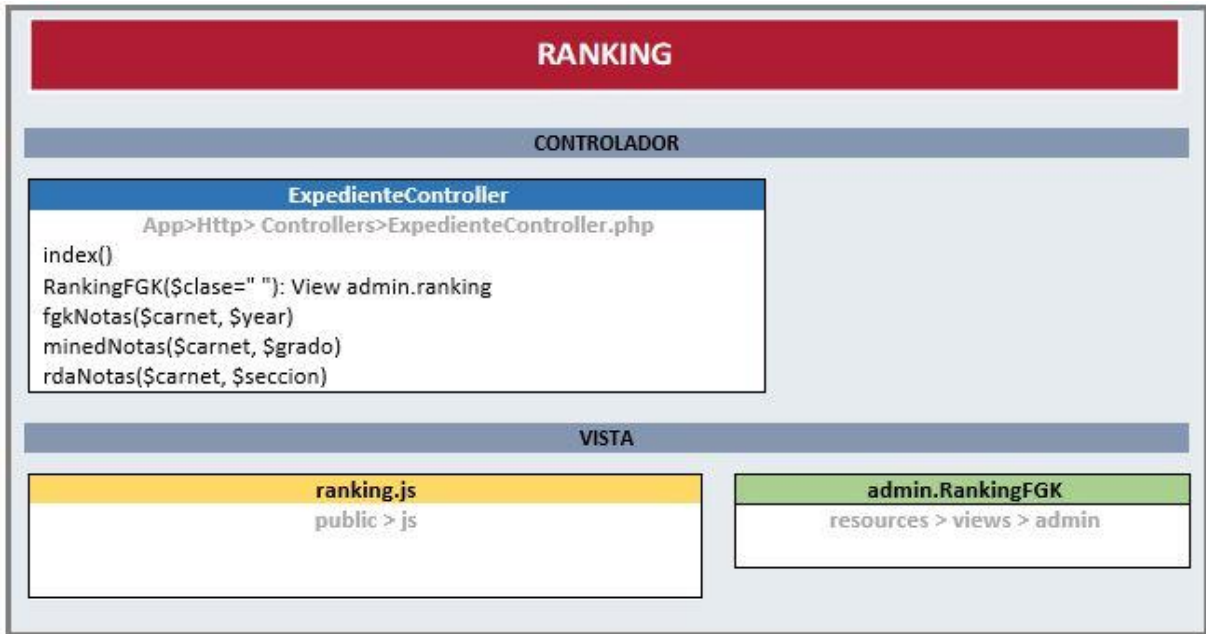


Figura 4.65 MVC del Ranking de promedios.

Archivo	función	Descripción
record.js	llenarBaseInicial()	Recibe como parámetro un listado de alumnos con sus promedios finales por periodo, de un año escolar específico. Recorre el listado de alumnos y construye el bloque HTML correspondiente para mostrar una fila con el nombre de cada alumno y las columnas correspondientes al promedio final de cada periodo.
record.js	calcularPromedios()	Recibe como parámetros 4 notas y calcula su promedio.
record.js	buscarPNombre()	Recibe como parámetros una cadena de texto y una tabla con el listado de alumnos. Recorre la tabla y realiza una búsqueda de coincidencias de la columna del nombre del alumno con la cadena de texto recibida como parámetro. Actualiza la tabla con las coincidencias de alumnos encontradas.
record_fundacion.js	cargarNotasPeriodoAlumno()	Recibe un listado de todas las calificaciones por materia de un alumno en un periodo específico. Por cada materia crea el bloque HTML correspondiente para mostrar las evaluaciones y las notas obtenidas en cada una. Además, agrega un botón de Edición y Guardar edición por cada materia.
record_fundacion.js	CargarDatosAlumno()	Recibe como parámetro los datos generales de un alumno específico para mostrarlos en el modal de edición de notas de la vista record_fgk.

record_fundacion.js	funcionalidadBtnEditar()	Define el callback de la función click de los botones con la clase btnEditar definida: Abrir el modal de edición.
record_fundacion.js	funcionalidadBtnGuardar()	Define el callback de la función click de los botones con la clase btnGuardar definida: Enviar los datos de edición al controlador.

Tabla 4.20 Métodos y funciones de las clases del submódulo Record de alumnos.

4.3 MODULO DOCENTE

El módulo Docente tiene como finalidad principal el permitir a los docentes del programa Oportunidades registrar las calificaciones de los alumnos pertenecientes a las secciones que le han sido asignadas.



Figura 4.66 Vista de Home del módulo docente.

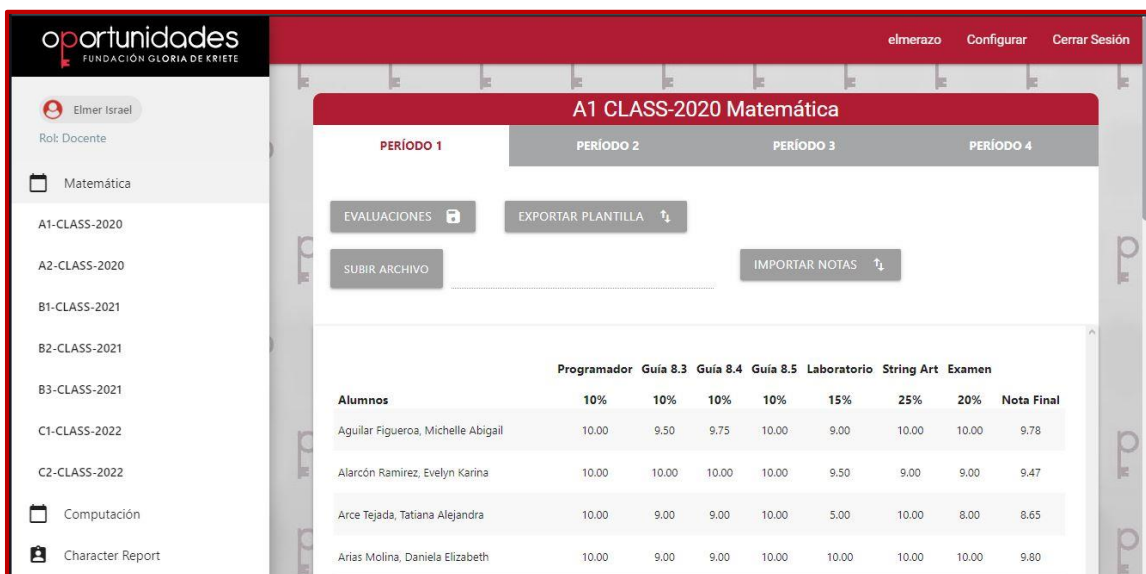


Figura 4.67 Vista del módulo docente con notas por sección, materia y periodo.



Figura 4.69 Ventana para el registro y edición de Evaluaciones por periodo.

RDA C1-CLASS-2022-MAT

PERÍODO 1 PERÍODO 2 PERÍODO 3 PERÍODO 4

EVALUACIONES EXPORTAR PLANTILLA

SUBIR ARCHIVO IMPORTAR NOTAS

Alumnos	Termina lo que empieza	Proyectos a largo plazo	Trata fuerte aún fallando	Comprometido con metas	Se mantiene trabajando	Perseverancia
Argueta Martinez, Erick Abid	5.00	5.00	4.00	4.00	5.00	4.60
Cardona Méndez, Carlos Rafael	5.00	4.00	4.00	4.00	4.00	4.20
Cruz Algarin, Katherine Jeannette	5.00	5.00	4.00	4.00	5.00	4.60
Díaz Guerreiro, Meybelyn Darleny	5.00	5.00	4.00	4.00	5.00	4.60
Gálvez Valencia, María Isabel	5.00	5.00	4.00	4.00	5.00	4.60

Figura 4.68 Notas actitudinales cargadas en pantalla.

FUNCIONALIDAD DE REGISTRO DE NOTAS	
Cargar vista de Docente	<p>Cuando un usuario se loguea y es identificado como docente el controlador HomeController se encarga de devolver como respuesta el layout docente.blade.php.</p> <p>Además del layout se envían las Secciones por materia que han sido asignadas al docente por cada clase del año en curso, las cuales son cargadas en el menú de navegación lateral con el método cargarPanel() de docente.js.</p> <p>La vista cuenta con el panel lateral de navegación, y la sección para cargar las notas de los alumnos la cual esta organizadas en pestañas. Una pestaña por cada periodo. E cada pestaña se cargan las notas del periodo correspondiente. Figura 4.66.</p>
Cargar notas por Sección y periodo	<p>Al seleccionar una sección del panel lateral, se ejecuta la función obtenerAlumnosNotas() y obtenerEvaluaciones() los cuales realizan una petición a los controladores NotasFgkController y EvaluacionesController. Los controladores devuelven el listado de alumnos correspondiente a la sección, con las notas de las evaluaciones del periodo que se encuentre activo en las pestañas de periodos. Los datos son cargados en pantalla en una tabla que muestra las notas de todos los alumnos por evaluación. Figura 4.67.</p>
Ver Evaluaciones del periodo	<p>La vista cuenta con el botón Evaluaciones. Al hacer clic al botón se abre la ventana emergente en la cual se carga un formulario para registrar evaluaciones y una tabla en la que se cargan todas las evaluaciones correspondientes a la clase de la sección seleccionada y el periodo activo. Figura 4.68.</p>
Registrar nueva evaluación	<p>Para el registro de las evaluaciones por periodo se debe seleccionar la materia y sección correspondiente a la clase de la cual se registrarán las evaluaciones. De igual forma se selecciona la pestaña del periodo correspondiente. Se abre la ventana de evaluaciones y se realiza el registro de la nueva evaluación especificando el nombre, nombre corto y porcentaje. Al guardar la evaluación se realiza el registro en la base de datos por medio del método store() del controlador EvaluacionesController.</p>
Editar Evaluación	<p>Para realizar la edición de una evaluación se hace clic sobre el botón de edición que corresponde a la evaluación, lo cual habilita los campos de edición y el botón para guardar los cambios. Una vez modificados los datos se confirma la edición con el botón guardar, con el cual se realiza la actualización en la base de datos por medio del método update() del controlador EvaluacionesController.</p>
Exportar plantilla de notas por periodo en archivo Excel	<p>Al hacer clic sobre el botón de exportar plantilla, se ejecuta una llamada al método exportarNotasAlumnosxPeriodo() del controlador NotasFgkController, el cual genera un archivo de Excel en el cual se carga la lista de alumnos de la sección y materia seleccionada en el menú y del periodo activo en las pestañas. Dentro de la cuadrícula del archivo se cargan todas las notas de los alumnos que han sido registradas hasta el momento.</p>
Importar notas desde archivo de Excel	<p>Para subir las notas registradas en la plantilla de Excel, se debe seleccionar la sección y periodo correspondiente en el panel y pestañas de navegación. Se hace clic al botón subir archivo para seleccionar el archivo Excel con la plantilla de notas. Al hacer clic en el botón importar notas. El archivo es enviado a la función importarNotasAlumnosxPeriodo(), el cual recorre el archivo y guarda en la base de datos, las notas del archivo que aún no es encuentran registradas.</p>

Tabla 4.21 Funcionalidades del módulo docente.

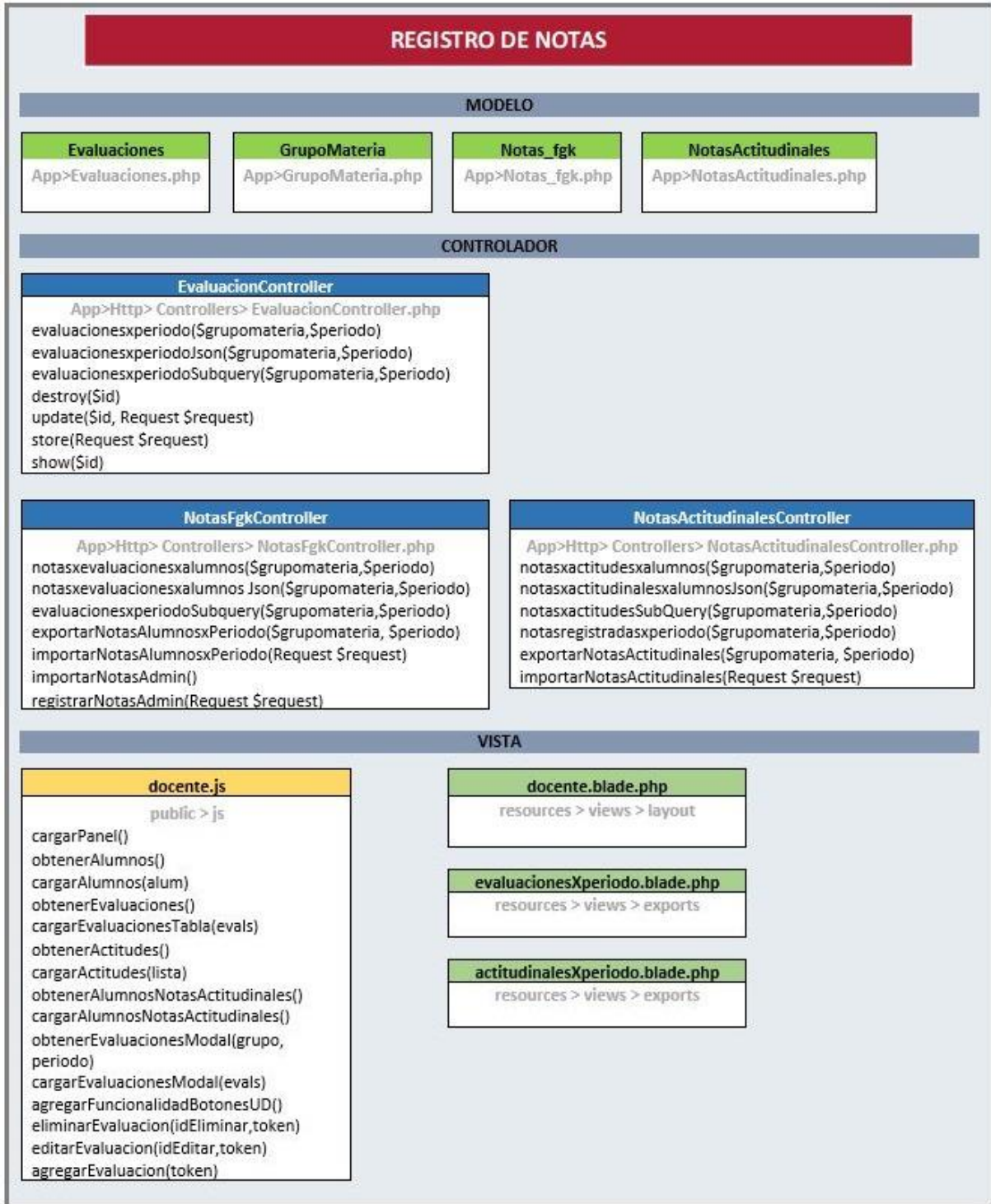


Figura 4.70 MVC Registro de Notas.

EXPORTACION E IMPORTACION DE CALIFICACIONES

CLASES

NotasFgkExport

```
App>NotasFgkExport.php
protected $grupoMat;
protected $periodo;
protected $listadoAlumnos;
protected $evaluacionesPeriodo;
protected $alumnos;
protected $evaluaciones;
protected $alumnosEvalsNotas;

__construct(string $gmat, string $per)
view(): View
registerEvents(): array
formulanotafinal(AfterSheet $event)
tablaBaseDatos(AfterSheet $event)
cargarNotas(AfterSheet $event)
```

NotasActitudinalesExport

```
App>NotasActitudinalesExport.php
protected $grupoMat;
protected $periodo;
protected $listadoAlumnos;
protected $actitudes;
protected $criterios;
protected $alumnos;
protected $actitudes;
protected $alumnosActitudesNotas;

__construct(string $gmat, string $per)
view(): View
registerEvents(): array
formulanotafinal(AfterSheet $event)
tablaBaseDatos(AfterSheet $event)
```

NotasFgkImport

```
App>NotasFgkImport.php
protected $grupoMat;
protected $periodo;
protected $listadoAlumnos;
protected $evaluacionesPeriodo;
protected $alumnos;
protected $evaluaciones;
protected $notas;

__construct(string $gmat, string $per)
collection(Collection $rows)
registerEvents(): array
```

NotasActitudinalesImport

```
App>NotasActitudinalesImport.php
protected $grupoMat;
protected $periodo;
protected $notas;
protected $listadoAlumnos;
protected $actitudes;
protected $alumnos;
protected $actitudes;

__construct(string $gmat, string $per)
collection(Collection $rows)
registerEvents(): array
```

Figura 4.71 Exportación e importación de Notas en Excel.

4.4 MODULO EXPEDIENTE DEL ALUMNO

El módulo del expediente académico del alumno, es el módulo por medio del cual el alumno puede ingresar al sistema y realizar la consulta del detalle de todas sus calificaciones registradas hasta la fecha, las notas FGK y actitudinales del programa oportunidades; así como también las notas obtenidas en el centro escolar correspondiente al alumno.

El alumno puede ver en detalle las notas del año en curso, las cuales se mostrarán primero, y de igual forma puede consultar las notas de los años anteriores cursados del programa. Las calificaciones se muestran de forma organizada por materia y periodo cada año.

2018-SA-FT-0001 Configurar Cerrar Sesión

oportunidades
FUNDACIÓN GLORIA DE KRIETE

Michelle Abigail

PRIMER AÑO

SEGUNDO AÑO

TERCER AÑO

FGK

RDA

MINED

Notas FGK - 2020

	PERIODO 1	PERIODO 2	PERIODO 3	PERIODO 4			
Estado: ---					Promedio: 9.10		
Matemática	-	-	-	-	PF		
	-	-	-	-	9.77		
	10%	15%	25%	20%	10%	10%	10%
	10.00	9.00	10.00	10.00	10.00	9.50	9.75
Formación Lingüística	-	-	-	-	PF		
	-	-	-	-	9.83		
	10%	10%	20%	20%	20%	20%	
	10.00	10.00	9.66	10.00	9.50	10.00	
Computación	-	-	-	-	PF		
	-	-	-	-	9.32		
	20%	20%	20%	30%	10%		
	9.50	10.00	8.50	9.06	10.00		

Figura 4.72 Vista de Notas FGK del expediente del alumno.

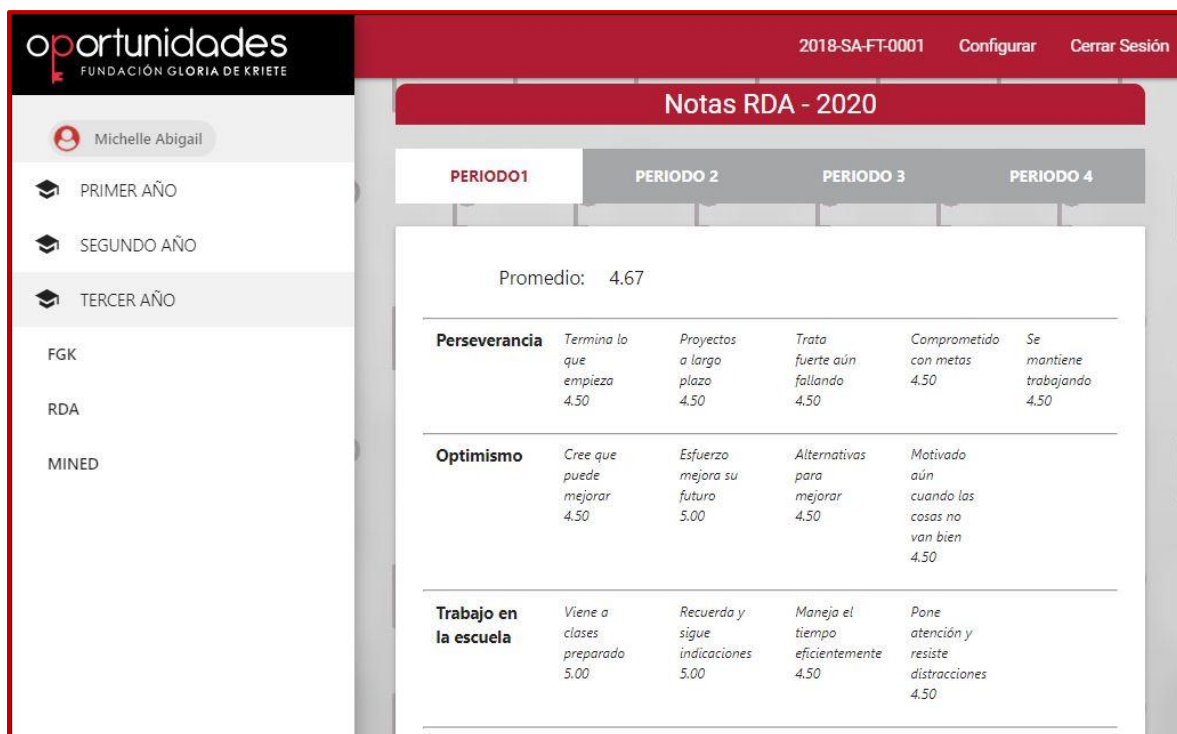


Figura 4.74 Vista de notas actitudinales del expediente del alumno.

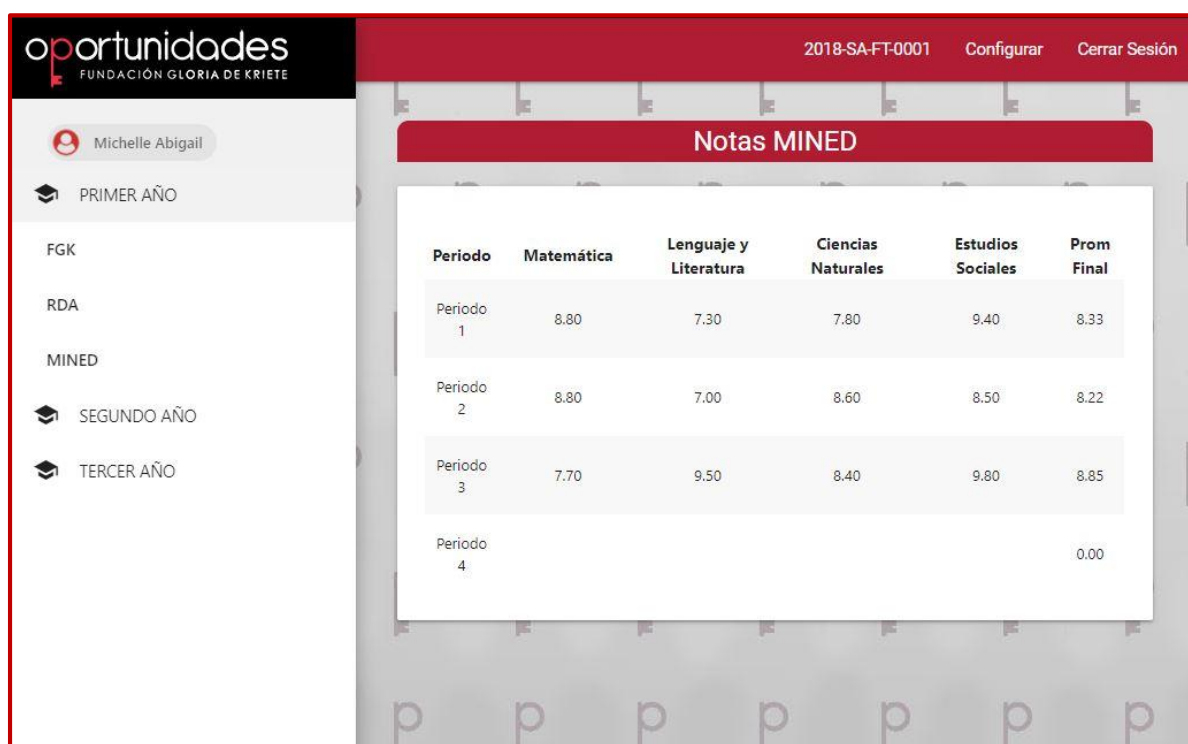


Figura 4.73 Vista de notas MINED del expediente del alumno.

FUNCIONALIDAD DEL EXPEDIENTE ACADMICO DEL ALUMNO	
Cargar vista de inicio del expediente del alumno	Cuando un usuario inicia sesión y es identificado como un alumno el controlador HomeController se encarga de devolver como respuesta el layout user.blade.php . La vista utiliza los métodos cargarMenu() e inicio() del archivo expediente.js para mostrar en pantalla las notas FGK del año en curso.
Cargar las notas FGK	Cuando el usuario selecciona las notas FGK, se determina de que año ha realizado la selección por medio del identificador del botón y se ejecuta la función cargarNotasFGK(id, anho) del archivo expediente.js la cual realiza la petición al controlador ExpedienteController para ejecutar el método fgkNotas(carnet, year) , el cual realiza la consulta de las notas FGK del año especificado y se cargan en pantalla.
Cargar las notas Actitudinales	Cuando el usuario selecciona las notas actitudinales, se determina de que año ha realizado la selección por medio del identificador del botón y se ejecuta la función cargarNotasRDA(id, seccion, anho) del archivo expediente.js la cual realiza la petición al controlador ExpedienteController para ejecutar el método rdaNotas(carnet, seccion) , el cual realiza la consulta de las notas actitudinales del año especificado y se cargan en pantalla.
Cargar las notas del MINED	Cuando el usuario selecciona las notas FGK, se determina de que año ha realizado la selección por medio del identificador del botón y se ejecuta la función cargarNotasMined(id, grado) del archivo expediente.js la cual realiza la petición al controlador ExpedienteController para ejecutar el método minedNotas(carnet, grado) , el cual realiza la consulta de las notas MINED del año especificado y se cargan en pantalla.

Tabla 4.22 Funcionalidades del Expediente académico del alumno.

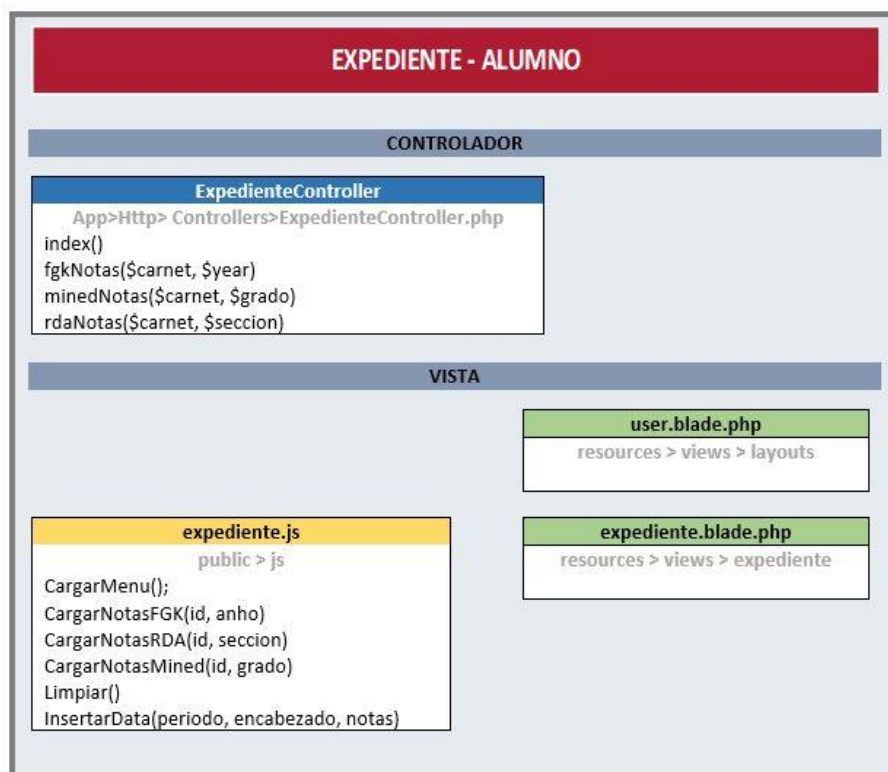


Figura 4.75 MVC del módulo del expediente del alumno.

4.5 MANUAL DE USUARIO

En la sección anterior del presente capítulo se ha detallado de forma generalizada, la estructura que conforma el expediente en línea EELO y de qué forma las clases del sistema interactúan entre sí obedeciendo el patrón MVC. Como ANEXO VIII se adjunta el Manual de Usuarios del Sistema EELO, el cual está dirigido a detallar por pasos todas las tareas que puede llevar a cabo cada uno de los roles de usuario definidos para el sistema.

4.6 ESTRUCTURA ORGANIZACIONAL PARA IMPLEMENTACIÓN.

- **ORGANIZACIÓN POR PROYECTO**

La Fundación Gloria de Kriete sede Santa Ana, opera varios programas correspondientes al área de educación, de los cuales cada uno cuenta con una estructura organizativa y administrativa definida.

El programa Oportunidades como un sistema cerrado tiene una estructura jerárquica básica que, en términos de autoridad lineal, está formada por dos niveles. El director ejerce la función de administrar, organizar y delegar funciones y tareas a los docentes y asistente administrativo.

Tal organización estructura toda la actividad y funcionamiento del programa, otorgando la administración de las herramientas y recursos al director del programa. El personal docente y asistente administrativo ejerce las funciones y tareas que son delegadas por el director, ya sea para el funcionamiento normal del programa, así como también para ejecutar nuevos proyectos y funciones dentro del mismo.

Para el presente proyecto de desarrollo e implementación del Sistema EELO, el equipo de desarrollo, se presenta como una unidad de Staff dentro de la organización definida.

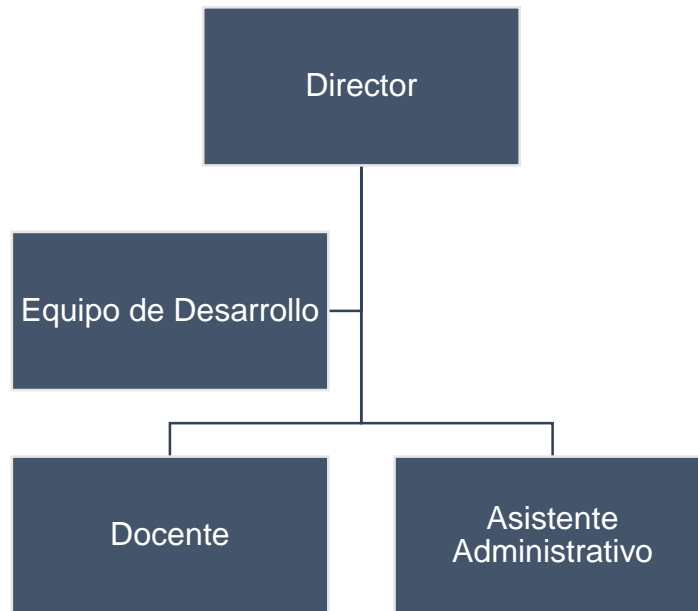


Figura 4.76 Organización estructural del programa Oportunidades.

El equipo de desarrollo funciona como una unidad autogestionada, la cual lleva a cabo la planeación (análisis y diseño del sistema), así como también la ejecución y control de la implementación. Todos los miembros del equipo tienen el mismo nivel de participación en las actividades y tareas de la implementación del sistema.

- Personal del programa Oportunidades designado para la implementación del sistema EELO:

Nombre	Doris Granados.
Cargo	Asistente Administrativo.
Rol del sistema	Asistente administrativo.
Categoría profesional	
Responsabilidad	Guardar toda la información necesaria de personal y alumnos a la base de datos. Llevar a cabo todas las tareas de control y organización delegadas por el director.

Nombre	Elmer Israel Galdámez.
Cargo	Docente
Rol del sistema	Docente
Categoría profesional	Educación superior, Técnico.
Responsabilidad	Registro de la información de las evaluaciones y sus resultados de sus materias a cargo.

Tabla 4.23 Personal designado del programa para la implementación del sistema.

4.7 PLAN DE IMPLEMENTACIÓN

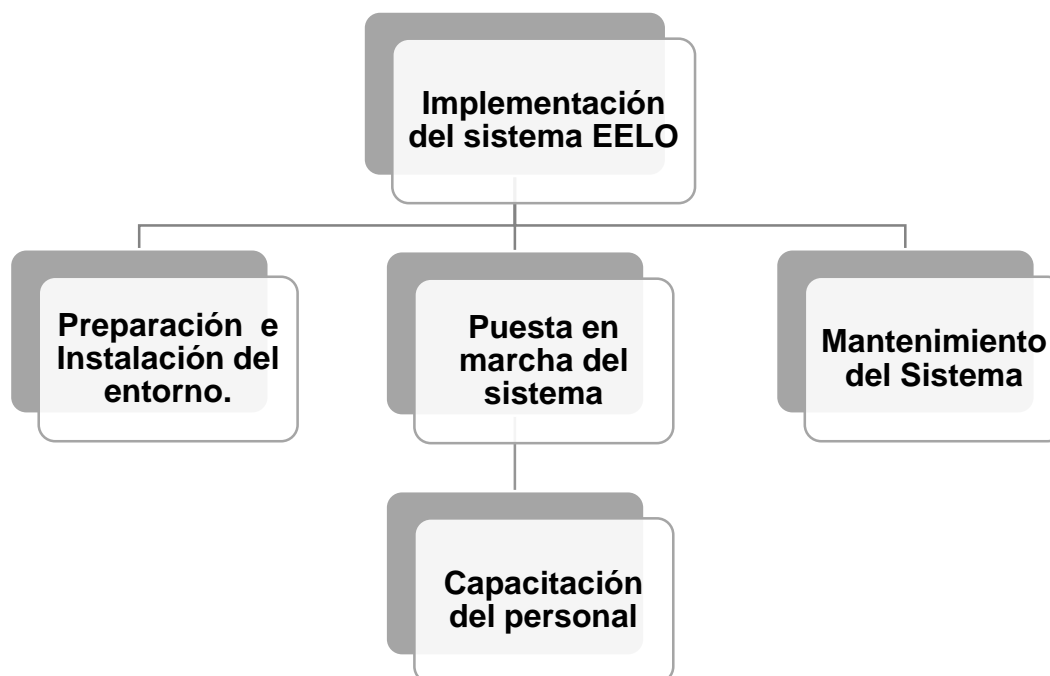


Figura 4.77 Fases de implementación del sistema EELO.

Para llevar a cabo la implementación del sistema se han considerado una serie de fases, dentro de las cuales se organizan las actividades que se deben ser ejecutadas. Más adelante se presentan las actividades clave de cada una de las fases de implementación, y cada una de ellas se encuentra dividida en tareas. Además, se presenta el tiempo estimado para llevar a cabo cada una de las actividades.

4.7.1 ENTORNO DE DESARROLLO Y PRODUCCIÓN

- En la primera fase de implementación se tiene como objetivo definir y configurar un entorno en el cual se ejecute el desarrollo del código fuente del sistema; así como también el entorno de producción, utilizando los medios y tecnologías que permitan realizar el despliegue de la aplicación web del expediente académico y este a disposición de sus usuarios.
- Se ha considerado el proceso de implementación a partir de la preparación del entorno de desarrollo y la codificación del sistema, ya que dichas tareas implican la ejecución y puesta en marcha de los recursos creados y considerados en el análisis y diseño realizado hasta el momento.
- Las herramientas, tecnologías y recursos fueron definidos al determinar la factibilidad del proyecto y en la fase de diseño del sistema. Por lo tanto, la preparación de los entornos de desarrollo y producción consiste en la configuración y utilización de dichas herramientas.
- La preparación de los entornos de desarrollo y producción consiste en una configuración sencilla que permite la modificación y depuración del código fuente de forma eficiente con los recursos y tecnologías delimitados mayormente por el plan de hosting contratado.

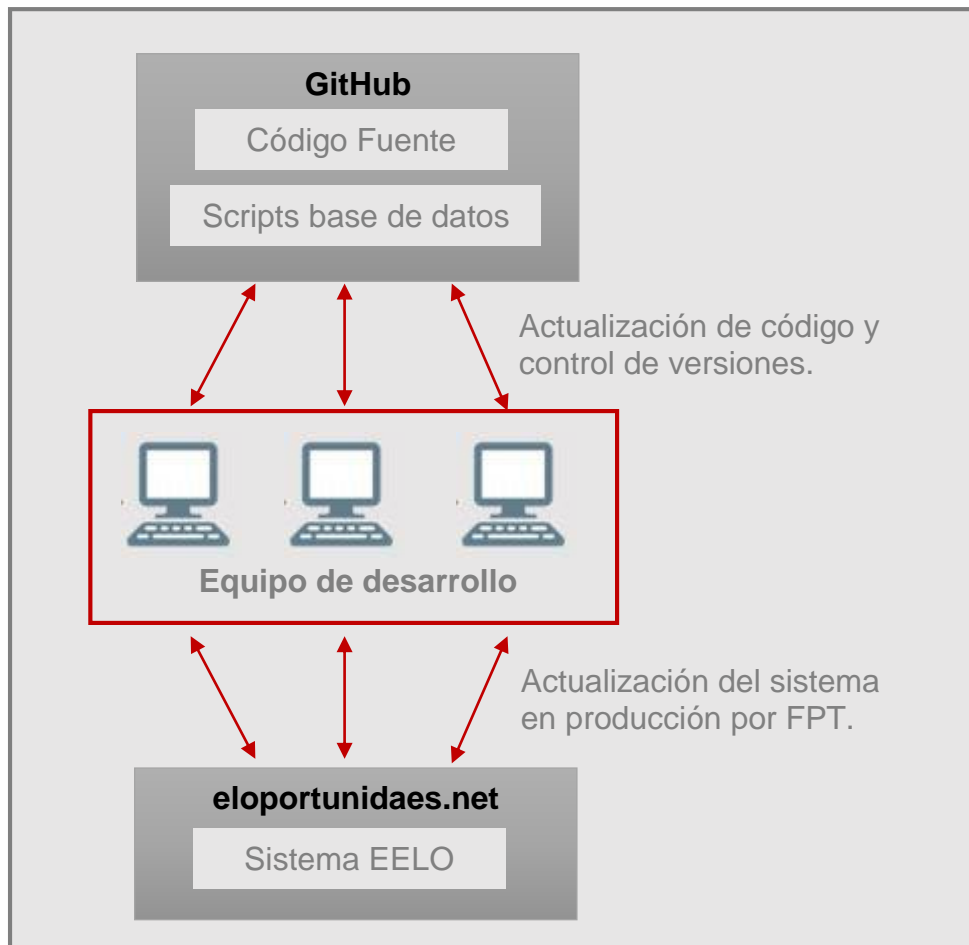


Figura 4.78 Entorno de desarrollo y producción.

4.7.2 MÉTODO DE IMPLEMENTACIÓN

- **MÉTODO DE IMPLEMENTACIÓN EN PARALELO.**

Para la puesta en marcha del Sistema EELO, se considera que la naturaleza del sistema actual facilita ejecución de las tareas del programa utilizando ambos sistemas de forma paralela.

Al iniciar la implementación del Sistema EELO es necesario realizar el ingreso de todos los datos generales y los expedientes de todos los alumnos; por lo tanto, mientras se desarrolla dicha actividad, el director y docentes seguirán utilizando los archivos Excel de registro de notas y los colectores de promedios que se han utilizado hasta el momento.

Una vez todos los datos generales y expedientes hayan sido registrados, de igual forma se podrá trabajar de forma paralela con los archivos de Excel actuales y el Sistema EELO ya que los registros de notas se podrán copiar fácilmente a los archivos generados por el Sistema EELO para que sean guardados en la nueva base de datos. Esto permitirá a su vez comparar y validar que el ranking de los alumnos se está calculando de forma correcta.

4.7.3 ACTIVIDADES ESTRATEGICAS PARA LA IMPLEMENTACIÓN.

En la tabla 4.24 se muestran de forma de tallada todas las actividades clave que son necesarias para la implementación del sistema. Para estimar la duración de la implementación del sistema se ha utilizado el método de la ruta crítica, figura 4.79; para el cual, también se ha realizado la distribución de las vistas y módulos a desarrollar entre los miembros del equipo de desarrollo en la tabla 4.25, para poder determinar las actividades que se pueden ejecutar de forma simultánea con otras.

Además, se presentan todas las actividades de implementación y demás etapas del proyecto completo en el cronograma del ANEXO X, representando la duración de cada una a lo largo del periodo de ejecución del proyecto del año 2018 al 2019.

	ACTIVIDADES Y TAREAS		DURACIÓN
Fase 1. Instalación y preparación del entorno.	act-01	Preparación de entorno de desarrollo	1
	tar-0101	Crear un repositorio privado en la plataforma GitHub para llevar a cabo el desarrollo del código fuente de la aplicación web de forma remota.	
	tar-0202	Configurar el editor de código VS Code en la computadora de cada uno de los miembros del equipo de trabajo para conectarse al repositorio del proyecto.	
	act-02	Preparar entorno de producción	1
	tar-0201	Contratar plan de servicio de hosting con proveedor SiteGround.	
	tar-0202	Instalación del código fuente base de la de aplicación con el framework Laravel	
	tar-0203	Creación de base de datos en MySQL y configurar la conexión del sistema con la base de datos.	
tar-0204	Configuración de cuentas FTP y Certificado SSL para realizar las transferencias de datos entre el servidor y los usuarios y para la transferencia de archivos entre el entorno de desarrollo y el de producción para realizar la actualización y depuración del sistema.		
Fase 2. Puesta en marcha	act-03	Desarrollo del código fuente de los módulos del sistema	47
	tar-0301	Crear lógica de negocio y desarrollar código de cada una de las vistas del sistema.	
	tar-0302	Realizar pruebas unitarias de las nuevas funcionalidades y documentar bugs detectados y solucionados.	
	tar-0303	Realizar reunión con personal asignado del programa, para evaluar funcionalidades implementadas.	
	tar-0304	Realizar la actualización del repositorio con la finalización de cada vista o funcionalidad finalizada.	
	act-04	Implementar funcionalidades nuevas en entorno de producción	39
	tar-0401	Actualizar instancia en producción, con la transferencia ftp de los archivos de las nuevas funcionalidades terminadas.	
	tar-0402	Capacitar al personal designado del programa sobre el uso de las funcionales implementadas.	
	tar-0403	Ingresar Datos a la base del sistema, por medio de las funcionalidades implementadas en producción.	

Fase 3. Capacitación del Personal	act-05	Realizar presentación de Capacitación de acceso y uso de los módulos del sistema, por rol de usuario.	7
	tar-0501	Crear manual de usuario de los módulos del sistema para cada uno de los diferentes roles.	
	tar-0502	Realizar sesiones de capacitación con el personal del programa, por rol de usuario.	
Fase 4. Mantenimiento del sistema	act-06	Solventar errores de programación y funcionalidad del sistema en producción	120
	tar-0601	Notificar sobre error detectado en el entorno de producción.	
	tar-0602	Registrar el error y dar seguimiento a la depuración del mismo.	
	tar-0603	Actualizar sistema en producción con la depuración del error detectado.	

Tabla 4.24 Actividades y tareas para la implementación del sistema EELO.

En el diagrama de la ruta crítica se puede observar que las actividades principales Act-03 y Act-04 se ejecutan de forma secuencial dos veces, indicando una implementación por fases. De esta forma se logra implementar en producción una primera parte del sistema, la cual consiste en los datos generales y expedientes de los alumnos, e ingresar dichos datos al sistema, mientras se desarrolla y codifican los demás módulos y vistas.

Para realizar la estimación del tiempo requerido por cada una de las secciones de los módulos en la tabla 4.25, se ha tomado en cuenta el proceso desde la creación de la lógica del negocio por cada sección, basada en la estructura definida por el framework Laravel. Esto es, la creación de los modelos, los controladores y los métodos para obtener y actualizar los datos de la base de datos, la creación de las vistas y los archivos JavaScript y sus métodos para obtener, guardar y mostrar los datos de forma de dinámica.

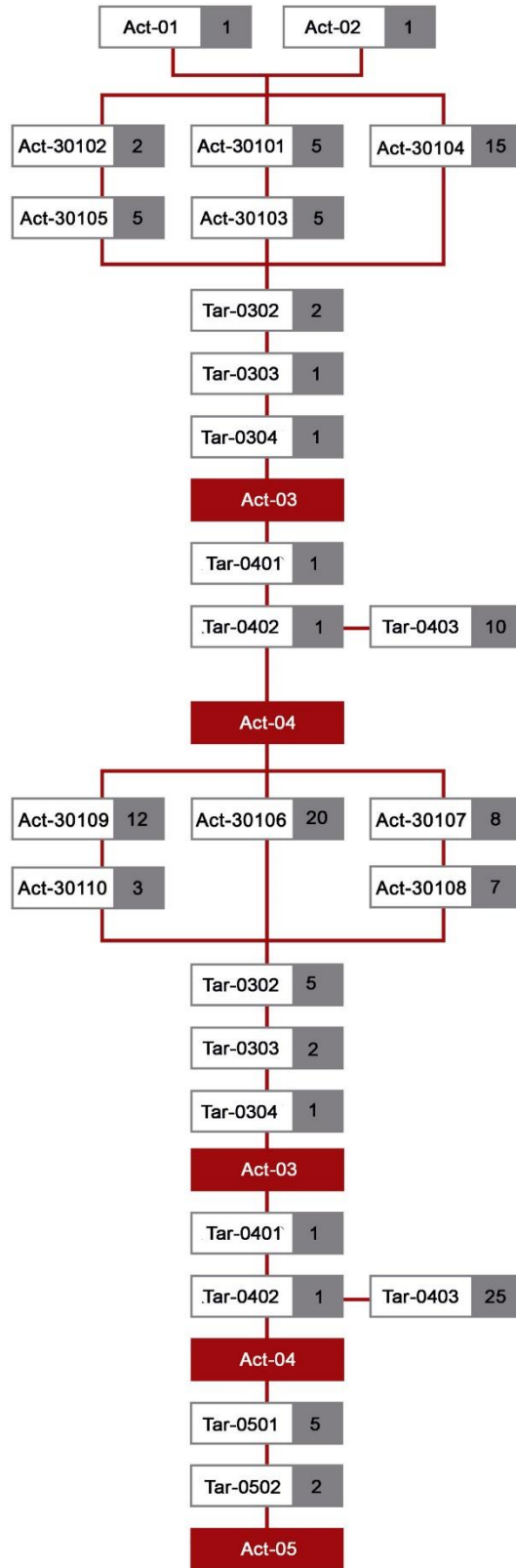


Figura 4.79 Diagrama de la ruta crítica para la implementación del sistema EELO.

ACTIVIDAD	MODULOS DEL SISTEMA	DESARROLLADOR	Tiempo estimado (Días)
	MODULO ADMINISTRADOR		27
30101	DATOS GENERALES	Eduardo Escalante	5
	Departamentos		1
	Municipios		1
	Centro Escolar		1
	Materias		1
	Criterios Actitudinales		1
	Actitudes		1
30102	PERSONAL	Alfredo Salazar	2
	Cargos		1
	Empleados		1
30103	ALUMNOS	Eduardo Escalante	5
	Listado Alumnos		1
	Registro Alumnos		2
	Registro de Responsables		2
30104	AÑO EN CURSO	David Fuentes	15
	Clases		2
	Secciones		2
	Asignación Alumno-sección		2
	Asignación Sección-docente		2
	Registro de Notas		5
	Generar Boletas de Notas		2
30105	LOGIN Y AUTENTICACION	Alfredo Salazar	5
30106	RECORD DE ALUMNOS		20
	Promedios Centro Escolar		5
	Promedios Materias Oportunidades		5
	Promedios Actitudinales		5
	Ranking	5	
	MODULO DOCENTE	David Fuentes	15
30107	REGISTRO DE NOTAS FGK		8
30108	REGISTRO DE NOTAS APTITUDINALES		7
	MODULO ALUMNO	Eduardo Escalante	15
30109	RECORD DE NOTAS POR AÑO		12
	Notas FGK		4
	Notas RDA		4
	Notas MINED		4
30110	CONFIGURACION DE USUARIO		3
	Actualización usuario y contraseña	3	

Tabla 4.25 Distribución de Vistas y Módulos con el equipo de desarrollo.

4.8 ESTADO DE LA IMPLEMENTACIÓN DEL SISTEMA

En esta sección se presenta el estado de la implementación del sistema, durante el primer semestre del año 2020. Como había sido mencionado en capítulos anteriores, el uso de un servicio de hosting compartido es un factor determinante en cuanto al uso de memoria y capacidad de tráfico de la aplicación web.

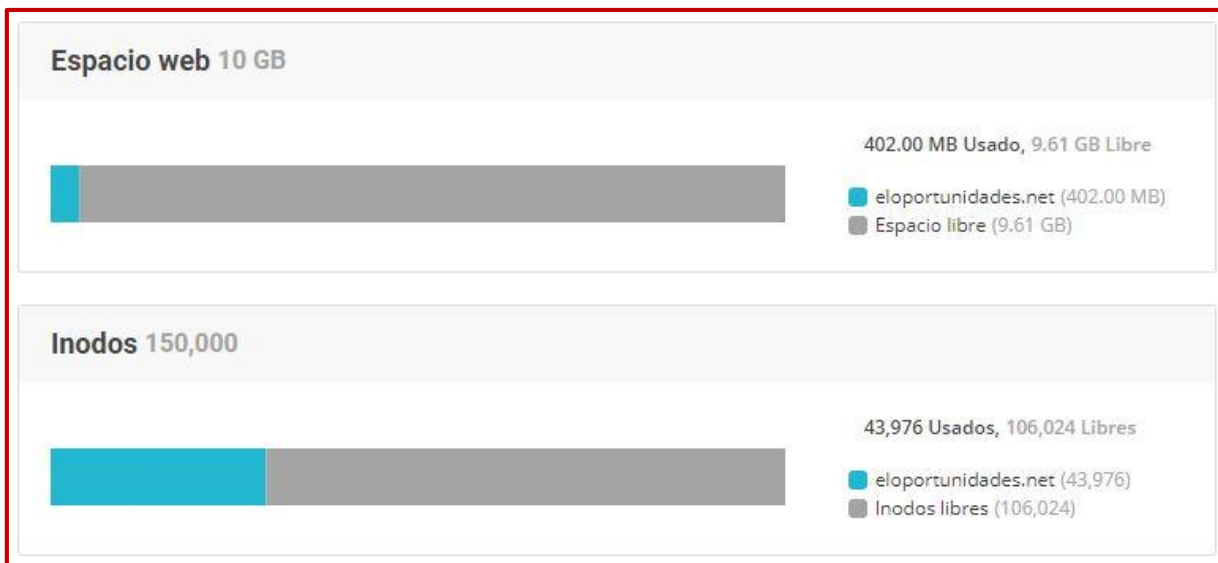


Figura 4.80 Espacio utilizado en disco del Sistema EELO.

En la figura 4.80 se puede observar el espacio en disco total utilizado por la aplicación del expediente en línea. La aplicación solamente utiliza un 4% de la capacidad de memoria del plan de hosting. Además, se tiene una estimación del crecimiento de 1% de uso de memoria en disco al año, en concepto de almacenamiento de las fotografías de los alumnos. La tabla 4.26 muestra el peso de la base de datos al primer semestre del año y a su vez permite realizar las estimaciones del crecimiento de la tabla en base a sus registros.

Nombre de tabla	Tamaño actual	Estimación/año
actitud	64.0 KiB	64.0 KiB
alumno	160.0 KiB	160.0 KiB
alumno_seccion	32.0 KiB	32.0 KiB
bitacora_alumno_centro_educativo	48.0 KiB	48.0 KiB
cargo	48.0 KiB	48.0 KiB
centro_educativo	48.0 KiB	48.0 KiB
class	48.0 KiB	48.0 KiB
criterio	48.0 KiB	48.0 KiB
departamento	48.0 KiB	48.0 KiB
empleado	112.0 KiB	112.0 KiB
estado	16.0 KiB	16.0 KiB
evaluaciones	64.0 KiB	64.0 KiB
evaluaciones_extras	16.0 KiB	16.0 KiB
grado_mined	16.0 KiB	16.0 KiB
grupo actitudinal	80.0 KiB	80.0 KiB
grupo_materia	80.0 KiB	80.0 KiB
log_cambio_grupo_docente	48.0 KiB	48.0 KiB
log_cambio_seccion_alumno	64.0 KiB	64.0 KiB
materia	48.0 KiB	48.0 KiB
migrations	16.0 KiB	16.0 KiB
municipio	64.0 KiB	64.0 KiB
notas_extras	32.0 KiB	32.0 KiB
notas_fgk	592.0 KiB	
notas_mined	528.0 KiB	
nota_actitudinal	5.1 MiB	
ocupacion	16.0 KiB	16.0 KiB
password_resets	32.0 KiB	32.0 KiB
periodo	16.0 KiB	16.0 KiB
responsable	112.0 KiB	
sección	48.0 KiB	48.0 KiB
tipo_responsable	16.0 KiB	16.0 KiB
users	96.0 KiB	96.0 KiB
Total de tablas 32	7.7 MiB	

Tabla 4.26 Detalle del espacio en disco de la Base de datos EELO.

Las figuras 4.81 y 4.82 muestran el número de usuarios únicos y número de visitas que se tuvieron en la fecha del 5 de junio de 2020. En dicha fecha, los alumnos del programa Oportunidades fueron notificados, para que accedieran al sistema a consultar sus calificaciones del periodo 1 del año escolar. Aun cuando el plan no tiene un límite de visitas por mes, si considera 10,000 como número de visitas óptimo de acuerdo a la capacidad del servidor. Durante el resto del mes de junio, el total de visitas fue del 27% sobre el número de visitas óptimo.



Figura 4.82 143 usuarios únicos el 5 de junio 2020.

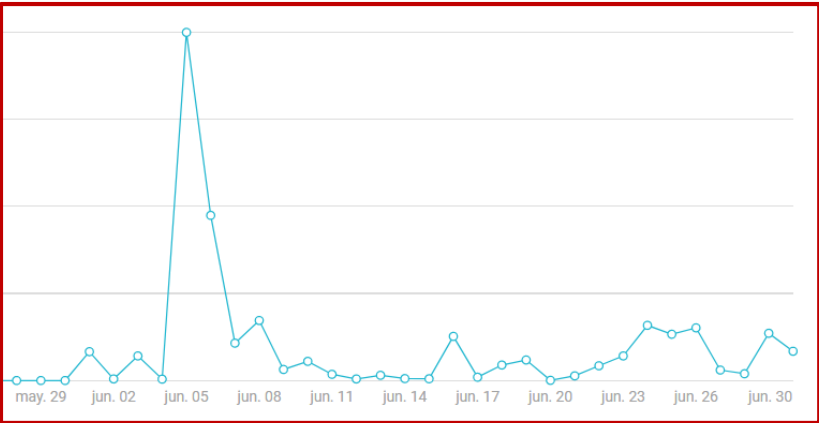


Figura 4.81 1798 Visitas realizadas el 5 de junio 2020.

La fase final del plan de implementación consiste en dar mantenimiento a todos los errores y fallos que presente el sistema EELO. A partir del segundo semestre del año 2020 el sistema se encuentra completamente en producción y accesible para el personal administrativo, docentes y alumnos del programa Oportunidades.

En el ANEXO V se presenta el registro de todos los errores y bugs que el sistema ha presentado hasta el mes de noviembre del 2020. Los errores han sido reportados por el personal administrativo y docentes y cada uno ha sido solventado por parte del equipo de desarrollo y dichas soluciones son actualizadas en el repositorio del código fuente del sistema.

CAPITULO V: CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- **Logro de Objetivos planteados al inicio del proyecto.**

La implementación del sistema EELO aun dentro de las capacidades ofrecidas por los planes de hosting compartido, que en un primer lugar fueron consideradas una limitante, ha logrado cumplir con los objetivos principales del proyecto de implementación, ya que el sistema ha permitido a los alumnos y sus padres de familia poder tener acceso a un detalle completo de sus calificaciones y su expediente en general, en cualquier momento y lugar. De igual forma, la labor de los docentes y personal administrativo de mantener y actualizar los registros de las calificaciones se ha reducido ya que una sola tarea de cargar las notas al sistema, permite que toda la información necesaria para realizar las demás tareas de control académico esté disponible con solamente un par de clics.

- **Eficiencia de la metodología de trabajo para la ejecución del proyecto.**

La metodología del ciclo de vida del sistema informático tradicional definida al inicio, para llevar a cabo el desarrollo e implementación del proyecto, se consideró adecuada dada la naturaleza del sistema a desarrollar. Aun cuando el sistema tenía establecidos sus requerimientos de forma puntual, a lo largo del desarrollo del sistema surgieron cambios a nivel de diseño. Además, los retrasos y demoras en la fase de desarrollo, afectaron directamente la implementación del sistema. Como equipo de desarrollo, estos resultados sugieren que, sin importar la naturaleza o tipo de aplicación a desarrollar, cada vez resulta más conveniente la aplicación de metodologías ágiles las cuales permiten ir teniendo un ciclo de implementación más rápido.

- **La importancia del hardware como servicio.**

En el área de implementación de aplicaciones y sistemas informáticos en las empresas y organizaciones de todo tipo, resulta evidente y cada vez con mayor frecuencia, la factibilidad de optar por la adquisición de infraestructura de hardware como servicio. Esto porque permite aplicar una capa de abstracción al área de

mantenimiento y redes informáticas, permitiendo a las unidades de IT de las empresas, enfocarse más en el área de lógica y desarrollo de las aplicaciones, permitiendo una adaptabilidad más rápida a los cambios del entorno.

5.2 RECOMENDACIONES

- Se deben realizar respaldos manualmente de la base de datos del Sistema EELO de forma periódica, además de los respaldos automáticos que son generados por el servicio de hosting del proveedor de SiteGround.
- Es necesario considerar y contratar los servicios adicionales ofrecidos por el proveedor de hosting compartido, referentes a mantener y aumentar la seguridad y estabilidad del sitio web.
- Para mantener la integridad de los datos y hacer un buen uso del sistema, se debe llevar a cabo cada una de las tareas de acuerdo a los pasos detallados en el manual de usuario; ya que dichos procedimientos son los que se encuentran depurados por el equipo de desarrollo.
- El diseño del sistema se llevó a cabo tomando en cuenta la delimitación de su implementación en la sede de Santa Ana. Por lo tanto, para futuras implementaciones en otra sede del programa Oportunidades, la implementación recomendada es la de crear otra instancia de la aplicación y de la base de datos dedicada a gestionar los datos para dicha sede. En caso de que se quiera manejar todos los datos de múltiples sedes, es necesario llevar a cabo trabajo de diseño para escalar la base de datos y crear nuevas vistas y lógica del negocio para llevar a cabo dicha integración.
- Se recomienda una participación activa e interés hacia el Sistema sobre todas las funciones que este desempeña.

- Se recomienda el registro adecuado de toda la información para un mejor manejo de ella y confiabilidad al momento de necesitarla.
- Realizar reuniones con el personal que hace uso del sistema para mantener al tanto de nuevas funciones o características del sistema, además de errores comunes que se realizan por parte de los usuarios.
- Se recomienda al personal administrativo del programa Oportunidades designar personal que cuente con las habilidades técnicas adecuadas, para el mantenimiento y monitoreo del sistema.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS BIBLIOGRÁFICAS

- Andrade Ruiz, M. G., Aquino Pineda, G. H., Mestizo Aguilar, F. d., & Monge, L. E. (2011). Sistema Informático de Administración del Expediente Académico en Línea para la Facultad de Ingeniería y Arquitectura de la Universidad de El Salvador. San Salvador, El Salvador.
- Autores contribuyentes de documentación PHP. (s.f.). Manual de PHP. Obtenido de php.net: <https://www.php.net/manual/es/intro-whatcando.php>
- Colaboradores Wikipedia. (2020). Alojamiento compartido. Obtenido de Wikipedia, La enciclopedia libre.: https://es.wikipedia.org/w/index.php?title=Alojamiento_compartido&oldid=124363814
- Colaboradores Wikipedia. (febrero de 2020). Metodología. Obtenido de Wikipedia, La enciclopedia libre.: <https://es.wikipedia.org/wiki/Metodología>
- DataTables. (s.f.). DataTables Manual. Obtenido de <https://datatables.net/manual/>
- El Programa - Programa Oportunidades. (s.f.). Obtenido de Programa Oportunidades: <http://www.oportunidades.org.sv/el-programa--3>
- Kendall, K. E., & Kendall, J. E. (2011). Análisis y diseño de sistemas. México: PEARSON EDUCACIÓN.
- Laravel. (s.f.). Laravel Documentation. Obtenido de <https://laravel.com/docs>

- Materialize. (s.f.). Materialize Documentation. Obtenido de <https://materializecss.com/>
- MySQL. (s.f.). MySQL Documentation. Obtenido de <https://dev.mysql.com/doc/>
- Programa Oportunidades - Quienes Somos. (s.f.). Obtenido de programa oportunidades: <http://www.oportunidades.org.sv/quienes-somos>
- Rivera Vázquez, L. d., Rodríguez Meléndez, H. V., Sánchez Garay, D. C., & Sánchez
- Montoya, F. P. (2011). Sistema Informático para la Administración Académica en el Centro de Enseñanza de Idiomas Extranjeros de la Universidad de El Salvador. San Salvador, El Salvador.
- SiteGround. (s.f.). Web Hosting Platform. Obtenido de <https://www.siteground.com/>
- Software & Systems Engineering Standards Committee. (1998). Especificación de Requisitos, IEEE 830.

ANEXOS

- **Anexo i.** Documentos Excel de registro de notas y colector de promedios, previo al sistema EELO.

No.	APELLIDOS	NOMBRES	SECCIÓN	NOTA	10'	10'	15'	15'	50.0%	5.0%	5.0%	20.0%	20.0%	50.0%
				100.0%	Tarea1 Documento Maestro	Tarea 2 Pestaña Programador	Práctica Evaluada Diseño y referencias	Práctica Correspondencia		Avance 1	Avance 2	Entrega Final Periódico	Campaña Bullying	
1	Aguilar Figueroa	Michelle Abigail	C2	9.6	10.0	10.0	9.0	10.0	4.9	9.0	9.3	9.6	9.5	4.7
2	Aguilar Saldaña	Ivania Lizbeth	C3	8.7	10.0	9.0	8.0	8.0	4.3	9.0	9.0	9.1	8.5	4.4
3	Alarcón Ramírez	Evelyn Karina	C2	9.4	10.0	10.0	9.0	9.0	4.7	9.0	9.3	9.6	9.5	4.7
4	Arce Tejada	Tatiana Alejandra	C3	8.8	10.0	9.0	8.0	8.0	4.3	9.0	8.9	8.8	9.0	4.5
5	Arias Molina	Daniela Elizabeth	C2	9.6	10.0	10.0	9.0	10.0	4.9	9.0	9.3	9.6	9.5	4.7
6	Barrientos Cashpal	Susana Marcela	C3	8.7	9.0	10.0	8.0	8.0	4.3	9.0	9.0	9.1	8.5	4.4
7	Calderón Flores	Kenia Abigail	C1	8.8	10.0	9.0	8.0	8.0	4.3	9.0	9.3	9.7	8.5	4.5
8	Calderón Hernández	Jeansy Stephannie	C2	9.3	10.0	10.0	9.0	9.0	4.7	9.0	9.3	9.6	9.0	4.6
9	Carballo Flores	Ratchell Jazmín	C1	9.1	9.0	9.0	8.0	8.0	4.2	9.0	9.4	9.7	10.0	4.9
10	Castellanos Calles	José Rodrigo	C3	9.1	10.0	10.0	8.0	9.0	4.6	9.0	9.1	9.2	9.0	4.5

Sec.	Class 2020	compu	english	values	math	lingui	ov	prom	pal's Honor	Honor Roll	Principal's	de Prev	Deserción/Expulsión
1 C2	Aguilar Figueroa, Michelle Abigail	9.6	7.7	9.9	10.0	9.7	8.8	9.3					
2 C3	Aguilar Saldaña, Ivania Lizbeth	8.7	8.5	9.3	9.5	9.5	9.1	9.1		Honor Roll			
3 C2	Alarcón Ramírez, Evelyn Karina	9.4	8.3	9.9	10.0	9.5	9.1	9.4		Honor Roll			
4 C3	Arce Tejada, Tatiana Alejandra	8.8	7.4	9.4	9.7	9.3	9.0	8.9					
5 C2	Arias Molina, Daniela Elizabeth	9.6	8.7	9.9	10.0	9.7	9.4	9.5		Honor Roll			
6 C3	Barrientos Cashpal, Susana Marcela	8.7	8.0	9.3	9.5	9.5	9.0	9.2		Honor Roll			
7 C1	Calderón Flores, Kenia Abigail	8.8	8.7	9.9	10.0	9.7	8.7	9.3		Honor Roll			
8 C2	Calderón Hernández, Jeansy Stephannie	9.3	9.1	9.8	9.7	9.8	9.1	9.5	Principal's				
9 C1	Carballo Flores, Ratchell Jazmín	9.1	8.1	9.4	9.4	9.3	8.7	9.0		Honor Roll			
10 C3	Castellanos Calles, José Rodrigo	9.1	8.3	9.7	9.6	9.5	9.2	9.2		Honor Roll			









- **Anexo ii.** Documentos Excel de registro de notas actitudinales, previos al sistema EELO.

	Apellidos	Nombres	Temple			PROM1	Autocontrol (dentro del aula)		PROM2	Autocontrol (Interpersonal)			PROM3	Inteligencia Social			PROM4	
			puntual	tareas	asistencia		instrucciones	enfocado a tareas		interacción grupal	autodisciplina	ideas claras		respetuoso	empatía	rol de líder		
1	Aguilar Figueroa	Michelle Abigail	5	5	5	5.00	5	5	5.00	4	5	5	5	5	5	5	5	5
2	Aguilar Saldaña	Ivania Lizbeth	5	5	5	5.00	5	5	5.00	4	5	5	5	5	5	3	4.3	
3	Alarcón Ramírez	Evelyn Karina	5	5	5	5.00	5	5	5.00	3	5	5	4	5	5	5	5	5
4	Arce Tejada	Tatiana Alejandra	5	5	4	4.67	3	3	3.00	4	4	4	4	5	5	3	4.3	
5	Arias Molina	Daniela Elizabeth	5	5	5	5.00	5	5	5.00	4	5	5	5	5	5	5	5	5
6	Barrientos Cashpal	Susana Marcela	5	5	4	4.67	3	4	3.50	5	5	4	5	5	5	4	4.7	
7	Calderón Flores	Kenia Abigail	5	5	5	5.00	5	5	5.00	5	5	5	5	5	5	5	5	5
8	Calderón Hernández	Jeansy Stephannie	5	5	5	5.00	5	5	5.00	4	5	5	5	5	5	4	4.7	
9	Carballo Flores	Ratchell Jazmín	5	5	5	5.00	5	5	5.00	5	5	4	5	5	5	5	5	5
10	Castellanos Calles	José Rodrigo	5	5	5	5.00	5	5	5.00	5	5	4	5	5	5	5	5	5

			Temple			PROM1	Autocontrol (dentro del aula)		PROM2	Autocontrol (Interpersonal)			PROM3	Inteligencia Social			PROM4				
			puntual	tareas	asistencia		instrucciones	enfocado a tareas		interacción grupal	autodisciplina	ideas claras		lenguaje respetuoso	escucha y empatía	rol de líder					
1	Aguilar Cruz	Andrea Liliana	5	5	4	5	4	5	4.67	3	5	3	5	5	5	5	2	3.7			
2	Alvarado Tejada	Tatiana Abigail	5	5	5	5	4	5	4.83	5	5	5	5	5	5	5	2	4.5			
3	Asencio Castaneda	Katherine Lisbeth	5	4	5	4	4	4	4.33	5	4	5	4	5	5	4	4	4.7			
4	Barillas Hernández	Gerardo Javier	5	5	5	4	5	5	4.83	5	5	4	5	5	5	5	3	4.7			
5	Canizález Toledo	María de los Angeles	5	5	5	5	5	5	5.00	4	5	5	5	5	5	4	3	4.5			
6	Castillo Albanés	Josué Adiel	5	5	4	3	4	4	4.17	4	5	5	3	4.25	4	3	4	4	2	4.3	
7	Chámul Pacheco	Katherine Esmeralda	5	5	5	5	5	5	5.00	5	5	5	4	4.75	4	4	5	4	4	4	4.2
8	Chávez Baños	Paola Nicolle	5	5	5	5	5	5	5.00	3	3	3	5	3	4	3.5	5	5	5	2	4.5
9	Chávez García	Jessica Maricela	5	5	4	5	4	5	4.67	5	5	4	4	4.75	3	4	5	5	4	4	4.3
10	Choto Martínez	Fátima Lourdes	5	5	5	5	5	5	5.00	5	4	5	5	4.75	4	5	5	4	4	5	4.5

- **Anexo iii.** Plantillas de boletas de entrega de notas, previo al sistema EELO.

Programa Oportunidades – FGK		“Character Growth” REPORT CARD	
Nombre estudiante: Andrea Liliana Aguilar Cruz		Santa Ana FT	
Sección: C2	Class 2019 SA FT	Fecha: 14 de febrero de 2020	

OVERALL SCORE			6.99
Perseverancia			6.94
	1	Termina lo que empieza	6.83
	2	Se adhiere a proyectos o actividades a largo plazo	7.00
	3	Trata con más ganas aún después de haber fallado	5.33
	4	Permanece comprometido con las metas	7.50
	5	Se mantiene trabajando duro cuando siente desfallecer	7.00
Optimismo			7.00
	1	Cree que el esfuerzo puede mejorar su futuro	7.00
	2	Busca alternativas para mejorar cuando algo sale mal	7.00
	3	Se mantiene motivado aun cuando las cosas no van bien	7.17
	4	Cree que puede mejorar en lo que no ha salido bien	7.00
Trabajo en la escuela			7.00
	1	Viene a clases preparado/a	7.00
	2	Recuerda y sigue indicaciones	7.00
	3	Maneja el tiempo eficientemente	7.00
	4	Pone atención y resiste distracciones	7.00
Inteligencia interpersonal			7.00
	1	Se mantiene calmado ante críticas	7.00
	2	Permite a otros hablar sin interrumpir	7.00
	3	Es educado con los compañeros y maestros	7.00
	4	Controla sus emociones y temperamento	7.00
Gratitud			7.00
	1	Reconoce lo que otros hacen por él o ella	7.00
	2	Muestra aprecio por las oportunidades	7.00
	3	Expresa aprecio al decir ¡Gracias!	7.00
	4	Hace algo bueno para los demás en agradecimiento	7.00
Inteligencia social			7.00
	1	Encuentra soluciones durante conflictos con otros	7.00
	2	Muestra que le importan los sentimientos de otros	8.00
	3	Adaptable a diferentes ámbitos sociales	7.00
Curiosidad			7.00
	1	Dispuesto a explorar nuevas cosas	7.00
	2	Pregunta para aprender y entender mejor	7.00
	3	Demuestra interés en aprender	7.00
Ánimo			7.00
	1	Participa activamente	7.00
	2	Muestra entusiasmo	7.00
	3	Enfrenta nuevas situaciones con emoción y energía	7.00

Scale

1-3 needs to improve

4-7 a little more effort

8-10 outstanding student

f. 
 f. Roberto Flores
 Director

f. _____
 Padre/Madre Familia

f. _____
 Estudiante



oportunidades
 FUNDACIÓN GLORIA DE KRIETE



Nombre: **Zepeda Vásquez, Sonia Abigail**

Centro Oportunidades: **Santa Ana**

Sección: **C3**

Modalidad: **Tiempo Completo**

Fecha: **7 jul 2018**

ÁREAS ACADÉMICAS	
Materias	Notas
Inglés	8.4
Matemáticas	7.7
Computación	9.5
Formación Lingüística	9.0
Valores	9.3
Orientación Vocacional	8.8
Promedio	8.8

Status

Fundación



Observaciones:

Luis Roberto Flores Perla
Director Centro NIVELA SA FT

Firma Responsable de alumno

o **Anexo v.** Detalle de mantenimiento de errores y modificaciones.

Fecha	Modulo/Submodulo/Vista	Detalle del problema	Tipo de problema	Estado
27/01/2020	vista alumno	el carnet en vez de autogenerarse, debe ser introducido por el administrador	modificación	solventado
11/2/2020	vista alumno	fallo en el registro de los responsables del alumno	error	solventado
27/2/2020	asignacion secciones-docente	la vista solamente carga las secciones de tercer año	error	solventado
10/3/2020	vista alumno	el carnet de los alumnos cuenta con un correlativo de 4 digitos, no 3	modificación	solventado
11/3/2020	vista alumno	edicion de carnet. El carnet debe estar habilitado para poder modificarse	modificación	solventado
21/4/2020	vista alumno	campos obligatorios en la vista para agregar un alumno, dejarlos opcionales	modificación	solventado
12/5/2020	importar notas RDA	tiempo de esperar de respuesta de la peticion	error	solventado
30/5/2020	docente	cargar listado de alumno en orden alfabetico	modificación	solventado

30/5/2020	login	poder realizar cambio de nombre de usuario y de contraseña	nueva funcionalidad	solventado
30/5/2020	docente -evaluaciones	agregar nombre corto a las evaluaciones	nueva funcionalidad	solventado
30/5/2020	docente -evaluaciones	validar el total del porcentaje de todas las evaluaciones de un periodo	error	solventado
30/5/2020	alumno	mostrar nombre corto de las evaluaciones en el record de notas	modificación	solventado
30/5/2020	alumno	alinear promedio al final de la fila	modificación	solventado
30/5/2020	alumno	cambiar "Diploma" por "Estatus"	modificación	solventado
30/5/2020	alumno - estado	agregar campo de observacion y quitar estado "reprobado"	modificación	solventado
8/6/2020	record fgk	redondeo en calculo de los promedios	error	solventado
11/7/2020	docente	vista distorcionada de cuadrícula de notas RDA	error	solventado
11/7/2020	docente	cargar en la cuadrícula y en las plantillas de notas, solamente los alumnos activos	modificación	solventado

20/7/2020	vistas data table	las vistas que tienen datatable, no cargan las funciones de los botones despues de la primer pagina.	error	solventado
25/7/2020	configuracion usuario	error al cambiar la contraseña por defecto, no se realiza el cambio	error	solventado
25/7/2020	configuracion usuario	solicitud de boton para reestablecer contraseña de alumnos, en caso de que la olviden	nueva funcionalidad	solventado
31/8/2020	Erecord notas Centro escolar	no aparecen las notas que ya fueron registradas	error	solventado