

UNIVERSIDAD PRIVADA ANTONIO GUILLERMO URRELO



FACULTAD DE INGENIERÍA

Carrera Profesional de Ingeniería Informática y de Sistemas

IMPLEMENTACIÓN DE UN SISTEMA ACADÉMICO WEB Y SU INFLUENCIA
PARA MEJORAR EL PROCESO DE MATRÍCULAS EN LA I.E. TÚPAC AMARU II

N° 16498–HUARANGO–SAN IGNACIO– CAJAMARCA

Bach. Aliaga Cabrera, Israel David

Bach. Álvarez Sarmiento, Erwin Miler

Asesor(a) :

Dra. Cruzado Vásquez, Diana Jakelin

Cajamarca-Perú

Mayo, 2023

UNIVERSIDAD PRIVADA ANTONIO GUILLERMO URRELO



FACULTAD DE INGENIERÍA

Carrera Profesional de Ingeniería Informática y de Sistemas

IMPLEMENTACIÓN DE UN SISTEMA ACADÉMICO WEB Y SU INFLUENCIA

PARA MEJORAR EL PROCESO DE MATRÍCULAS EN LA I.E. TÚPAC

AMARU II N° 16498–HUARANGO–SAN IGNACIO– CAJAMARCA

Tesis presentada en cumplimiento parcial de los requerimientos para optar por el Título

Profesional de Ingeniero Informático y de Sistemas

Bach. Aliaga Cabrera, Israel David

Bach. Álvarez Sarmiento, Erwin Miler

Asesor(a):

Dra. Cruzado Vásquez, Diana Jakelin

Cajamarca-Perú

Mayo, 2023

COPYRIGHT © 2023 by

Bach. Aliaga Cabrera Israel David

Bach. Álvarez Sarmiento Erwin Miler

Todos los derechos reservados.

UNIVERSIDAD PRIVADA ANTONIO GUILLERMO URRELO
FACULTAD DE INGENIERIA
CARRERA PROFESIONAL DE INGENIERIA INFORMATICA Y DE
SISTEMAS

Aprobación de Tesis para optar Título Profesional de Ingeniero Informático y de
Sistemas

**“IMPLEMENTACIÓN DE UN SISTEMA ACADÉMICO WEB Y SU
INFLUENCIA PARA MEJORAR EL PROCESO DE MATRÍCULAS EN LA I.E.
TÚPAC AMARU II N° 16498–HUARANGO–SAN IGNACIO– CAJAMARCA”**

Jurado evaluador

Presidente: _____

Secretario: _____

Vocal: _____

Asesor: _____

DEDICATORIA(S)

Agradecer en primer lugar a nuestro creador Dios todopoderoso por ayudarme a seguir adelante cada día y superarme como persona y como profesional. Luego a mis padres por contar conmigo en cada momento de mi vida y estar apoyándome emocionalmente y moralmente.

Bach. Erwin Miler Álvarez Sarmiento

Este trabajo se puedo realizar con la ayuda de nuestro DIOS que nos ayudó día a día, asimismo lo quiero dedicar a toda mi familia, quienes han sido mi motor y motivo para cumplir uno de mis sueños, ayudándome a mejorar como ser humano, a realizar este reto el cual es de mucha importancia en mi vida profesional.

Bach. Israel David Aliaga Cabrera

AGRADECIMIENTO(S)

En primer lugar, agradecer a DIOS por permitirnos a cumplir nuestros sueños, agradecer a nuestros padres ya que gracias a ellos cumplimos nuestras metas trazadas, a nuestros docentes de escuela, colegio y universidad por su apoyo incondicional el cual nos sirve para ser mejores personas en la vida diaria.

Al director de la Institución Educativa Túpac Amaru II N°16498-Huarango-San Ignacio-Cajamarca, por permitirnos desarrollar el sistema académico web durante este periodo de investigación de nuestra tesis.

Bach. Israel David Aliaga Cabrera

Bach. Erwin Miler Alvarez Sarmiento

RESUMEN

Las tecnologías de información han aumentado innumerablemente, por ello nos hemos orientado a sistemas académicos web para la mejora de los procesos en las IE.

Nuestro presente proyecto se titula implementación de un sistema académico web y su influencia para mejorar el proceso de matrículas en la I.E. Túpac Amaru II N° 16498–Huarango–San Ignacio– Cajamarca. Nuestro objetivo principal es mejorar el proceso de matrículas en la “I.E. Túpac Amaru II N° 16498, Huarango – San Ignacio- Cajamarca, mediante la implementación de un sistema web, teniendo la variable independiente Sistema Académico web y su variable Dependiente mejoras en el proceso de matrículas, contando con dimensiones e indicadores respectivos. El tipo de investigación es aplicada y de diseño cuasiexperimental, enfoque cuantitativo, teniendo como unidad de análisis al personal administrativo (3) y los padres de familia (54) de la IE, incluyendo los expertos (3); hemos utilizado cuestionarios y encuestas para extraer resultados utilizando como herramientas estadísticas los programas de Microsoft Excel e IBM SPSS. Utilizamos la técnica de Chi-cuadrado para ver si se acepta o se rechaza la hipótesis para confirmar si en realidad se mejoró o no a optimizar el tiempo en matrículas en la IE. Teniendo como resultados de los expertos un 66.7% totalmente de acuerdo y un 33.3% de acuerdo y la comunidad educativa mostró un 56.1% totalmente de acuerdo y un 36.8% de acuerdo y un 7% ni de acuerdo ni en desacuerdo, con respecto a la facilidad del sistema web. Como respuesta de la hipótesis, utilizando la técnica de Chi-cuadrado, que fue tabulado 5.9915, fue menor al X^2 calculado que fue 28.2 y se rechazó la H_0 y se aceptó la H_1 , en otras palabras, la implementación del sistema académico web influye en la mejora de procesos de matrículas en la IE.

Palabras clave: Sistema web, Implementación.

ABSTRACT

Information technologies have increased innumerable, so we have focused on web academic systems for the improvement of processes in the IE.

Our present project is titled implementation of a web academic system and its influence to improve the enrollment process at I.E. Tupac Amaru II No. 16498–Huarango–San Ignacio–Cajamarca. Our main objective is to improve the registration process in the "I.E. Tupac Amaru II N° 16498, Huarango - San Ignacio- Cajamarca, having the independent variable Academic web system and its dependent variable improvements in the enrollment process, having respective dimensions and indicators. The type of research is applied and of quasi-experimental design, quantitative approach, having as unit of analysis the administrative staff (3) and parents (54) of the IE, including experts (3); we used questionnaires and surveys to extract results using Microsoft Excel and IBM SPSS programs as statistical tools. We used the Chi-square technique to see if the hypothesis is accepted or rejected to confirm whether or not there was an improvement in optimizing enrollment time at the IE. The results of the experts were 66.7% totally agree and 33.3% agree and the educational community showed 56.1% totally agree and 36.8% agree and 7% neither agree nor disagree, with respect to the ease of the web system. As a response of the hypothesis, using the Chi-square technique, which was tabulated 5.9915, was less than the calculated X^2 which was 28.2 and H_0 was rejected and H_1 was accepted, in other words, the implementation of the web academic system influences the improvement of enrollment processes in the IE.

Key words: Web system, Implementation.

ÍNDICE

DEDICATORIA(S)	i
AGRADECIMIENTO(S)	ii
RESUMEN	iii
ABSTRACT	iv
1. INTRODUCCION	1
1.1. Planteamiento del problema	1
1.2. Formulación del problema	2
1.3. Justificación	2
1.4. Objetivos de la investigación	3
1.4.1. Objetivo General	3
1.4.2. Objetivos específicos	3
1.5. Hipótesis de la investigación	3
1.5.1. Hipótesis	4
2. MARCO TEÓRICO	6
2.1. Antecedentes que sustentan la investigación	6
2.2. Bases conceptuales	9
2.2.1. Diseño	9
2.2.2. Implementar	9
2.2.3. Bases de datos	9
2.2.4. Sistemas web	10
2.2.5. Implementación de software	10
2.2.6. Servidor de aplicación web	10
2.2.7. Modelo cliente/servidor	10
2.2.8. Metodología ágil	11
2.2.8.1. Proceso Unificado de Rational	11
2.2.8.2. Método en cascada	12
2.2.9. Metodologías XP	12
2.2.9.1. Fases de la metodología XP	12
2.2.10. Proceso de matriculas	13
2.2.11. ISO 25000 – 25010	13
2.2.12. Laragon	14
2.2.13. Php	15
2.2.14. Admin	15

2.2.15.	Sublime text	15
2.2.16.	Microsoft Excel.....	15
2.2.17.	Escala de Likert.....	16
3.	METODOLOGÍA DE LA INVESTIGACIÓN	16
3.1.	Unidad de análisis, universo y muestra	16
3.1.1.	Unidad de análisis.....	16
3.1.2.	Población	16
3.1.3.	Muestra.....	17
3.2.	Métodos de investigación	18
3.2.1.	Tipo de investigación.....	18
3.2.2.	Diseño de la investigación	19
3.3.	Enfoque de la investigación	19
3.4.	Técnicas e instrumentos de recolección de datos.....	20
3.4.1.	Técnicas de investigación	20
3.4.1.1.	Encuestas	20
3.4.1.2.	Entrevistas.....	21
3.4.2.	Instrumentos	22
3.4.2.1.	Cuestionarios	22
3.4.3.	Técnicas de análisis de datos.....	23
3.4.4.	Validación de instrumentos	23
3.4.5.	Herramienta estadística	24
4.	IMPLEMENTACIÓN	24
4.1.	RECURSOS PARA LA IMPLEMENTACIÓN	24
4.2.	Planificación del proyecto.....	25
5.	RESULTADOS Y DISCUSIÓN.....	153
5.1.	RESULTADOS	153
5.1.1.	Resultados del cuestionario del sitio web dirigido a expertos.....	153
5.1.2.	Resultados de cuestionario del sistema web dirigido a la comunidad educativa	159
5.2.	Contrastación de hipótesis.....	163
5.3.	DISCUSIÓN DE RESULTADOS	167
6.	CONCLUSIONES.....	172
7.	RECOMENDACIONES.....	173
8.	REFERENCIAS	174
	ANEXOS	179

LISTA DE TABLAS

Tabla 1: Matriz de Operacionalización de Variables.....	5
Tabla 2: Encargados de evaluar el sistema.	18
Tabla 3: Recursos necesarios	25
Tabla 4: Historia de Usuario N.º 01	25
Tabla 5: Historia de Usuario N.º 02	26
Tabla 6: Historia de Usuario N.º 03	26
Tabla 7: Historia de Usuario N.º 04	27
Tabla 8: Historia de Usuario N.º 05	27
Tabla 9: Historia de Usuario N.º 06	28
Tabla 10: Historia de Usuario N.º 07	28
Tabla 11: Historia de Usuario N.º 08	29
Tabla 12: Historia de Usuario N.º 09	29
Tabla 13: Historia de Usuario N.º 10	30
Tabla 14: Historia de Usuario N.º 11	30
Tabla 15: Historia de Usuario N.º 12	31
Tabla 16: Historia de Usuario N.º 13	31
Tabla 17: Historia de Usuario N.º 14	32
Tabla 18: Historia de Usuario N.º 15.....	32
Tabla 19: Historia de Usuario N.º 16	33
Tabla 20: Historia de Usuario N.º 17	33
Tabla 21: Historia de Usuario N.º 18	34
Tabla 22: Historia de Usuario N.º 19	34
Tabla 23: Historia de Usuario N.º 20	35
Tabla 24: Historia de Usuario N.º 21	35
Tabla 25: Historia de Usuario N.º 22	36
Tabla 26: Historia de Usuario N.º 23	36
Tabla 27: Historia de Usuario N.º 24	37
Tabla 28: Plan de entregas	38
Tabla 29: Plan de iteraciones	39
Tabla 30: Frecuencia observada, datos de la investigación.	165
Tabla 31: Frecuencia esperada.....	165
Tabla 32: Cálculo de chi-cuadrado.	166

FIGURAS

Figura 1: Botón de registrar matrícula.	41
Figura 2: Botón de detalle de registrar matrícula.	41
Figura 3: Botón de perfil de usuario	42
Figura 4: Botón de registrar apoderado.	43
Figura 5: Botón de registrar alumno.	43
Figura 6: Botón de registrar docentes.	44
Figura 7: Botón de registrar cursos.	44
Figura 8: Botón de registrar horario.	45
Figura 9: Botón de registrar año académico.	45
Figura 10: Botón de registrar secciones(Grado).	46
Figura 11: Botón de registrar secciones(Nivel).	46
Figura 12: Botón de registrar secciones(Sección).	46
Figura 13: Migraciones.	47
Figura 14: Comunicación de la base de datos de Lavarel con phpmyadmin.	47
Figura 15: Migración de crear Matrícula (Registrar Matrícula).	48
Figura 16: Migración de crear Detalle de Matrícula (Detalle Matrícula).	49
Figura 17: Migración de crear usuario.	50
Figura 18: Migración de crear apoderado.	50
Figura 19: Migración de crear alumno.	51
Figura 20: Migración de crear Docente.	51
Figura 21: Migración de crear Cursos.	52
Figura 22: Migración de crear Horario.	52
Figura 23: Migración de crear Año Académico.	53
Figura 24: Migración de crear Secciones (Registrar Grados).	53
Figura 25: Migración de crear Secciones (Registrar Nivel).	54
Figura 26: Migración de crear Secciones (Registrar Sección).	54
Figura 27: Controladores.	55
Figura 28: Controlador de Matrícula (Registrar Matrícula).	55
Figura 29: Controlador de Matrícula (Registrar Matrícula).	56
Figura 30: Controlador de Matrícula (Registrar Matrícula).	56
Figura 31: Controlador de Matrícula (Registrar Matrícula).	57
Figura 32: Controlador de Matrícula (Registrar Matrícula).	58
Figura 33: Controlador de Usuario (Perfil de usuario).	59

Figura 34: Controlador de Usuario (Perfil de usuario).	59
Figura 35: Controlador de Apoderado (Registrar Apoderado).	60
Figura 36: Controlador de Apoderado (Registrar Apoderado).	60
Figura 37: Controlador de Apoderado (Registrar Apoderado).	61
Figura 38: Controlador de Alumno (Registrar Alumno).	61
Figura 39: Controlador de Alumno (Registrar Alumno).	62
Figura 40: Controlador de Alumno (Registrar Alumno).	62
Figura 41: Controlador de Docente (Registrar Docente).	63
Figura 42: Controlador de Docente (Registrar Docente).	63
Figura 43: Controlador de Cursos (Registrar Cursos).	64
Figura 44: Controlador de Horario (Gestionar Horario).	65
Figura 45: Controlador de Año Académico (Gestionar Año Académico).	66
Figura 46: Controlador de Secciones(Grado).	67
Figura 47: Controlador de Secciones(Nivel).	68
Figura 48: Controlador de Secciones(Sección).	69
Figura 49: Controlador de Secciones(Sección).	69
Figura 50: Controlador ReportController (Generar reportes)	70
Figura 51: Controlador ReportController (Generar reportes)	70
Figura 52: Controlador ReportController (Generar reportes)	71
Figura 53: Vistas Registrar Matrícula (Create.blade.php)..	72
Figura 54: Vistas Registrar Matrícula (Create.blade.php).	72
Figura 55: Vistas Registrar Matrícula (Create.blade.php).	73
Figura 56: Vistas Registrar Matrícula (Create.blade.php).	74
Figura 57: Vistas Registrar Matrícula (Create.blade.php).	75
Figura 58: Vistas Registrar Matrícula (Create.blade.php).	75
Figura 59: Vistas Registrar Matrícula (edit.blade.php).	76
Figura 60: Vistas Registrar Matrícula (edit.blade.php).	76
Figura 61: Vistas Registrar Matrícula (index.blade.php).	77
Figura 62: Vistas Registrar Matrícula (index.blade.php).	77
Figura 63: Vistas Registrar Matrícula (modal.blade.php).	78
Figura 64: Vistas Registrar Matrícula (pdf.blade.php).	78
Figura 65: Vistas Registrar Matrícula (pdf.blade.php).	79
Figura 66: Vistas Registrar Matrícula (pdf.blade.php).	79
Figura 67: Vistas Registrar Matrícula (pdfb.blade.php).	80
Figura 68: Vistas Registrar Matrícula (pdfb.blade.php).	80

Figura 69: Vistas Registrar Matrícula (search.blade.php).	81
Figura 70: Vistas Registrar Matrícula (show.blade.php).	81
Figura 71: Vistas Registrar Matrícula (show.blade.php).	82
Figura 72: Vistas de Perfil Usuario (create.blade.php).	83
Figura 73: Vistas de Perfil Usuario (create.blade.php).	83
Figura 74: Vistas de Perfil Usuario (edit.blade.php).....	84
Figura 75: Vistas de Perfil Usuario (edit.blade.php).....	84
Figura 76: Vistas de Perfil Usuario (index.blade.php).....	85
Figura 77: Vistas de Perfil Usuario (index.blade.php).....	85
Figura 78: Vistas de Perfil Usuario (modal.blade.php).....	86
Figura 79: Vistas de Perfil Usuario (search.blade.php).	86
Figura 80: Vistas de Registrar Apoderado (create.blade.php).	87
Figura 81: Vistas de Registrar Apoderado (create.blade.php).	87
Figura 82: Vistas de Registrar Apoderado (edit.blade.php).....	88
Figura 83: Vistas de Registrar Apoderado (edit.blade.php).....	88
Figura 84: Vistas de Registrar Apoderado (index.blade.php).....	89
Figura 85: Vistas de Registrar Apoderado (index.blade.php).....	89
Figura 86: Vistas de Registrar Apoderado (modal.blade.php).....	90
Figura 87: Vistas de Registrar Apoderado (search.blade.php)	90
Figura 88: Vistas de Registrar Alumnos (create.blade.php).	91
Figura 89: Vistas de Registrar Alumnos (create.blade.php)	91
Figura 90: Vistas de Registrar Alumnos (edit.blade.php).....	92
Figura 91: Vistas de Registrar Alumnos (edit.blade.php).....	92
Figura 92: Vistas de Registrar Alumnos (index.blade.php).	93
Figura 93: Vistas de Registrar Alumnos (index.blade.php).	93
Figura 94: Vistas de Registrar Alumnos (modal.blade.php).....	94
Figura 95: Vistas de Reporte de Alumnos (pdf.blade.php).....	94
Figura 96: Vistas de Reporte de Alumnos (pdf.blade.php).....	95
Figura 97: Vistas de Registrar Alumnos (search.blade.php).....	95
Figura 98: Vistas de Registrar Docentes (create.blade.php).	96
Figura 99: Vistas de Registrar Docentes (create.blade.php).	96
Figura 100: Vistas de Registrar Docentes (edit.blade.php).....	97
Figura 101: Vistas de Registrar Docentes (edit.blade.php).....	97
Figura 102: Vistas de Registrar Docentes (index.blade.php).....	98
Figura 103: Vistas de Registrar Docentes (index.blade.php).....	98

Figura 104: Vistas de Registrar Docentes (modal.blade.php).....	99
Figura 105: Vistas de Registrar Docentes (search.blade.php).	99
Figura 106: Vistas de Registrar Cursos (create.blade.php).....	100
Figura 107: Vistas de Registrar Cursos (create.blade.php).....	100
Figura 108: Vistas de Registrar Cursos (edit.blade.php).	101
Figura 109: Vistas de Registrar Cursos (edit.blade.php).	101
Figura 110: Vistas de Registrar Cursos (index.blade.php).....	102
Figura 111: Vistas de Registrar Cursos (index.blade.php).....	102
Figura 112: Vistas de Registrar Cursos (modal.blade.php).	103
Figura 113: Vistas de Registrar Cursos (search.blade.php).	103
Figura 114: Vistas de Gestionar Horario (create.blade.php).....	104
Figura 115: Vistas de Gestionar Horario (create.blade.php).....	104
Figura 116: Vistas de Gestionar Horario (edit.blade.php).	105
Figura 117: Vistas de Gestionar Horario (edit.blade.php).	105
Figura 118: Vistas de Gestionar Horario (index.blade.php).	106
Figura 119: Vistas de Gestionar Horario (index.blade.php).	106
Figura 120: Vistas de Gestionar Horario (modal.blade.php).	107
Figura 121: Vistas de Gestionar Horario (search.blade.php).	107
Figura 122: Vistas de Gestionar Año Académico (create.blade.php).....	108
Figura 123: Vistas de Gestionar Año Académico (create.blade.php).....	108
Figura 124: Vistas de Gestionar Año Académico (edit.blade.php).....	109
Figura 125: Vistas de Gestionar Año Académico (edit.blade.php).....	109
Figura 126: Vistas de Gestionar Año Académico (index.blade.php).....	110
Figura 127: Vistas de Gestionar Año Académico (index.blade.php).....	110
Figura 128: Vistas de Gestionar Año Académico (modal.blade.php).....	111
Figura 129: Vistas de Gestionar Año Académico (search.blade.php).	111
Figura 130: Vistas de Gestionar Secciones Grados (create.blade.php).....	112
Figura 131: Vistas de Gestionar Secciones Grados (create.blade.php).....	112
Figura 132: Vistas de Gestionar Secciones Grados (edit.blade.php).	113
Figura 133: Vistas de Gestionar Secciones Grados (index.blade.php).	113
Figura 134: Vistas de Gestionar Secciones Grados (index.blade.php).	114
Figura 135: Vistas de Gestionar Secciones Grados (modal.blade.php).	114
Figura 136: Vistas de Gestionar Secciones Nivel (create.blade.php).	115
Figura 137: Vistas de Gestionar Secciones Nivel (create.blade.php).	115
Figura 138: Vistas de Gestionar Secciones Nivel (edit.blade.php).....	116

Figura 139: Vistas de Gestionar Secciones Nivel (index.blade.php).....	116
Figura 140: Vistas de Gestionar Secciones Nivel (index.blade.php).....	117
Figura 141: Vistas de Gestionar Secciones Nivel (modal.blade.php).....	117
Figura 142: Vistas de Gestionar Secciones Nivel (search.blade.php).	118
Figura 143: Vistas de Gestionar Secciones Sección (create.blade.php).	118
Figura 144: Vistas de Gestionar Secciones Sección (create.blade.php).	119
Figura 145: Vistas de Gestionar Secciones Sección (edit.blade.php).	119
Figura 146: Vistas de Gestionar Secciones Sección (edit.blade.php).	120
Figura 147: Vistas de Gestionar Secciones Sección (index.blade.php).	120
Figura 148: Vistas de Gestionar Secciones Sección (index.blade.php).	121
Figura 149: Vistas de Gestionar Secciones Sección (modal.blade.php).	121
Figura 150: Vistas de Gestionar Secciones Sección (search.blade.php).....	121
Figura 151: Vistas de Layouts (Diseños) (app.blade.php).	122
Figura 152: Vistas de Layouts (Diseños)(app.blade.php).	122
Figura 153: Vistas de Layouts (Diseños) (home.blade.php).	123
Figura 154: Vistas de Layouts (Diseños) (home.blade.php).	123
Figura 155: Vistas de Layouts (Diseños) (home.blade.php).	124
Figura 156: Vistas de Layouts (Diseños) (home.blade.php).	124
Figura 157: Vistas de Layouts (Diseños) (welcome.blade.php).	125
Figura 158: Vistas de Layouts (Diseños) (welcome.blade.php).	125
Figura 159: Vistas de Layouts (Diseños) (welcome.blade.php).	126
Figura 160: Vistas de Layouts (Diseños) (welcome.blade.php).	126
Figura 161: Vistas de Layouts (Diseños) (welcome.blade.php).	127
Figura 162: Routes (api.php).....	128
Figura 163: Routes (api.php).....	128
Figura 164: Routes (channels.php).....	128
Figura 165: Routes (console.php).	129
Figura 166: Routes (web.php).....	129
Figura 167: Routes (web.php).....	130
Figura 168: Barra de navegación para (1) Pc y (2) Teléfono.....	130
Figura 169: Inicio.....	131
Figura 170: Inicio.....	131
Figura 171: Inicio.....	131
Figura 172: Acerca de	132
Figura 173: Acerca de	132

Figura 174: Contacto	133
Figura 175: Contacto	133
Figura 176: Contacto	133
Figura 177: Iniciar sesión	134
Figura 178: Panel Principal	134
Figura 179: Formulario de Registrar Matrícula	135
Figura 180: Registrar Matrícula	135
Figura 181: Formulario Perfil Usuario	136
Figura 182: Perfil Usuario	136
Figura 183: Formulario de Registrar Apoderado	137
Figura 184: Listado de Apoderados	137
Figura 185: Formulario de Registrar Alumno	138
Figura 186: Listado de Alumnos registrados	138
Figura 187: Reporte de listado de alumnos PDF.	139
Figura 188: Formulario de Registrar Docente	139
Figura 189: Listado de Docentes	140
Figura 190: Formulario de Registrar Cursos	140
Figura 191: Listado de Cursos	141
Figura 192: Formulario de Gestionar Horario	141
Figura 193: Listado de Horarios	142
Figura 194: Formulario de Gestionar Año Académico	142
Figura 195: Listado de Gestionar Año Académico	143
Figura 196: Formulario de Gestionar Secciones (Registrar Grado)	143
Figura 197: Listado de Gestionar Secciones (Registrar Grado)	144
Figura 198: Formulario de Gestionar Secciones (Registrar Nivel)	144
Figura 199: Listado de Gestionar Secciones (Registrar Nivel)	145
Figura 200: Formulario de Gestionar Secciones (Registrar Sección)	145
Figura 201: Listado de Gestionar Secciones (Registrar Sección)	146
Figura 202: Reportes por día de alumnos matriculados por día	146
Figura 203: PDF de reportes por día de alumnos matriculados.	147
Figura 204: Reportes por rango de fecha de alumnos matriculados.	147
Figura 205: PDF de reportes por rango de fecha de alumnos matriculados.	148
Figura 206: Buscar por grado, sección y nivel de alumnos matriculados	148
Figura 207: Reportes por grado, sección y nivel de alumnos matriculados	149

Figura 208: Descargar PDF de reportes por grado, sección y nivel de alumnos matriculados.	149
Figura 209: Buscar cantidad de secciones registradas por alumno.	150
Figura 210: Listado de cantidad de secciones registradas por alumno.	150
Figura 211: Capturas de prueba de proyecto guardado en hosting y dominio en cPanel.	151
Figura 212: Administrador de archivos cPanel.	151
Figura 213: Información general.	152
Figura 214: Link de ingreso (subdominio).	152
Figura 215: Tabla de chi-cuadrado, valor tabulado.	165

GRÁFICOS

Gráfico 1: ¿El sistema académico web está conforme con respecto a funcionalidad?	153
Gráfico 2: ¿El sistema académico web puede adaptarse de un entorno a otro?	154
Gráfico 3: ¿El sistema académico web es fácil de entender y usar?	155
Gráfico 4: ¿El sistema académico web requiere de mantenimiento constante?	155
Gráfico 5: ¿Considera usted que el sistema académico web realiza un procesamiento eficiente en la ejecución de tareas?.....	156
Gráfico 6: ¿El sistema académico web es eficiente con respecto a la atención de personas en un periodo de tiempo?	157
Gráfico 7: ¿Al llenar y guardar datos en el sistema no se tarda en responder la solicitud?	157
Gráfico 8: ¿El sistema académico web es seguro y fácil de acceder a los datos?	158
Gráfico 9: ¿Considera usted que el sistema académico web logró resolver los problemas de tiempo en las matrículas en dicha Institución?	159
Gráfico 10: ¿Considera usted que el sistema web de la Institución Educativa es eficiente a la atención de personas en un periodo de tiempo?.....	160
Gráfico 11: ¿El sistema académico web es de calidad al tiempo de respuesta de solicitudes?	161
Gráfico 12: ¿Considera que el sitio web es fácil de acceder?	162

1. INTRODUCCION

1.1.Planteamiento del problema

Las tecnologías digitales han crecido exponencialmente y su uso se ha globalizado. La conectividad ubicua y continua llega a gran parte de la humanidad gracias a la masificación del uso de teléfonos inteligentes y al consiguiente acceso a la información, a las redes sociales y al entretenimiento audiovisual. La aceleración del progreso técnico en el universo digital ha vuelto cotidiana el empleo de dispositivos y aplicaciones que usan la computación en la nube, la analítica de grandes datos, las cadenas de bloques o la inteligencia artificial. La revolución tecnológica, aunada al cambio en las estrategias de las empresas líderes en el uso de las tecnologías digitales, ha llevado al auge de las plataformas globales, dando lugar a una excesiva concentración de poder económico y político en no más de una veintena de corporaciones de dos o tres potencias mundiales, un conjunto demasiado pequeño de empresas cuyo valor de mercado supera o se acerca a un billón de dólares. (Rovira, 2022)

El “Colegio Túpac Amaru II Huarango” actualmente cuenta con numerosos expedientes de matrículas, debido a lo peculiar de su enseñanza y formación a nivel primario y secundario, influyendo de manera directa en el deficiente desarrollo de las inscripciones, por cuanto el proceso de matrícula se maneja de una manera manual e insegura, ya que la información es almacenada en carpetas, en sus respectivos archivos, retrasando los procesos de búsqueda de cada expediente de los estudiantes, y a la vez, poniendo en peligro la información confidencial de cada estudiante, ya que puede ser plagiada o dañada por desastres naturales u otros sucesos. Dicha Institución no cuenta con

un sistema de información que pueda realizar este proceso, por tanto, se cree conveniente que se realice la implementación de un sistema de información.

Teniendo en cuenta que el sistema a implementar permitirá a personal administrativo y docentes, acceder de una manera rápida y eficaz a la base de datos de la institución donde en la misma mostrará la información del alumno, a que grado va a realizar su matrícula, si tiene deuda, etc. En caso el alumno sea nuevo, el sistema mostrará un formulario a llenar, el cual optimiza tiempo, ya que no se estará buscando el formulario en físico o sacar copia, o imprimir, ya que el sistema mostrará de una manera rápida.

1.2. Formulación del problema

¿Cómo influye el sistema académico web para mejorar el proceso de matrículas en la Institución Educativa Túpac Amaru II N°16498-Huarango-San Ignacio-Cajamarca?

1.3. Justificación

En la I.E. Tupac Amaru II se brinda servicios de educación tanto nivel primario y secundario, es por ello que al iniciar el año se lleva a cabo el proceso de matrículas y los estudiantes con sus padres de familia se acercan para realizar sus tareas de matrícula, justificamos que la institución se encuentra en un proceso de mejora es decir de optimización, lo que importa es que los datos de los estudiantes sean guardados de manera segura y confiable, para así llevar a

cabo el orden o un control de ello para luego al buscar los datos sean fáciles de encontrar.

Nuestra investigación es realizada con la forma de mejorar y automatizar los procesos de matrículas, este sistema va a dar resultados óptimos como el de acceder de manera fácil, manejar correspondiente, sobre todo reducir el tiempo de matrícula física. Esto ayudará a que los sistemas de información son fáciles de usar y ayudan a mejorar la forma manual y digital.

1.4. Objetivos de la investigación

1.4.1. Objetivo General

- ✓ Mejorar el proceso de matrículas en la “I.E. Túpac Amaru II N° 16498, Huarango – San Ignacio- Cajamarca, mediante la implementación de un sistema académico web.

1.4.2. Objetivos específicos

- ✓ Desarrollar el sistema académico web según los requerimientos.
- ✓ Validar el sistema académico web mediante la aprobación de expertos.

1.5. Hipótesis de la investigación

1.5.1. Hipótesis

La influencia del sistema académico web mejorará a optimizar el tiempo de matrículas de la institución educativa Tupac Amaru II N° 16498, Huarango-San Ignacio-Cajamarca.

1.5.2. Operacionalización de variables

Variable Independiente

a) Variable 1:

Sistema académico web

Se conceptúa como sistema académico web a un software informático desarrollado bajo la tecnología web, realizando el proceso académico como es el registro de matrícula, gestión de notas, y la generación de reportes en una institución educativa. (Chaoca, 2005)

Variable Dependiente

b) Variable 2:

Mejora en el proceso de matriculas

Es el acto académico administrativo por el cual el aspirante admitido o el Estudiante antiguo, adquiere o renueva voluntariamente la calidad de estudiante y se obliga a cumplir con los estatutos, reglamentos y demás normas de la Institución, y se formaliza con la firma del documento respectivo. (Gastelboldo, 2014)

Tabla 1: *Matriz de Operacionalización de Variables*

Variable	Definición	Dimensiones	Indicador (es)	Instrumento (s)
Independiente Sistema académico web	Se conceptúa como sistema académico web a un software informático desarrollado bajo la tecnología web, realizando el proceso académico como es el registro de matrícula, gestión de notas, y la generación de reportes en una institución educativa. (Chaoca, 2005)	Funcionalidad Portabilidad Usabilidad Mantenibilidad	Conformidad de la funcionalidad Porcentaje de aceptación de facilidad de adaptación Porcentaje de satisfacción Calificación de calidad de servicio	Cuestionario Escala de Likert
Dependiente Mejora en el proceso de registro de matriculas	Es el acto académico administrativo por el cual el aspirante admitido o el Estudiante antiguo, adquiere o renueva voluntariamente la calidad de estudiante y se obliga a cumplir con los estatutos, reglamentos y demás normas de la Institución, y se formaliza con la firma del documento respectivo (Gastelboldo, 2014).	Tiempo Eficiencia Relevancia Organizacional	Tiempo promedio invertido para el procesamiento y ejecución de tareas. Número de personas atendidas en un periodo de tiempo Tiempo de respuesta Seguridad en el almacenamiento de datos y facilidad de acceso	Cuestionario Escala de Likert

2. MARCO TEÓRICO

2.1. Antecedentes que sustentan la investigación

A nivel internacional Diaz Fernández realizó una investigación para optar el título profesional de ingeniero de sistemas, en el tema Sistema de información para el proceso de matrículas, control de asistencias y gestión de notas académicas en la institución educativa departamental integrada Antonio Nariño del municipio de Apulo, teniendo como objetivo general Desarrollar e implementar un sistema de información donde se lleven a cabo los procesos de las matrículas, control de asistencia y gestión de notas de los estudiantes para mejorar los procesos administrativos de la institución educativa Antonio Nariño del municipio de Apulo departamento de Cundinamarca, Nos dice en una de sus conclusiones que “la investigación realizada, determina que hay factores que influyen en el buen procesamiento de la información, pero como se está manejando de manera física, los procesos académicos tardan mucho en realizarse. por lo tanto, no se puede negar la hipótesis del presente proyecto y la viabilidad al momento de generar una solución con un sistema de información propio ya que este mejoraría las condiciones administrativas de los entes que rigen la institución”. (Diaz Fernandez, 2021, pág. 119).

Por otro lado, Guevara Ascoy en su investigación realizada para optar por el título profesional de ingeniero teleinformático en el tema Diseño e implementación de un sistema de matrícula y agenda virtual para mejorar la eficiencia administrativa de la i.e. 10824 “san Nicolás”, Chiclayo 202, nos dice en una de sus conclusiones que “El sistema de matrícula y agenda virtual es un software de calidad, accesibilidad segura y manejo sencillo lo cual se logró un impacto positivo en los administrativos de la institución educativa 10824 “San

Nicolas” ya que con el sistema se ahorran el tiempo de volver a solicitar la información al momento del proceso de matrícula”. (Guevara Ascoy , 2021, pág. 63).

Según Moyán y Palomino (2021) en su tesis realizada para optar por el título profesional de ingeniero de sistemas, presentado a la Universidad Privada Antonio Guillermo Urrelo, con el tema de Implementación de un sitio web como medio de difusión de contenidos de la Institución Educativa N°028 “Santa Elena”-Cajamarca-2021, tuvo como objetivo general Implementar un sitio web como medio de difusión de contenidos de la Institución Educativa N°028 “Santa Elena”-Cajamarca-2021, nos dice en una de sus conclusiones La implementación del sitio web es significativamente influyente ya que permite mostrar información más organizada, muestra contenido multimedia sin límite de almacenamiento, se accede desde cualquier dispositivo lo que facilita su acceso, logrando de esta manera que la Institución Educativa N°028 “Santa Elena” disponga de una herramienta tecnológica rápida, confiable y eficaz”. (Moyán Oblitas & Palomino Correa, 2021, pág. 134).

En cambio, Osorio Álvarez, en su tesis realizada para optar por el título profesional de ingeniero de sistemas e informática, presentado a la universidad de Ciencias y Humanidades de Lima, teniendo por tema Diseño e implementación de un sistema de matrícula web usando software libre en el centro educativo “España”, distrito – Breña 2013, teniendo como objetivo general, Diseñar e implementar un sistema de matrícula web usando software libre en el centro educativo España del distrito de Breña, nos dice en una de sus conclusiones que, “Se logró implementar el módulo de reportes con el cual

el uso del sistema de matrícula web permite una mayor viabilidad, usabilidad y seguridad en la información, así como el uso de los recursos (hardware, software, trabajadores) en el centro educativo. De igual manera, se optimiza la toma de decisiones ya que cuenta con reportes personalizados”. (Osorio Álvarez, 2016, pág. 126).

A nivel local tenemos a, Nava Alarcón, con su tesis presentada a la universidad nacional de Cajamarca, para optar por el título profesional de ingeniero de sistemas, titulada, Mejoramiento del proceso de control de pagos y matrícula de la institución educativa privada Ramón Castilla a través de un sistema de información desktop, nos dice en una de sus conclusiones que “Los requerimientos de información satisfechos mensualmente con el desarrollo e implementación del sistema de información desktop “Software a Medida v1.0 – Ramón Castilla” aumento en un 20% de 11 - 20 a 21 - 30 y 60% de 21 - 30 a más de 31; por lo tanto; la satisfacción del usuario ha mejorado sustancialmente en relación al manejo de información de matrículas y pagos; exactitud de la información y la satisfacción propiamente dicha de los medios utilizados para la obtención de la información de 60% a 100%”. (Nava Alarcón , 2018, pág. 81).

Por otro lado Ortiz Garay, con su tesis presentada a la universidad nacional de Cajamarca, para optar por el título profesional de ingeniero de sistemas, titulada, Influencia de un sistema de información web en la gestión académica de la institución educativa particular maría de Nazaret de Cajamarca, nos dice en una de sus conclusiones que “En esta investigación no

se han llegado a automatizar todos los procesos académicos, ya que se presentaron ciertas limitantes, como el hecho de que no se podía implementar una plataforma de pagos ya que requería costos adicionales, que la institución no creyó conveniente realizar, por cierta desconfianza en cuanto a la seguridad de los medios de pago. Esto solo demuestra que aún existe cierta resistencia a usar las tecnologías de la información, sobre todo, cuando el riesgo es un mayor y las pérdidas que se pueden presentar son económicas”. (Ortiz Garay, 2020, pág. 79).

2.2. Bases conceptuales

2.2.1. Diseño

Se considera diseño a las actividades que son creativas y que tienen por fin conseguir una proyección de objetos que son útiles. Diseño es la forma que se traza para conseguir alguna cosa.

2.2.2. Implementar

Implementar es llevar a cabo en funcionamiento una idea planificada en un espacio determinado.

2.2.3. Bases de datos

Programa capaz de realizar diferentes actividades de una forma muy organizada posible, es la agrupación de datos de información agrupada.

2.2.4. Sistemas web

Son las diferentes aplicaciones de software, estas son utilizadas por las diferentes personas en la cual acceden a internet mediante un navegador.

2.2.5. Implementación de software

La implementación de un software son las etapas de elaboración ya concluidas al modificar un código fuente, también son las formas de adaptar nuevos requerimientos a un software.

2.2.6. Servidor de aplicación web

Un servidor de aplicaciones web son las diferentes redes de computadoras que deben ejecutar distintas aplicaciones en la cual una persona accede a ello para beneficiarse de información que está buscando.

2.2.7. Modelo cliente/servidor

El modelo cliente-servidor es quien a través de una computadora busca respuestas a través de un servidor brindado y por lo general se conecta a muchos servidores en línea a la vez.

Figura 1. Ejemplo de una estructura simple cliente servidor



2.2.8. Metodología ágil

Entendemos por metodologías ágiles al grupo de técnicas, métodos que se aplican en partes de trabajos con el fin u objetivo de que un proyecto sea más efectivo. A cada parte terminada se lleva a cabo por avances cada cierto tiempo. Ayuda mucho a las tareas de investigación que son de corto tiempo.

2.2.8.1. Proceso Unificado de Rational

Es una metodología que se utiliza para investigaciones y es la más utilizada constituida por una análisis, implementación y entrega de documentación de sistemas especialmente en objetos. Está compuesta por fases que son Inicio, elaboración, construcción y transición.

2.2.8.2.Método en cascada

Se refiere a un conjunto de pasos a realizar de forma ordenada y organizacionalmente, cumpliendo cada paso para seguir con el siguiente, también es el método de gestión de proyectos. Se dividen en fases empezando por los requisitos, diseño e implementación, pruebas, implantación y mantenimiento.

2.2.9. Metodologías XP

La metodología XP o Programación Extrema es una metodología ágil y flexible utilizada para la gestión de proyectos.

Extreme Programming nos brinda la facilidad de ciclos cortos, en la cual influye mucho la comunicación entre los miembros de equipo, control de avances, realizar un listado de tareas. Requiere retroalimentación continua entre cliente y el equipo de desarrollo.

2.2.9.1. Fases de la metodología XP

a). Planificación

- Historias de usuarios
- Plan de entregas
- Plan de iteraciones
- Reuniones de seguimiento

b). Diseño

- Caracterizarse por simplicidad
- Regirse a especificaciones necesarias

c). Desarrollo

- Uso de estándares
- Programaciones dirigidas por las pruebas

d). Pruebas

- Pruebas unitarias
- Detectar y corregir posibles errores
- Pruebas de aceptación

2.2.10. Proceso de matriculas

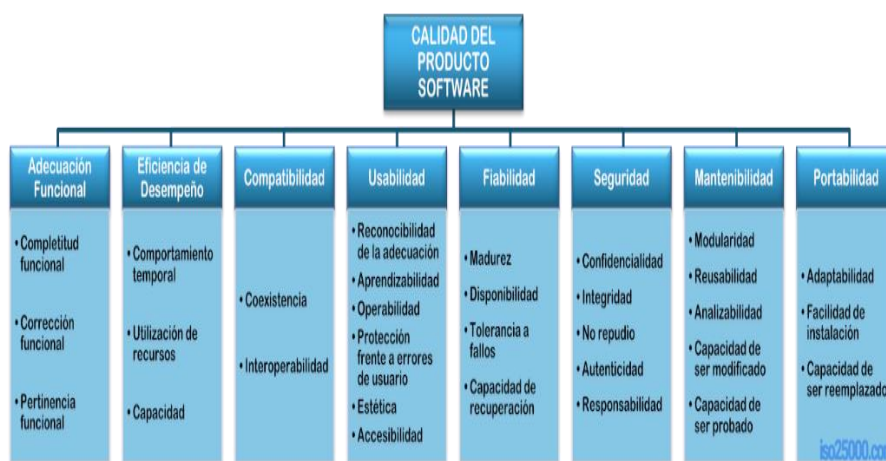
Cuando hablamos de proceso de matrículas nos estamos refiriendo al conjunto de pasos o procedimientos a realizar para llevar a cabo una matriculas de un estudiante en una institución pública o privada.

2.2.11. ISO 25000 – 25010

Son normas que hacen que nuestras investigaciones sumen puntos a favor y mostrar resultados mejorados con respaldos de estas.

La familia ISO/IEC 25000 son las diferentes características que se deben de cumplir en las diferentes investigaciones de proyectos con el fin de que estas se evalúen al terminar un proyecto.

El modelo de calidad del producto definido por la ISO/IEC 25010 se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente figura: (Iso, 2022)



2.2.12. Laragon

Laragon es una bandeja de almacenamiento de datos para PHP, Node.js, Python, Java, Go, Ruby. Este entorno es rápido, fácil de utilizar y sobre todo se entiende para cualquier usuario que pueda ingresar o tener el aplicativo.

En Laragon podemos crear diferentes aplicaciones web actuales. La libertad es una característica muy importante ya que podemos crear de una aplicación simple hasta una grande.

2.2.13. Php

PHP es un lenguaje de programación en el cual es de código abierto especialmente para desarrollos web. PHP es fácil de almacenar en distintos documentos, es versátil, rápido y seguro.

2.2.14. Admin

Admin se refiere a la persona que controla, vela el procesamiento de actividades que realizan terceras personas, estas las lleva a cabo mediante una computadora.

Los administradores son capaces de gestionar los errores que se comenten en un sistema cualquiera.

2.2.15. Sublime text

Sublime text es un software editor de códigos de desarrollo de aplicaciones. Este editor de texto subraya en caso el código contenga un error y avisa al programador que está cometiendo una infracción. Esta es fácil de entender, la pueden utilizar diferentes personas. Las utilizan en la actualidad: Java, Python, Perl, HTML, JavaScript, CSS, HTML, XML, PHP, C, C++, etc.

2.2.16. Microsoft Excel

Microsoft Excel está en el paquete de Microsoft, esta es una pieza de de ello, la cual es un programa para analizar diferentes datos. Excel nos ayuda porque se aplican diferentes formulas la cual nos beneficia

a calcular grandes cantidades de interrogantes. Es usado en una empresa pequeña para llevar a cabo su contabilidad de datos y así evitar la pérdida de información.

2.2.17. Escala de Likert

La escala de Likert es una forma de calificar una cierta o conjunto de preguntas con el fin de conocer el nivel de que si está de acuerdo o en desacuerdo con alguna actividad o investigación realizada. Sirve como instrumento de recolección de datos de una actividad terminada, estos datos pueden ser cuantitativos es decir contables o mejor dicho números.

3. METODOLOGÍA DE LA INVESTIGACIÓN

3.1. Unidad de análisis, universo y muestra

3.1.1. Unidad de análisis

La unidad de análisis está representada principalmente por los expertos (3), personal administrativo (3) y padres de familia (54) de la Institución Túpac Amaru II N° 16498 Huarango – San Ignacio- Cajamarca.

3.1.2. Población

La población está dada por 1 director, 2 personas administrativas y padres de familia (150) de la Institución Túpac Amaru II N° 16498, Huarango – San Ignacio- Cajamarca, consideramos a los expertos (3) que llevaran a cabo la verificación y conformidad.

3.1.3. Muestra

En nuestra investigación se aplicó un muestreo no probabilístico y probabilístico por conveniencia para el director (1), Personal administrativo (2) de la Institución Túpac Amaru II N° 16498, Huarango – San Ignacio- Cajamarca y los 3 expertos.

El muestreo por conveniencia es una técnica de muestreo no probabilístico y no aleatorio utilizada para crear muestras de acuerdo a la facilidad de acceso, la disponibilidad de las personas de formar parte de la muestra, en un intervalo de tiempo dado o cualquier otra especificación práctica de un elemento particular. (QuestionPro, 2018)

Respecto a la población de los padres que nos apoyaron se aplicó un muestreo probabilístico por el cálculo estadístico, utilizando la siguiente fórmula, (para poblaciones finitas).

$$n = \frac{N \times z_{\alpha}^2 \times p \times q}{d^2 \times (N - 1) + z_{\alpha}^2 \times p \times q}$$

Donde:

- N= Total de la población.
- $Z_{\alpha} = 95\%$
- p= Proporción esperada (5%=0.05).
- $q=1 - p = (1-0.005=0.95)$.
- d= precisión (5%).

Tabla 2: Encargados de evaluar el sistema.

UNIDAD DE ANALISIS	POBLACIÓN	TAMAÑO DE LA MUESTRA (n)
Personal administrativo	3	3
Expertos	3	3
Padres de familia	150	54
TOTAL	156	60

Nota. Fuente. Elaboración propia.

El tamaño del cálculo de la muestra es de 54 con un margen de error del 0.05, por lo que se realizó encuestas a 54 padres de familia de la institución educativa Tupac Amaru II N°16498.

En total, personal administrativo (3) y padres de familia (54) de la Institución educativa Tupac Amaru II N°16498 y los expertos que son (3), estos suman 60 integrantes.

3.2. Métodos de investigación

3.2.1. Tipo de investigación

La presente investigación es de tipo aplicada, ya que se aplicaron los conocimientos para desarrollar el sistema académico web, el cual que ayudará a resolver los problemas de matrículas de la Institución Túpac Amaru II N° 16498, Huarango-san Ignacio -Cajamarca.

Para Murillo (2008), la investigación aplicada recibe el nombre de “investigación práctica o empírica”, que se caracteriza porque busca la aplicación o utilización de los conocimientos adquiridos, a la vez que se adquieren otros, después de implementar y sistematizar la práctica basada en investigación. El uso del conocimiento y los

resultados de investigación que da como resultado una forma rigurosa, organizada y sistemática de conocer la realidad.

3.2.2. Diseño de la investigación

La presente investigación tiene un diseño cuasiexperimental, ya que está compuesta por una variable independiente Sistema Académico web, para observar su influencia en la variable dependiente, del proceso de matrículas.

Al igual que los diseños experimentales, los diseños de investigación cuasiexperimentales contrastan hipótesis causales. Tanto en los diseños experimentales (ensayos controlados aleatorios) como en los cuasiexperimentales, el programa o política se considera como una «intervención» en la que se comprueba en qué medida un tratamiento—incluidos los elementos del programa o la política evaluados—logra sus objetivos, de acuerdo a las mediciones de un conjunto preestablecido de indicadores.

3.3. Enfoque de la investigación

El enfoque que se utilizó fue cuantitativo, debido a que las variables son susceptibles de medición y enfatizó y se realizó un procedimiento estadístico denominado Chi cuadrado para demostrar la hipótesis planteada.

El diseño de la investigación cuantitativa constituye el método experimental común de la mayoría de las disciplinas científicas. El objetivo de una investigación cuantitativa es adquirir conocimientos fundamentales y la elección del modelo más adecuado que nos permita conocer la realidad de una manera más imparcial, ya que se recogen y analizan los datos a través de los conceptos y variables medibles. La investigación cuantitativa es una forma estructurada de recopilar y analizar datos obtenidos de distintas fuentes, lo que implica el uso de herramientas informáticas, estadísticas, y matemáticas para obtener resultados. Es concluyente en su propósito ya que trata de cuantificar el problema y entender qué tan generalizado está mediante la búsqueda de resultados proyectables a una población mayor. Esta hipótesis debe ser demostrable por medios matemáticos y estadísticos, constituyéndose en la base alrededor de la cual se diseña todo el experimento. (Alan Neill, David; Cortez Suárez, Liliana;, 2017)

3.4. Técnicas e instrumentos de recolección de datos

3.4.1. Técnicas de investigación

3.4.1.1. Encuestas

Para este instrumento se formulará el cuestionario de preguntas para posteriormente ser aplicadas a los padres de familia y personal administrativo de la Institución Educativa Túpac Amaru II N°16498, Huarango – San Ignacio- Cajamarca.

Las encuestas son un método de investigación y recopilación de datos utilizados para obtener información de personas sobre diversos temas. Las encuestas tienen una variedad de propósitos y se pueden llevar a cabo de muchas maneras dependiendo de la metodología elegida y los objetivos que se deseen alcanzar.

Los datos suelen obtenerse mediante el uso de procedimientos estandarizados, esto con la finalidad de que cada persona encuestada responda las preguntas en una igualdad de condiciones para evitar opiniones sesgadas que pudieran influir en el resultado de la investigación o estudio. (questionpro, 2023)

3.4.1.2. Entrevistas

En esta parte se aplicó dos (02) cuestionarios, el primero para recolectar la información si el sistema académico web cumple con los requisitos, y el segundo cuestionario sirvió para recolectar la información relacionada a las matriculas por parte de los usuarios, todo esto después de la implementación del sistema antes mencionado.

La entrevista, es la comunicación interpersonal establecida entre el investigador y el sujeto de estudio a fin de obtener respuestas verbales a los interrogantes planteados sobre el problema propuesto. Se considera que este método es más eficaz que el cuestionario, ya que permite obtener una información más completa.

Se podrá definir que la entrevista consiste en obtención de información oral de parte de una persona (entrevistado) lograda por el entrevistador directamente, en una situación de cara a cara, a veces la información no se transmite en un solo sentido, sino en ambos, por lo tanto, una entrevista es una conversación entre el investigador y una persona que responde a preguntas orientadas a obtener información exigida por los objetivos específicos de un estudio. (Galán Amador, 2009)

3.4.2. Instrumentos

Los instrumentos de investigación son necesarios en los tipos de recopilación de datos y tecnología más utilizadas en la investigación científica. En cuanto a las herramientas, según Hernández (1991), señala que las adecuadas son aquellas que se utilizan para registrar datos observables, de modo que puedan representar verdaderamente la definición de variable que tienen los investigadores.

3.4.2.1. Cuestionarios

El cuestionario es un procedimiento considerado clásico en las ciencias sociales para la obtención y registro de datos. Su versatilidad permite utilizarlo como instrumento de investigación y como instrumento de evaluación de personas, procesos y programas de formación. Es una técnica de evaluación que puede abarcar aspectos cuantitativos y cualitativos. Su característica

singular radica en que, para registrar la información solicitada a los mismos sujetos, ésta tiene lugar de una forma menos profunda e impersonal, que el "cara a cara" de la entrevista. Al mismo tiempo, permite consultar a una población amplia de una manera rápida y económica. (García Muñoz, 2003)

3.4.3. Técnicas de análisis de datos

Para el análisis de datos se realizó mediante la estadística descriptiva usando las herramientas de Excel y IBM SPSS, aplicando a la información recolectada mediante las fórmulas de tendencia central entre las cuales tenemos a la moda, mediana y media.

3.4.4. Validación de instrumentos

Se elaboraron cuestionarios con el fin de resolver la validación del sistema académico web, así como para evaluar la confiabilidad de la información en la herramienta tecnológica planteada, estos cuestionarios los realizaron los expertos, se necesitó la colaboración de tres expertos para que lleven a cabo los análisis de cada pregunta que se planteó, estas preguntas están en el cuestionario, y también los objetivos e hipótesis. Para luego aplicarlos y obtener resultados que posteriormente serán procesados.

3.4.5. Herramienta estadística

En nuestra investigación se aplicará el estadístico chi-cuadrado “ X^2 ” con la finalidad de demostrar que se acepta o se rechaza la hipótesis.

El estadístico ji-cuadrado (o chi cuadrado), que tiene distribución de probabilidad del mismo nombre, sirve para someter a prueba hipótesis referidas a distribuciones de frecuencias. En términos generales, esta prueba contrasta frecuencias observadas con las frecuencias esperadas de acuerdo con la hipótesis nula. (Quevedo Ricardi, 2011)

Se utilizará Microsoft Excel para poder darle interpretación estadísticamente los resultados de las encuestas, esto permitirá contrastar la hipótesis que se planteó anteriormente mediante los resultados obtenidos de las encuestas.

4. IMPLEMENTACIÓN

Durante la implementación de nuestro proyecto sistema académico web y su influencia para la mejora de matrículas en la institución Educativa Túpac Amaru II Huarango – San Ignacio – Cajamarca, realizado con la metodología XP.

4.1.RECURSOS PARA LA IMPLEMENTACIÓN

En los recursos para la implementación del sistema académico web se utilizaron diferentes medios para llevar a cabo la investigación, los gastos fueron puestos a disponibilidad de los bachilleres con el fin de cumplir las diferentes tareas.

Tabla 3: Recursos necesarios

Descripción	Cantidad	Precio (S/.)
Laptop: Asus	1	Dispuesta por el bachiller
Laptop: Asus	1	Dispuesta por el bachiller
Disco extraíble	1	Dispuesta por el bachiller
Software(Laragon)	-	Disponible
Dominio y Hosting	1 año	\$ 50 USD
TOTAL:		\$ 50 USD

Nota. Fuente: elaboración propia.

4.2. Planificación del proyecto

✓ Análisis de requerimientos

Tabla 4: Historia de Usuario N.º 01

Historia de Usuario	
Número: 01	Usuario: Todos
Nombre Historia	
INICIO	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: MEDIA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 1
Programadores responsables:	
Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: la página de inicio cuenta con el nombre de la institución, su insignia, algunas fotos de la propia escuela, noticias e historias.	
Observaciones: Ninguna.	

Tabla 5: *Historia de Usuario N.º 02*

Historia de Usuario	
Número: 02	Usuario: Todos
Nombre Historia	
ACERCA DE	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: MEDIA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 1
Programadores responsables:	
Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: acerca de, mostrará un poco de información de la escuela y su personal educativo de tanto nivel primario, secundario y personal directivo y administrativo.	
Observaciones: Ninguna.	

Tabla 6: *Historia de Usuario N.º 03*

Historia de Usuario	
Número: 03	Usuario: Todos
Nombre Historia	
CONTACTO	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: MEDIA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 1
Programadores responsables:	
Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: el usuario podrá tomar apuntes sobre la información del personal directivo y administrativo, tanto como docentes de nivel primaria, secundaria.	
Observaciones: Ninguna.	

Tabla 7: *Historia de Usuario N.º 04*

Historia de Usuario	
Número: 04	Usuario: Todos
Nombre Historia INICIAR SESIÓN	
Prioridad en negocio: MEDIA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: MEDIA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 1
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: al abrir iniciar sesión, nos enviará a otra ventana donde se visualizará un login con un correo y una contraseña.	
Observaciones: Ninguna.	

Tabla 8: *Historia de Usuario N.º 05*

Historia de Usuario	
Número: 05	Usuario: Todos
Nombre Historia SOBRE NOSOTROS	
Prioridad en negocio: MEDIA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: MEDIA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 1
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: se podrá visualizar una información pequeña de la institución educativa.	
Observaciones: Ninguna.	

Tabla 9: *Historia de Usuario N.º 06*

Historia de Usuario	
Número: 06	Usuario: Todos
Nombre Historia CURSOS	
Prioridad en negocio: MEDIA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 1
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: aparecerán todos los cursos disponibles por la institución educativa.	
Observaciones: Ninguna.	

Tabla 10: *Historia de Usuario N.º 07*

Historia de Usuario	
Número: 07	Usuario: Todos
Nombre Historia IMÁGENES	
Prioridad en negocio: MEDIA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 1
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: la institución cuenta con distintas imágenes disponibles con una descripción de ellas.	
Observaciones: Ninguna.	

Tabla 11: *Historia de Usuario N.º 08*

Historia de Usuario	
Número: 08	Usuario: Administrador
Nombre Historia	
LOGIN	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 2
Programadores responsables:	
Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: el login se abrirá al ingresar un usuario y contraseña correcta y si no, saldrá un error.	
Observaciones: se deberá tener un usuario y contraseña correcta para poder ingresar al sistema.	

Tabla 12: *Historia de Usuario N.º 09*

Historia de Usuario	
Número: 09	Usuario: Administrador
Nombre Historia	
PANEL PRINCIPAL (INICIO)	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 2
Programadores responsables:	
Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: en el panel principal se visualizará la gestión de cada cuadro como es el de matrículas, usuarios, apoderados, alumnos, docentes, cursos, horarios, año académico, grados, niveles, sección.	
Observaciones: Ninguna.	

Tabla 13: *Historia de Usuario N.º 10*

Historia de Usuario	
Número: 10	Usuario: Administrador
Nombre Historia REGISTRAR MATRÍCULA	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 2
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: al hacer clic en nueva matricula, se podrá matricular al alumno cuando ya se haya registrado al alumno, también el curso, año, horario, sección, grado, nivel en los siguientes botones. También se podrá poner un respectivo costo.	
Observaciones: para realizar una matrícula se deberá tener datos llenos en los siguientes botones.	

Tabla 14: *Historia de Usuario N.º 11*

Historia de Usuario	
Número: 11	Usuario: Administrador
Nombre Historia PERFIL DE USUARIO	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 2
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: en el perfil de usuario se mostrará el listado de usuarios con su nombre, email y el tipo de empleado, estos se podrán editar y eliminar si es necesario, también se podrá agregar un nuevo usuario para que pueda ingresar en el login.	
Observaciones: Ninguna.	

Tabla 15: *Historia de Usuario N.º 12*

Historia de Usuario	
Número: 12	Usuario: Administrador
Nombre Historia	
REGISTRAR APODERADO	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 2
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: en el perfil registro de apoderado estará vacío, en el cual habrá una opción donde se podrá registrar un nuevo apoderado con su nombre, sexo, DNI, dirección, celular, una imagen, una vez guardado se podrá editar y eliminar si es necesario.	
Observaciones: Ninguna.	

Tabla 16: *Historia de Usuario N.º 13*

Historia de Usuario	
Número: 13	Usuario: Administrador
Nombre Historia	
REGISTRAR ALUMNOS	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 3
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: en el registro de alumnos se mostrará una opción de nuevo alumno en el cual se llenará datos como es el de nombres, email, DNI, fecha de nacimiento, sexo, dirección, celular, imagen y automáticamente el apoderado. Este se podrá editar y eliminar. Existe otra opción que es generar PDF en el cual al hacer clic en ello se abrirá un reporte de alumnos registrados y este se podrá guardar e imprimir.	
Observaciones: Ninguna.	

Tabla 17: *Historia de Usuario N.º 14*

Historia de Usuario	
Número: 14	Usuario: Administrador
Nombre Historia REGISTRAR DOCENTES	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 3
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: en el registro de docentes se mostrará un botón de nuevo docente en el cual al hacer clic se podrá agregar un docente con las opciones de llenado de datos como es nombres, apellidos, sexo, email, descripción, teléfono y la opción de editar y eliminar.	
Observaciones: Ninguna.	

Tabla 18: *Historia de Usuario N.º 15*

Historia de Usuario	
Número: 15	Usuario: Administrador
Nombre Historia REGISTRAR CURSOS	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 3
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: en el registro de cursos se podrá agregar un nuevo curso con su nombre, cantidad de alumnos al curso, descripción, automáticamente su docente y las opciones de editar y eliminar.	
Observaciones: Ninguna.	

Tabla 19: *Historia de Usuario N.º 16*

Historia de Usuario	
Número: 16	Usuario: Administrador
Nombre Historia GESTIONAR HORARIOS	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 3
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: al ingresar a horarios este se podrá llenar un nuevo horario con su día, hora de inicio, hora fin y la opción de editar y eliminar.	
Observaciones: Ninguna.	

Tabla 20: *Historia de Usuario N.º 17*

Historia de Usuario	
Número: 17	Usuario: Administrador
Nombre Historia GESTIONAR AÑO ACADÉMICO	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 3
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: al ingresar un año académico tendrá la opción de poner el año en números, fecha de inicio y fecha fin, con otra opción de editar y eliminar.	
Observaciones: Ninguna.	

Tabla 21: *Historia de Usuario N.º 18*

Historia de Usuario	
Número: 18	Usuario: Administrador
Nombre Historia GESTIONAR GRADOS	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 3
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: al ingresar a grados se podrá registrar los diferentes grados de primaria(6) y secundaria(5) con su nombre y también se podrá editar y eliminar.	
Observaciones: Ninguna.	

Tabla 22: *Historia de Usuario N.º 19*

Historia de Usuario	
Número: 19	Usuario: Administrador
Nombre Historia GESTIONAR NIVELES	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 4
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: al ingresar a niveles se podrá registrar dos niveles primario y secundario con la opción de editar y eliminar.	
Observaciones: Ninguna.	

Tabla 23: *Historia de Usuario N.º 20*

Historia de Usuario	
Número: 20	Usuario: Administrador
Nombre Historia GESTIONAR SECCIÓN	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 4
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: al ingresar a secciones y al botón nueva sección, aparecerá un formulario para llenar el nombre de la sección, y automáticamente tomará el grado y el nivel que se agregó anteriormente, también estará la opción de editar y eliminar .	
Observaciones: Ninguna.	

Tabla 24: *Historia de Usuario N.º 21*

Historia de Usuario	
Número: 21	Usuario: Administrador
Nombre Historia REPORTE POR DÍA	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 4
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: reporte por día de listado de alumnos matriculados.	
Observaciones: Ninguna.	

Tabla 25: *Historia de Usuario N.º 22*

Historia de Usuario	
Número: 22	Usuario: Administrador
Nombre Historia REPORTE POR FECHA	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 4
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: reporte por fecha (día, mes, año) de alumnos matriculados.	
Observaciones: Ninguna.	

Tabla 26: *Historia de Usuario N.º 23*

Historia de Usuario	
Número: 23	Usuario: Administrador
Nombre Historia REPORTE POR GRADO, SECCIÓN Y NIVEL	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 4
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: listado de reporte por grado, sección y nivel de los alumnos matriculados en PDF.	
Observaciones: Ninguna.	

Tabla 27: *Historia de Usuario N.º 24*

Historia de Usuario	
Número: 24	Usuario: Administrador
Nombre Historia REPORTE POR SECCIÓN	
Prioridad en negocio: ALTA (ALTA/MEDIA/BAJA)	Riesgo del Desarrollo: ALTA (ALTA/MEDIA/BAJA)
Puntos estimados:	Iteración asignada: 4
Programadores responsables: Alvarez Sarmiento, Erwin Miler. Aliaga Cabrera, Israel David	
Descripción: reporte por sección de cada grado y nivel de los alumnos que se hayan matriculado.	
Observaciones: Ninguna.	

✓ **Plan de entregas**

Tabla 28: Plan de entregas

NÚMERO	HISTORIA DE USUARIO	ITERACIONES			
		1	2	3	4
1	INICIO	X			
2	ACERCA DE	X			
3	CONTACTO	X			
4	INICIAR SESIÓN	X			
5	SOBRE NOSOTROS	X			
6	CURSOS	X			
7	IMÁGENES	X			
8	LOGIN		X		
9	PANEL PRINCIPAL DE LA INSTITUCIÓN		X		
10	REGISTRAR MATRÍCULA		X		
11	PERFIL DE USUARIO		X		
12	REGISTRAR APODERADO		X		
13	REGISTRA ALUMNOS			X	
14	REGISTRAR DOCENTES			X	
15	REGISTRAR CURSOS			X	
16	REGISTRAR HORARIOS			X	
17	GESTIONAR AÑO ACADÉMICO			X	
18	GESTIONAR GRADOS			X	
19	GESTIONAR NIVELES				X
20	GESTIONAR SECCIONES				X
21	REPORTE POR DÍA				X
22	REPORTE POR FECHA				X
23	REPORTE POR GRADO, SECCIÓN Y NIVEL				X
24	REPORTE POR SECCIÓN				X

✓ Plan de iteraciones

Tabla 29: Plan de iteraciones

ITERACIÓN	SEMANAS																									
	N.º	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
1	1	X																								
	2		X																							
	3			X																						
	4				X																					
	5					X																				
	6						X																			
	7							X																		
2	8							X																		
	9								X																	
	10									X																
	11										X															
3	12											X														
	13												X													
	14													X												
	15														X											
	16															X										
	17																X									
	18																	X								
4	19																		X							
	20																			X						
	21																				X					
	22																					X				
	23																						X			
	24																							X		X

La tabla 29, representa como se agrupó cada historia de usuario a cada iteración y el tiempo en semanas para cumplir cada uno de ellos.

4.3. DISEÑO Y DESARROLLO

El desarrollo de nuestro sistema académico web se ha realizado con los siguientes programas:

- ✓ Lenguaje de Programación PHP.
- ✓ Herramienta de desarrollo HTML (HyperText Markup Language).
- ✓ CSS, en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.
- ✓ Laragon (suite de desarrollo para PHP que integra ssh, nodejs, php, phpmyadmin, etc).
- ✓ LARAVEL (Framework de desarrollo).
- ✓ Composer (Gestor de Dependencias de PHP).
- ✓ PhpMyAdmin (Sistema gestor de base de datos).
- ✓ Sublime Text (editor de texto y editor de código fuente).
- ✓ MySQL Workbench (Motor de base de datos).
- ✓ Desarrollado bajo la arquitectura MVC (Modelo-Vista-Controlador).
- ✓ Servidor Web Apache.
- ✓ Sistema Operativo Windows 11.
- ✓ Adobe Photoshop.
- ✓ JavaScript y DOM, para la interacción con el usuario.
- ✓ AdminLTE es una plantilla open source para paneles de control o dashboards desarrollada con Bootstrap 3, la cual nos brinda vistas reusables y responsive de muchos componentes comunes para la parte administrativa de un proyecto.
- ✓ Blade, el motor de plantillas.
- ✓ DomPDF es una sencilla alternativa para la construcción de PDF en PHP.

Creación de código PHP en Sublime Text con el Modelo – Vista – Controlador

Figura 1: Botón de registrar matrícula.

```
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Matricula extends Model
{
    use HasFactory;

    protected $table='matricula';
    protected $primaryKey="id_matricula";

    public $timestamp=false;

    // ...

    protected $fillable = [
        'id_alumno',
        // 'id_matricula',
        // 'total_matricula',
        'tipo_comprobante',
        'serie_comprobante',
        'numero_comprobante',
        'fecha_hora_matricula'
    ];

    protected $guarded = [
    ];

    public function alumno()
    {
        return $this->belongsTo(Alumno::class,'id_alumno');
    }

    public function detallesMatricula()
    {
        return $this->hasMany(DetalleMatricula::class,'id_matricula'
        );
    }

    public function seccion()
    {
        return $this->belongsTo(Seccion::class,'id_matricula');
    }
}
```

Nota. Código PHP de registrar matrícula.

Figura 2: Botón de detalle de registrar matrícula.

```
<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class DetalleMatricula extends Model
{
    use HasFactory;

    protected $table='detalle_matricula';
    protected $primaryKey="id_dm";

    public $timestamp=false;

    protected $fillable = [
        'id_matricula',
        'id_curso',
        'id_anio_academico',
        'id_seccion',
        'id_horario',
        'costo_matricula',
        'estado'
    ];

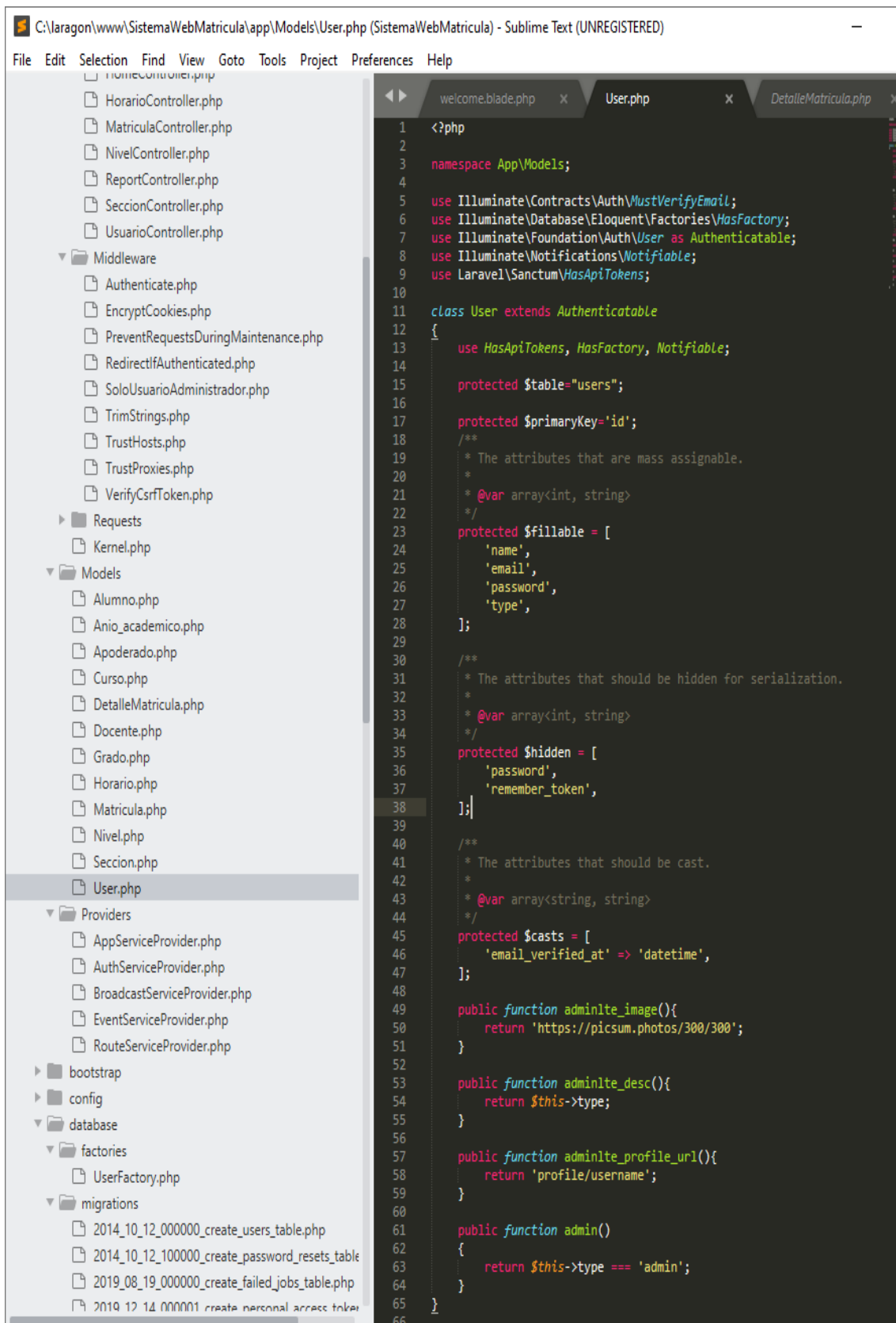
    protected $guarded = [
    ];

    public function matricula()
    {
        return $this->belongsTo(Matricula::class,'id_matricula');
    }

    public function seccion()
    {
        return $this->belongsTo(Seccion::class,'id_seccion');
    }
}
```

Nota. Código PHP de detalle de matrícula.

Figura 3: Botón de perfil de usuario

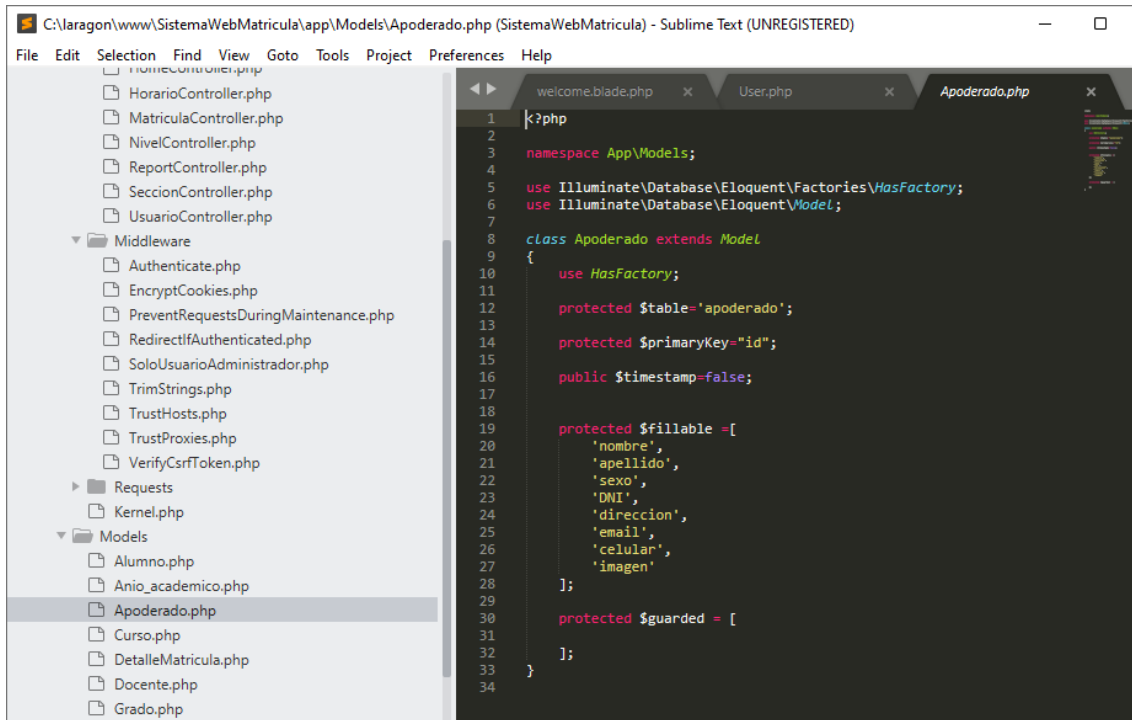


The image shows a screenshot of a code editor (Sublime Text) displaying the PHP code for a User model. The editor window title is "C:\laragon\www\SistemaWebMatricula\app\Models\User.php (SistemaWebMatricula) - Sublime Text (UNREGISTERED)". The left sidebar shows a file explorer with a tree view of the project structure, including folders like Middleware, Requests, Models, Providers, bootstrap, config, database, factories, and migrations. The main editor area shows the code for the User class, which extends the Authenticatable trait. The code includes namespace declarations, use statements for various Laravel traits and factories, and class-level attributes for mass assignment, hidden attributes, and casts. It also defines several public methods: adminlte_image(), adminlte_desc(), adminlte_profile_url(), and admin().

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\Factories\HasFactory;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8 use Illuminate\Notifications\Notifiable;
9 use Laravel\Sanctum\HasApiTokens;
10
11 class User extends Authenticatable
12 {
13     use HasApiTokens, HasFactory, Notifiable;
14
15     protected $table="users";
16
17     protected $primaryKey='id';
18     /**
19      * The attributes that are mass assignable.
20      *
21      * @var array<int, string>
22      */
23     protected $fillable = [
24         'name',
25         'email',
26         'password',
27         'type',
28     ];
29
30     /**
31      * The attributes that should be hidden for serialization.
32      *
33      * @var array<int, string>
34      */
35     protected $hidden = [
36         'password',
37         'remember_token',
38     ];
39
40     /**
41      * The attributes that should be cast.
42      *
43      * @var array<string, string>
44      */
45     protected $casts = [
46         'email_verified_at' => 'datetime',
47     ];
48
49     public function adminlte_image(){
50         return 'https://picsum.photos/300/300';
51     }
52
53     public function adminlte_desc(){
54         return $this->type;
55     }
56
57     public function adminlte_profile_url(){
58         return 'profile/username';
59     }
60
61     public function admin()
62     {
63         return $this->type === 'admin';
64     }
65 }
66
```

Nota. Código PHP de perfil de usuario.

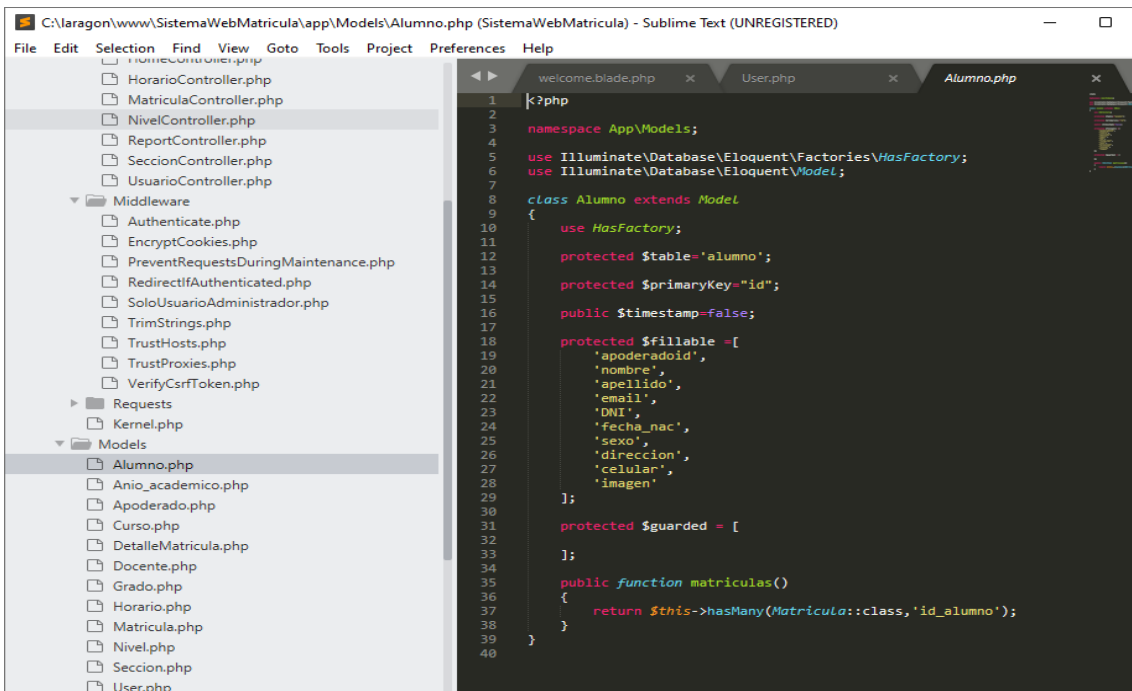
Figura 4: Botón de registrar apoderado.



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Apoderado extends Model
9 {
10     use HasFactory;
11
12     protected $table='apoderado';
13
14     protected $primaryKey="id";
15
16     public $timestamps=false;
17
18     protected $fillable =[
19         'nombre',
20         'apellido',
21         'sexo',
22         'DNI',
23         'direccion',
24         'email',
25         'celular',
26         'imagen'
27     ];
28
29     protected $guarded = [
30     ];
31
32 }
33
34
```

Nota. Código PHP de registrar apoderado.

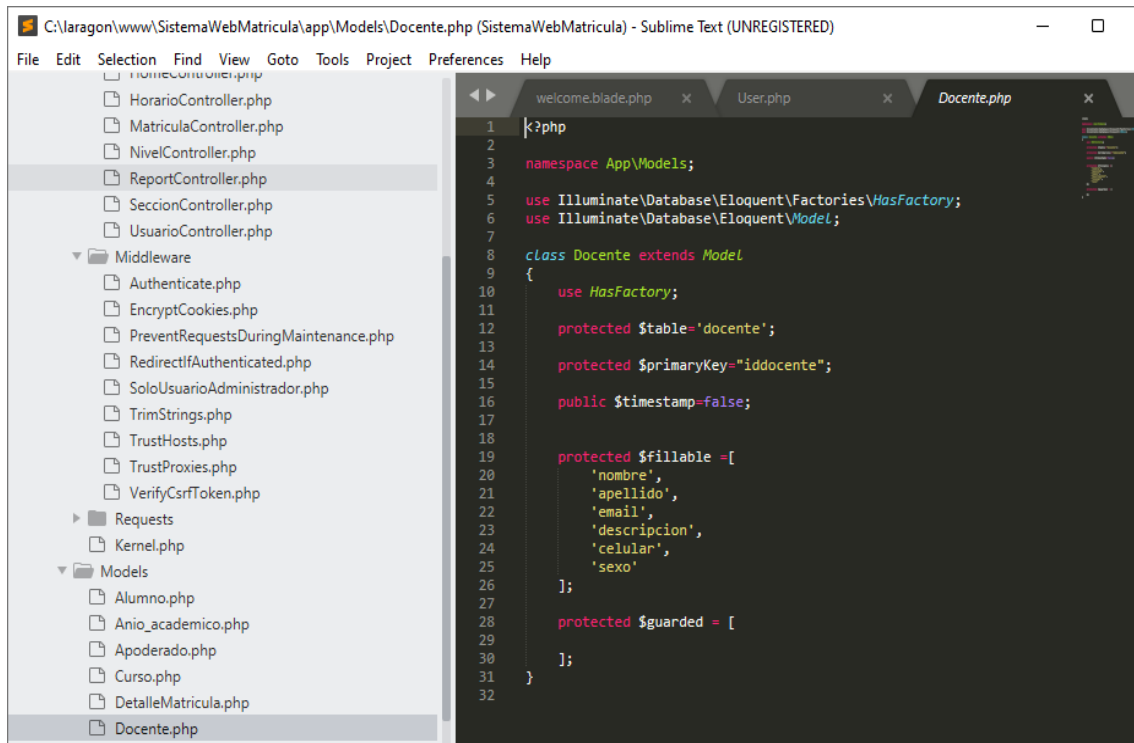
Figura 5: Botón de registrar alumno.



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Alumno extends Model
9 {
10     use HasFactory;
11
12     protected $table='alumno';
13
14     protected $primaryKey="id";
15
16     public $timestamps=false;
17
18     protected $fillable =[
19         'apoderadoid',
20         'nombre',
21         'apellido',
22         'email',
23         'DNI',
24         'fecha_nac',
25         'sexo',
26         'direccion',
27         'celular',
28         'imagen'
29     ];
30
31     protected $guarded = [
32     ];
33
34     public function matriculas()
35     {
36         return $this->hasMany(Matricula::class,'id_alumno');
37     }
38
39 }
40
```

Nota. Código PHP de registrar alumno.

Figura 6: Botón de registrar docentes.

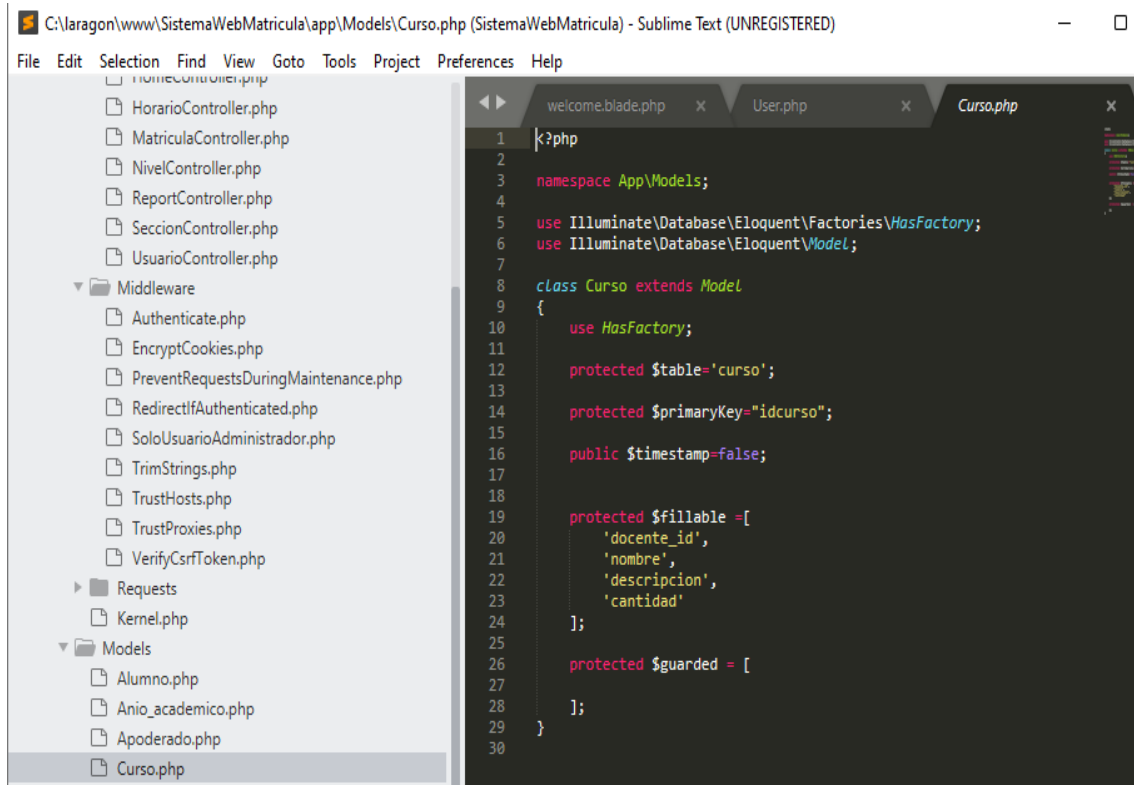


The screenshot shows a code editor window titled "C:\laragon\www\SistemaWebMatricula\app\Models\Docente.php (SistemaWebMatricula) - Sublime Text (UNREGISTERED)". The left sidebar displays a file explorer with a tree view of the project structure, including folders for Middleware and Models, and various controller and model files. The main editor area shows the code for the `Docente` model, which extends the `Model` class. The code includes namespace declarations, use statements for `HasFactory` and `Model`, and class properties for table name, primary key, timestamp, fillable attributes, and guarded attributes.

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Docente extends Model
9 {
10     use HasFactory;
11
12     protected $table='docente';
13
14     protected $primaryKey="iddocente";
15
16     public $timestamp=false;
17
18
19     protected $fillable =[
20         'nombre',
21         'apellido',
22         'email',
23         'descripcion',
24         'celular',
25         'sexo'
26     ];
27
28     protected $guarded = [
29
30     ];
31
32 }
```

Nota. Código PHP de registrar docentes.

Figura 7: Botón de registrar cursos.

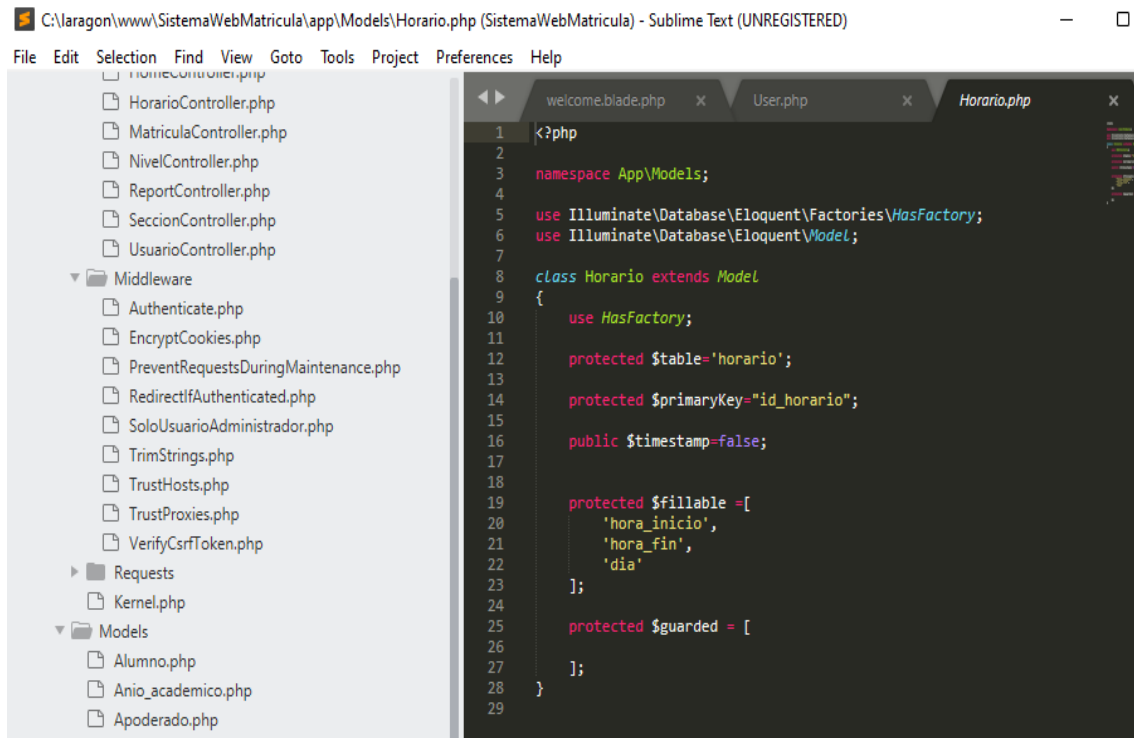


The screenshot shows a code editor window titled "C:\laragon\www\SistemaWebMatricula\app\Models\Curso.php (SistemaWebMatricula) - Sublime Text (UNREGISTERED)". The left sidebar displays a file explorer with a tree view of the project structure, including folders for Middleware and Models, and various controller and model files. The main editor area shows the code for the `Curso` model, which extends the `Model` class. The code includes namespace declarations, use statements for `HasFactory` and `Model`, and class properties for table name, primary key, timestamp, fillable attributes, and guarded attributes.

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Curso extends Model
9 {
10     use HasFactory;
11
12     protected $table='curso';
13
14     protected $primaryKey="idcurso";
15
16     public $timestamp=false;
17
18
19     protected $fillable =[
20         'docente_id',
21         'nombre',
22         'descripcion',
23         'cantidad'
24     ];
25
26     protected $guarded = [
27
28     ];
29
30 }
```

Nota. Código PHP de registrar cursos.

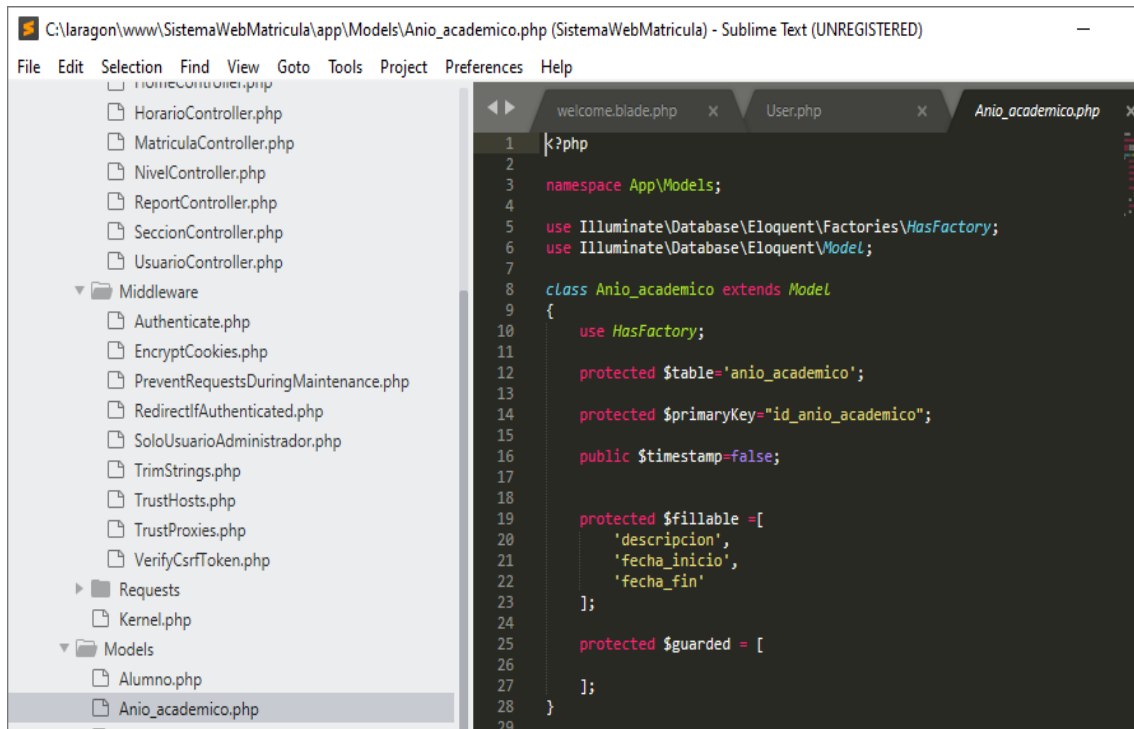
Figura 8: Botón de registrar horario.



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Horario extends Model
9 {
10     use HasFactory;
11
12     protected $table='horario';
13
14     protected $primaryKey="id_horario";
15
16     public $timestamp=false;
17
18
19     protected $fillable = [
20         'hora_inicio',
21         'hora_fin',
22         'dia'
23     ];
24
25     protected $guarded = [
26
27     ];
28 }
29
```

Nota. Código PHP de registrar horario.

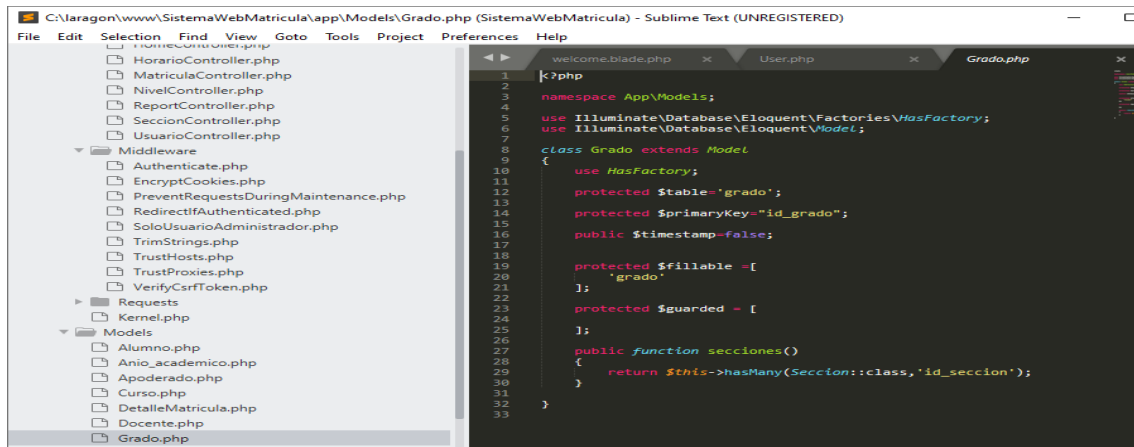
Figura 9: Botón de registrar año académico.



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Anio_academico extends Model
9 {
10     use HasFactory;
11
12     protected $table='anio_academico';
13
14     protected $primaryKey="id_anio_academico";
15
16     public $timestamp=false;
17
18
19     protected $fillable = [
20         'descripcion',
21         'fecha_inicio',
22         'fecha_fin'
23     ];
24
25     protected $guarded = [
26
27     ];
28 }
29
```

Nota. Código PHP de registrar año académico.

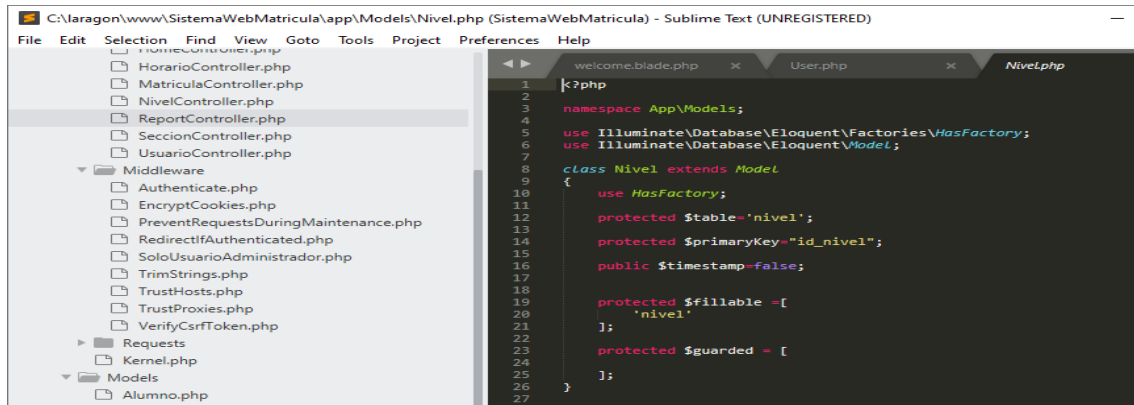
Figura 10: Botón de registrar secciones(Grado).



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Grado extends Model
9 {
10     use HasFactory;
11     protected $table='grado';
12     protected $primaryKey="id_grado";
13     public $timestamp=false;
14
15     protected $fillable =[
16         'grado'
17     ];
18     protected $guarded = [
19     ];
20
21     public function secciones()
22     {
23         return $this->hasMany(Seccion::class,'id_seccion');
24     }
25 }
26
27
28
29
30
31
32
33
```

Nota. Código PHP de registrar secciones(Grado).

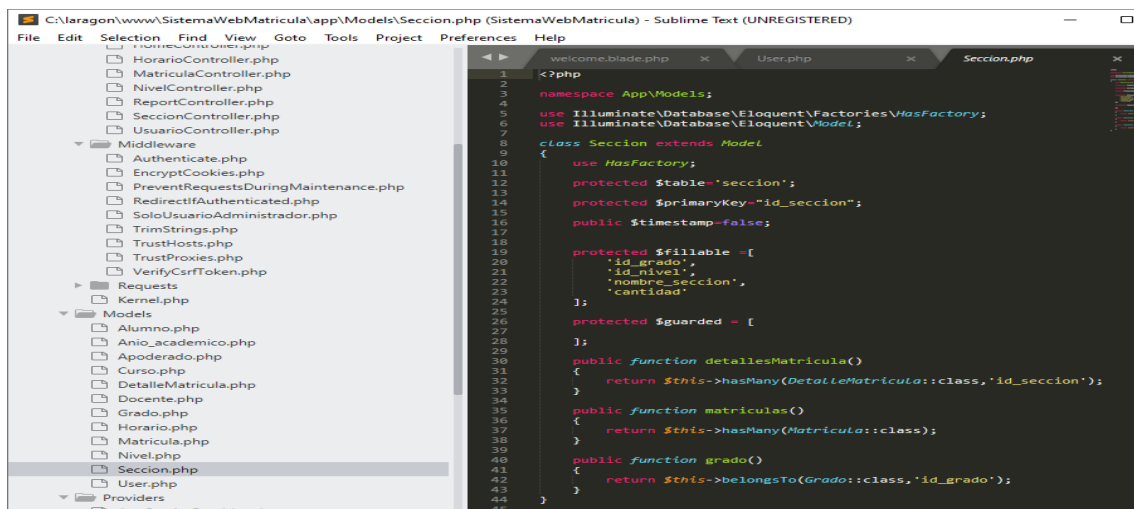
Figura 11: Botón de registrar secciones(Nivel).



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Nivel extends Model
9 {
10     use HasFactory;
11     protected $table='nivel';
12     protected $primaryKey="id_nivel";
13     public $timestamp=false;
14
15     protected $fillable =[
16         'nivel'
17     ];
18     protected $guarded = [
19     ];
20
21
22
23
24
25
26
27
28
29
```

Nota. Código PHP de registrar secciones (Nivel).

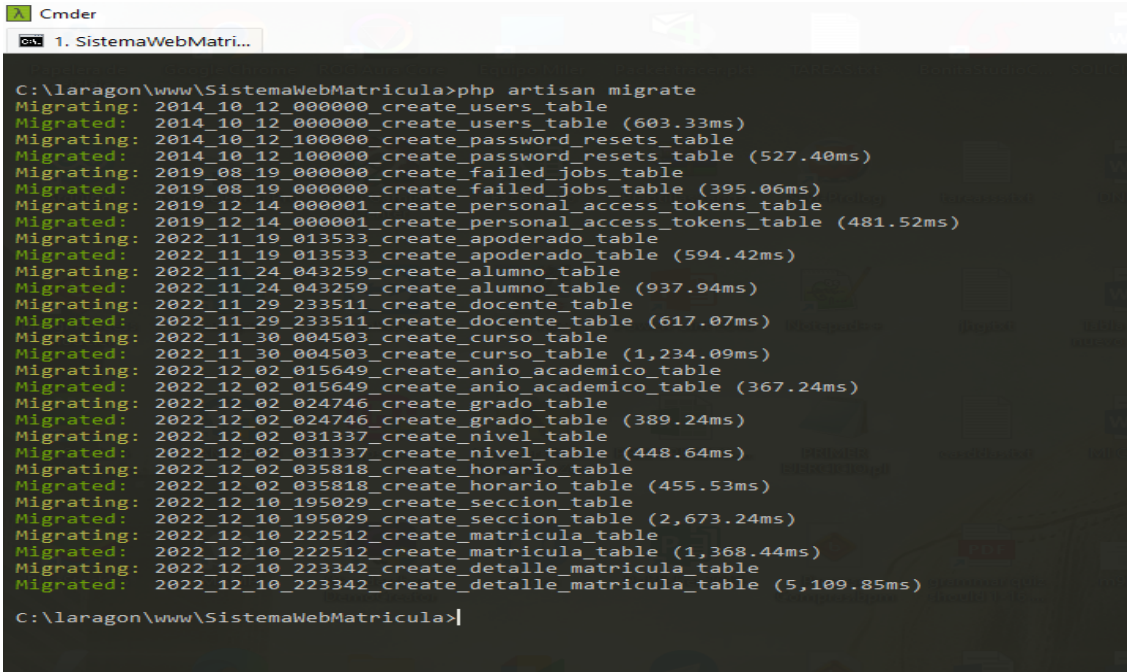
Figura 12: Botón de registrar secciones(Sección).



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Seccion extends Model
9 {
10     use HasFactory;
11     protected $table='seccion';
12     protected $primaryKey="id_seccion";
13     public $timestamp=false;
14
15     protected $fillable =[
16         'id_grado',
17         'id_nivel',
18         'nombre_seccion',
19         'cantidad'
20     ];
21     protected $guarded = [
22     ];
23
24     public function detallesMatricula()
25     {
26         return $this->hasMany(DetalleMatricula::class,'id_seccion');
27     }
28     public function matriculas()
29     {
30         return $this->hasMany(Matricula::class);
31     }
32     public function grado()
33     {
34         return $this->belongsTo(Grado::class,'id_grado');
35     }
36
37
38
39
40
41
42
43
44
45
```

Nota. Código PHP de registrar secciones (Sección).

Figura 13: Migraciones.

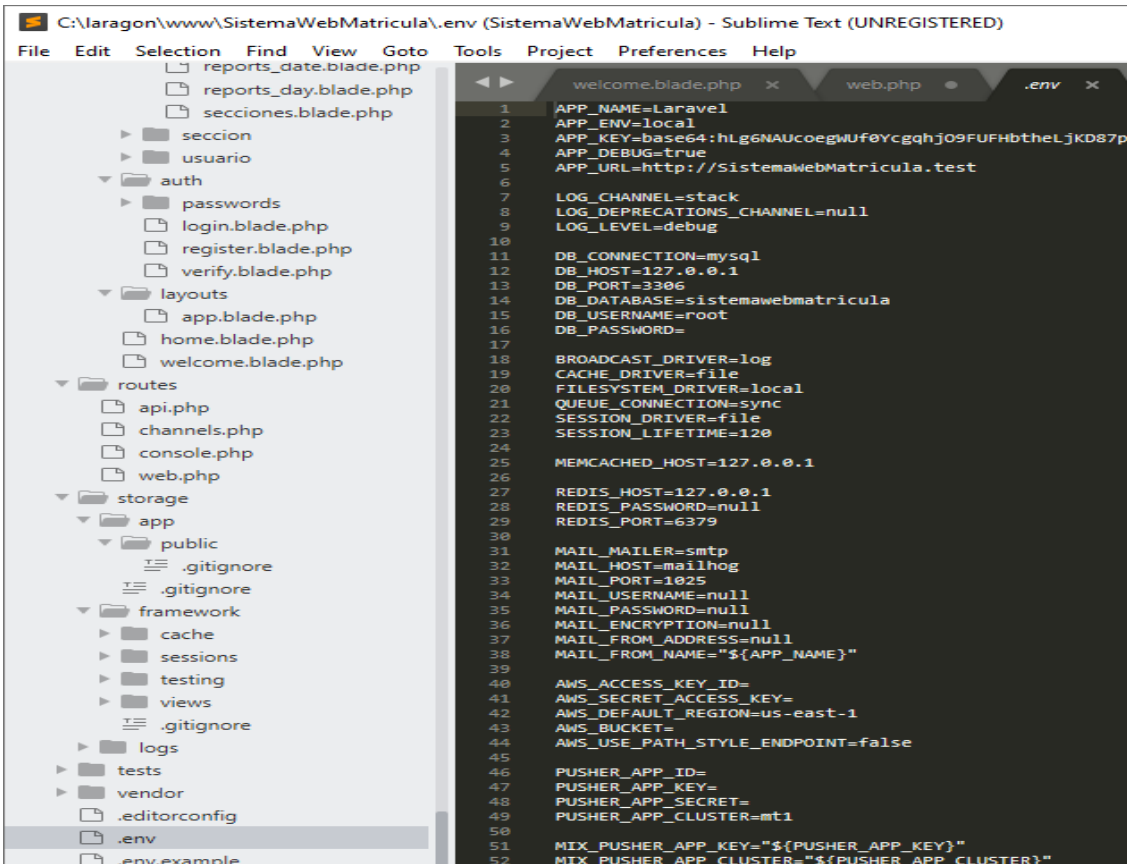


```
C:\laragon\www\SistemaWebMatricula>php artisan migrate
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (603.33ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (527.40ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (395.06ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (481.52ms)
Migrating: 2022_11_19_013533_create_apoderado_table
Migrated: 2022_11_19_013533_create_apoderado_table (594.42ms)
Migrating: 2022_11_24_043259_create_alumno_table
Migrated: 2022_11_24_043259_create_alumno_table (937.94ms)
Migrating: 2022_11_29_233511_create_docente_table
Migrated: 2022_11_29_233511_create_docente_table (617.07ms)
Migrating: 2022_11_30_004503_create_curso_table
Migrated: 2022_11_30_004503_create_curso_table (1,234.09ms)
Migrating: 2022_12_02_015649_create_anio_academico_table
Migrated: 2022_12_02_015649_create_anio_academico_table (367.24ms)
Migrating: 2022_12_02_024746_create_grado_table
Migrated: 2022_12_02_024746_create_grado_table (389.24ms)
Migrating: 2022_12_02_031337_create_nivel_table
Migrated: 2022_12_02_031337_create_nivel_table (448.64ms)
Migrating: 2022_12_02_035818_create_horario_table
Migrated: 2022_12_02_035818_create_horario_table (455.53ms)
Migrating: 2022_12_10_195029_create_seccion_table
Migrated: 2022_12_10_195029_create_seccion_table (2,673.24ms)
Migrating: 2022_12_10_222512_create_matricula_table
Migrated: 2022_12_10_222512_create_matricula_table (1,368.44ms)
Migrating: 2022_12_10_223342_create_detalle_matricula_table
Migrated: 2022_12_10_223342_create_detalle_matricula_table (5,109.85ms)

C:\laragon\www\SistemaWebMatricula>
```

Nota. Código en Sublim Text de migraciones (php artisan migrate).

Figura 14: Comunicación de la base de datos de Lavarel con phpmysqladmin.

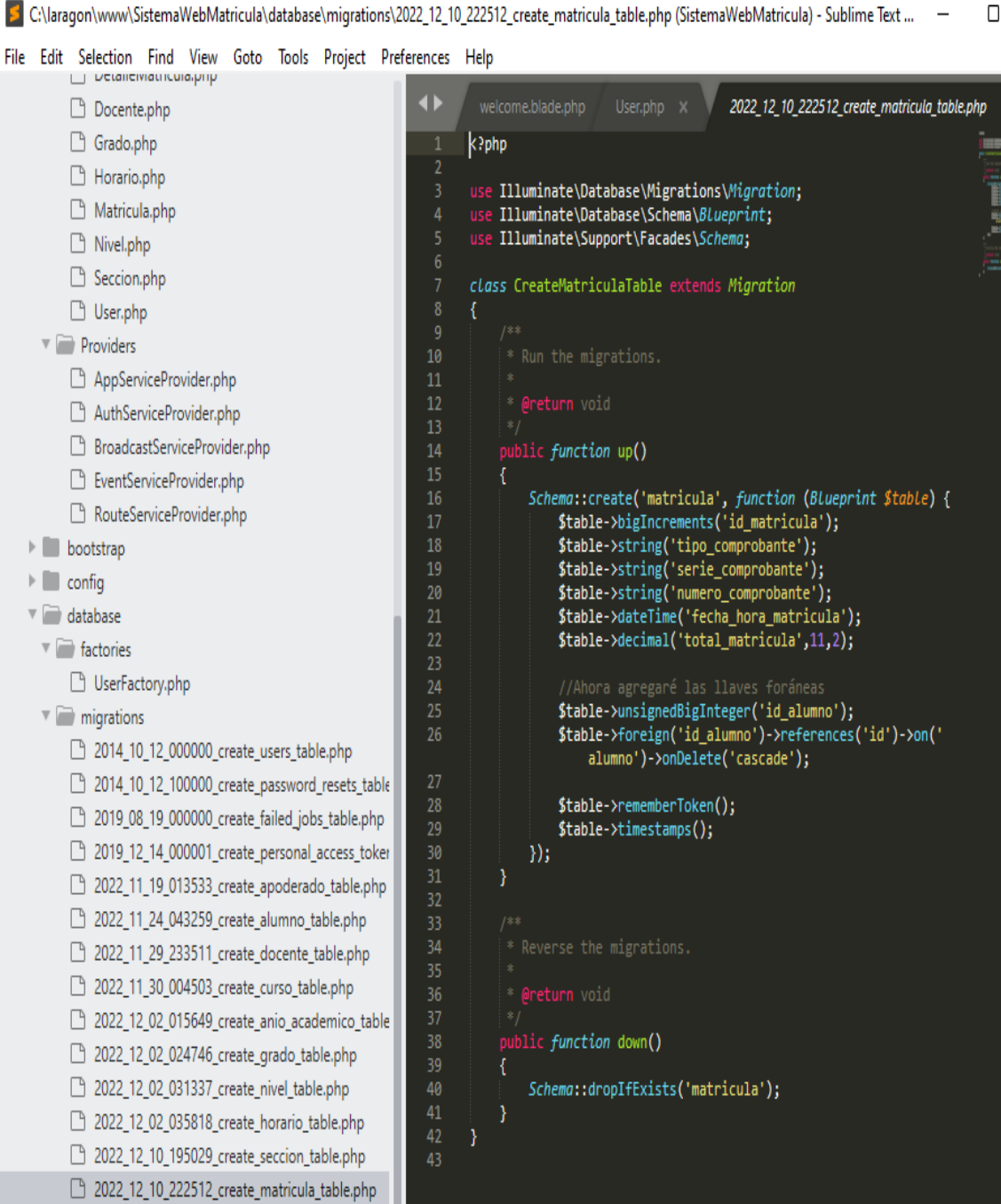


```
C:\laragon\www\SistemaWebMatricula\.env (SistemaWebMatricula) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
reports_date.blade.php
reports_day.blade.php
secciones.blade.php
seccion
usuario
auth
passwords
login.blade.php
register.blade.php
verify.blade.php
layouts
app.blade.php
home.blade.php
welcome.blade.php
routes
api.php
channels.php
console.php
web.php
storage
app
public
.gitignore
framework
cache
sessions
testing
views
.gitignore
logs
tests
vendor
.editorconfig
.env
.env.example

welcome.blade.php x web.php .env x
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:hLg6NAUcoegWUf0YcgqhJ09FUHbtheLjKD87pp
4 APP_DEBUG=true
5 APP_URL=http://SistemaWebMatricula.test
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=sistemawebmatricula
15 DB_USERNAME=root
16 DB_PASSWORD=
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DRIVER=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=null
29 REDIS_PORT=6379
30
31 MAIL_MAILER=smtp
32 MAIL_HOST=mailhog
33 MAIL_PORT=1025
34 MAIL_USERNAME=null
35 MAIL_PASSWORD=null
36 MAIL_ENCRYPTION=null
37 MAIL_FROM_ADDRESS=null
38 MAIL_FROM_NAME="${APP_NAME}"
39
40 AWS_ACCESS_KEY_ID=
41 AWS_SECRET_ACCESS_KEY=
42 AWS_DEFAULT_REGION=us-east-1
43 AWS_BUCKET=
44 AWS_USE_PATH_STYLE_ENDPOINT=false
45
46 PUSHER_APP_ID=
47 PUSHER_APP_KEY=
48 PUSHER_APP_SECRET=
49 PUSHER_APP_CLUSTER=mt1
50
51 MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
52 MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
```

Nota. Código en Sublim Text de Base de datos con Lavarel con phpmysqladmin.

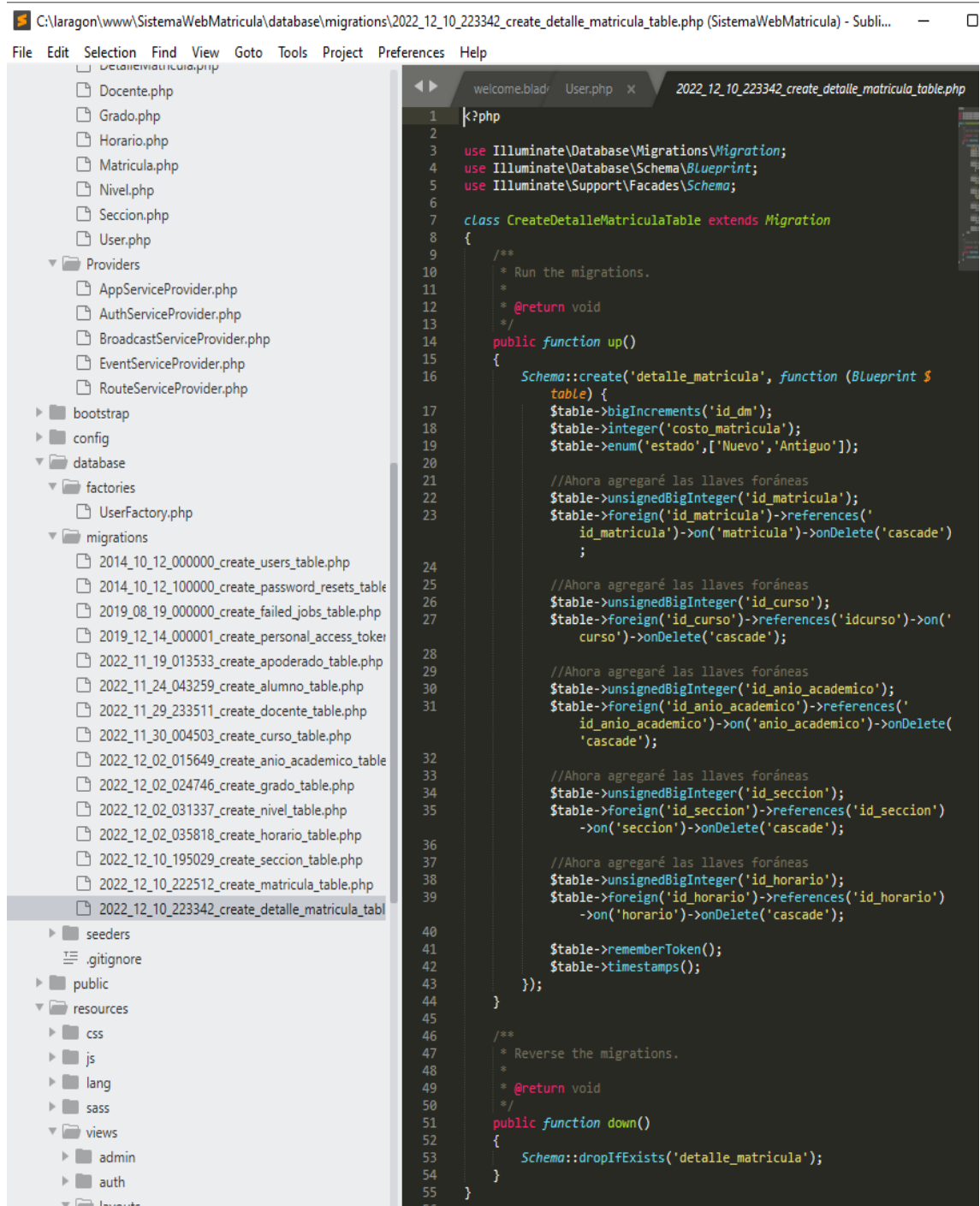
Figura 15: Migración de crear Matrícula (Registrar Matrícula).



```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateMatriculaTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('matricula', function (Blueprint $table) {
17             $table->bigIncrements('id_matricula');
18             $table->string('tipo_comprobante');
19             $table->string('serie_comprobante');
20             $table->string('numero_comprobante');
21             $table->dateTime('fecha_hora_matricula');
22             $table->decimal('total_matricula',11,2);
23
24             //Ahora agregaré las llaves foráneas
25             $table->unsignedBigInteger('id_alumno');
26             $table->foreign('id_alumno')->references('id')->on('
                alumno')->onDelete('cascade');
27
28             $table->rememberToken();
29             $table->timestamps();
30         });
31     }
32
33     /**
34      * Reverse the migrations.
35      *
36      * @return void
37      */
38     public function down()
39     {
40         Schema::dropIfExists('matricula');
41     }
42 }
43
```

Nota. Código PHP de la migración crear Matrícula (Registrar Matrícula).

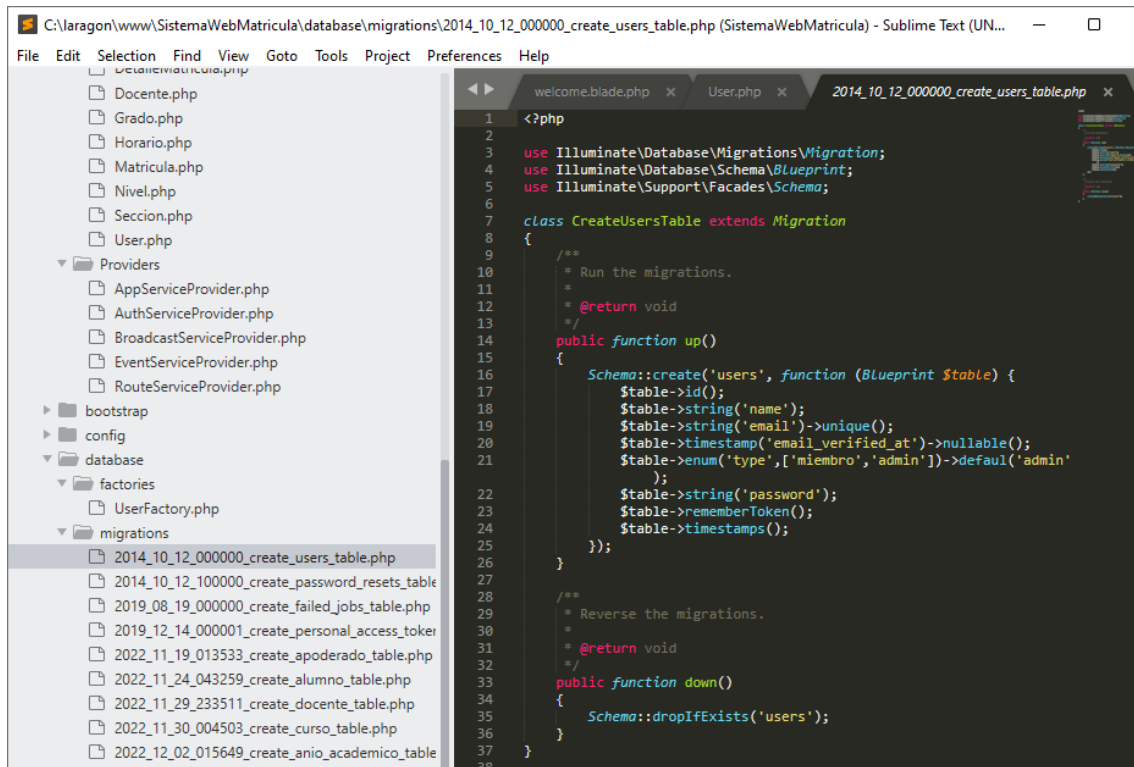
Figura 16: Migración de crear Detalle de Matrícula (Detalle Matrícula).



```
1 |<?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateDetalleMatriculaTable extends Migration
8 {
9
10     /**
11      * Run the migrations.
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('detalle_matricula', function (Blueprint $
17             table) {
18             $table->bigIncrements('id_dm');
19             $table->integer('costo_matricula');
20             $table->enum('estado', ['Nuevo', 'Antiguo']);
21
22             //Ahora agregaré las llaves foráneas
23             $table->unsignedBigInteger('id_matricula');
24             $table->foreign('id_matricula')->references('
25                 id_matricula')->on('matricula')->onDelete('cascade')
26             ;
27
28             //Ahora agregaré las llaves foráneas
29             $table->unsignedBigInteger('id_curso');
30             $table->foreign('id_curso')->references('idcurso')->on('
31                 curso')->onDelete('cascade');
32
33             //Ahora agregaré las llaves foráneas
34             $table->unsignedBigInteger('id_anio_academico');
35             $table->foreign('id_anio_academico')->references('
36                 id_anio_academico')->on('anio_academico')->onDelete(
37                 'cascade');
38
39             //Ahora agregaré las llaves foráneas
40             $table->unsignedBigInteger('id_seccion');
41             $table->foreign('id_seccion')->references('id_seccion')
42                 ->on('seccion')->onDelete('cascade');
43
44             //Ahora agregaré las llaves foráneas
45             $table->unsignedBigInteger('id_horario');
46             $table->foreign('id_horario')->references('id_horario')
47                 ->on('horario')->onDelete('cascade');
48
49             $table->rememberToken();
50             $table->timestamps();
51         });
52     }
53
54     /**
55      * Reverse the migrations.
56      * @return void
57      */
58     public function down()
59     {
60         Schema::dropIfExists('detalle_matricula');
61     }
62 }
```

Nota. Código PHP de la migración crear Matrícula (Detalle Matrícula).

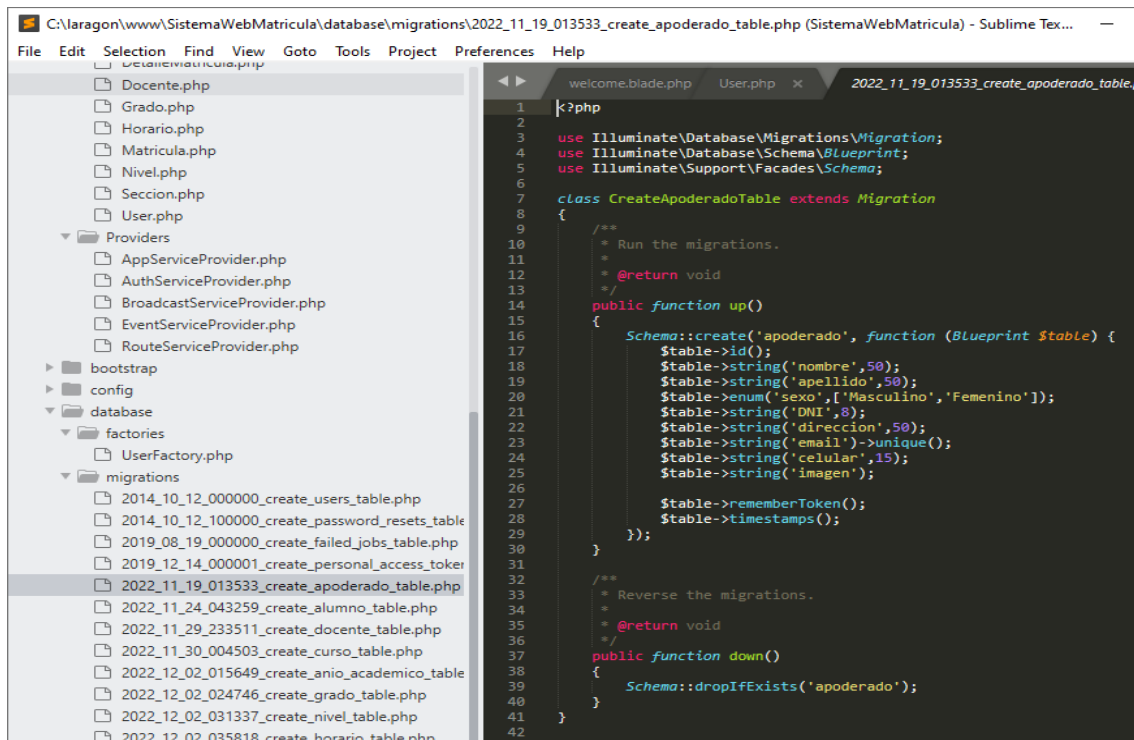
Figura 17: Migración de crear usuario.



```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateUsersTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->id();
18             $table->string('name');
19             $table->string('email')->unique();
20             $table->timestamp('email_verified_at')->nullable();
21             $table->enum('type', ['miembro', 'admin'])->default('admin');
22             $table->string('password');
23             $table->rememberToken();
24             $table->timestamps();
25         });
26     }
27
28     /**
29      * Reverse the migrations.
30      *
31      * @return void
32      */
33     public function down()
34     {
35         Schema::dropIfExists('users');
36     }
37 }
38
```

Nota. Código PHP de la migración crear Usuario (Perfil de usuario).

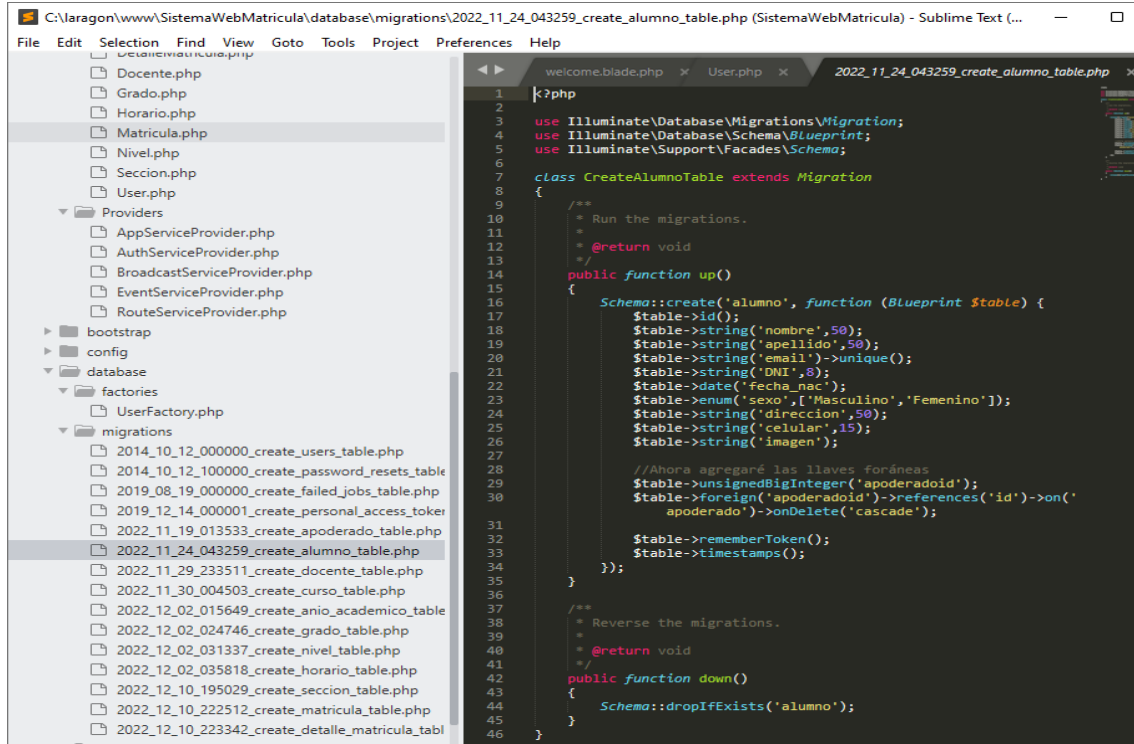
Figura 18: Migración de crear apoderado.



```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateAporeradoTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('apoderado', function (Blueprint $table) {
17             $table->id();
18             $table->string('nombre',50);
19             $table->string('apellido',50);
20             $table->enum('sexo', ['Masculino', 'Femenino']);
21             $table->string('DNI',8);
22             $table->string('direccion',50);
23             $table->string('email')->unique();
24             $table->string('celular',15);
25             $table->string('imagen');
26
27             $table->rememberToken();
28             $table->timestamps();
29         });
30     }
31
32     /**
33      * Reverse the migrations.
34      *
35      * @return void
36      */
37     public function down()
38     {
39         Schema::dropIfExists('apoderado');
40     }
41 }
42
```

Nota. Código PHP de la migración crear Apoderado (Registrar Apoderado).

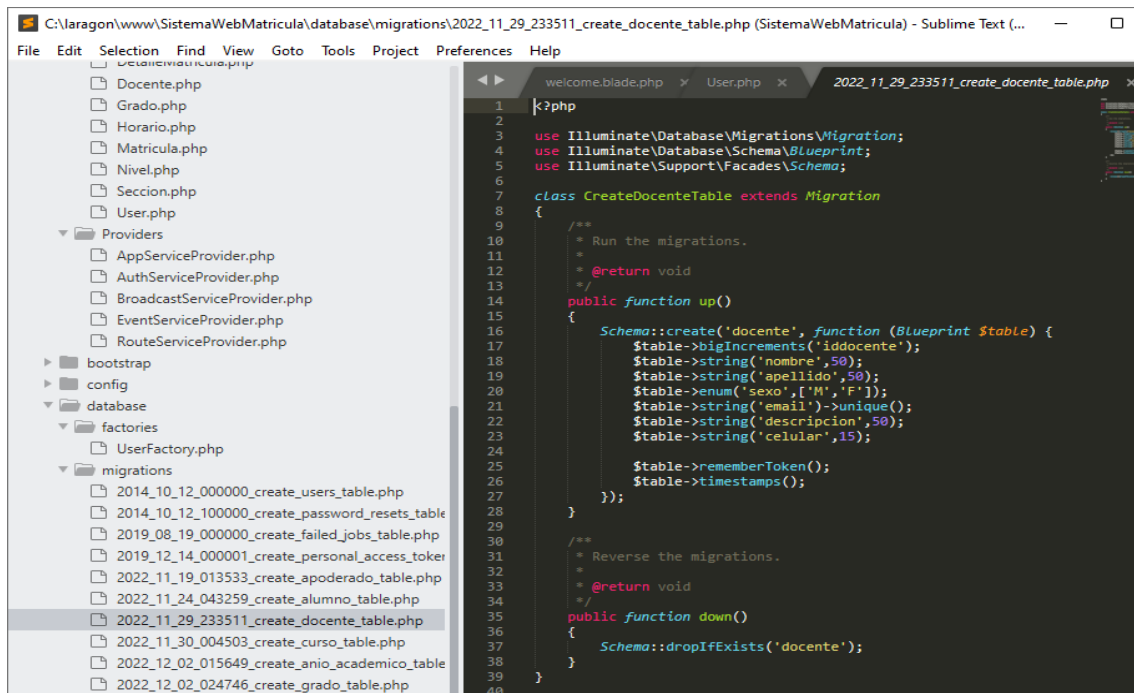
Figura 19: Migración de crear alumno.



```
1 k?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateAlumnoTable extends Migration
8 {
9
10     /**
11      * Run the migrations.
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('alumno', function (Blueprint $table) {
17             $table->id();
18             $table->string('nombre',50);
19             $table->string('apellido',50);
20             $table->string('email')->unique();
21             $table->string('DNI',8);
22             $table->date('fecha_nac');
23             $table->enum('sexo', ['Masculino', 'Femenino']);
24             $table->string('direccion',50);
25             $table->string('celular',15);
26             $table->string('imagen');
27
28             //Ahora agregaré las llaves foráneas
29             $table->unsignedBigInteger('apoderadoid');
30             $table->foreign('apoderadoid')->references('id')->on('apoderado')->onDelete('cascade');
31
32             $table->rememberToken();
33             $table->timestamps();
34         });
35     }
36
37     /**
38      * Reverse the migrations.
39      * @return void
40      */
41     public function down()
42     {
43         Schema::dropIfExists('alumno');
44     }
45 }
46
```

Nota. Código PHP de la migración crear Alumno (Registrar Alumno).

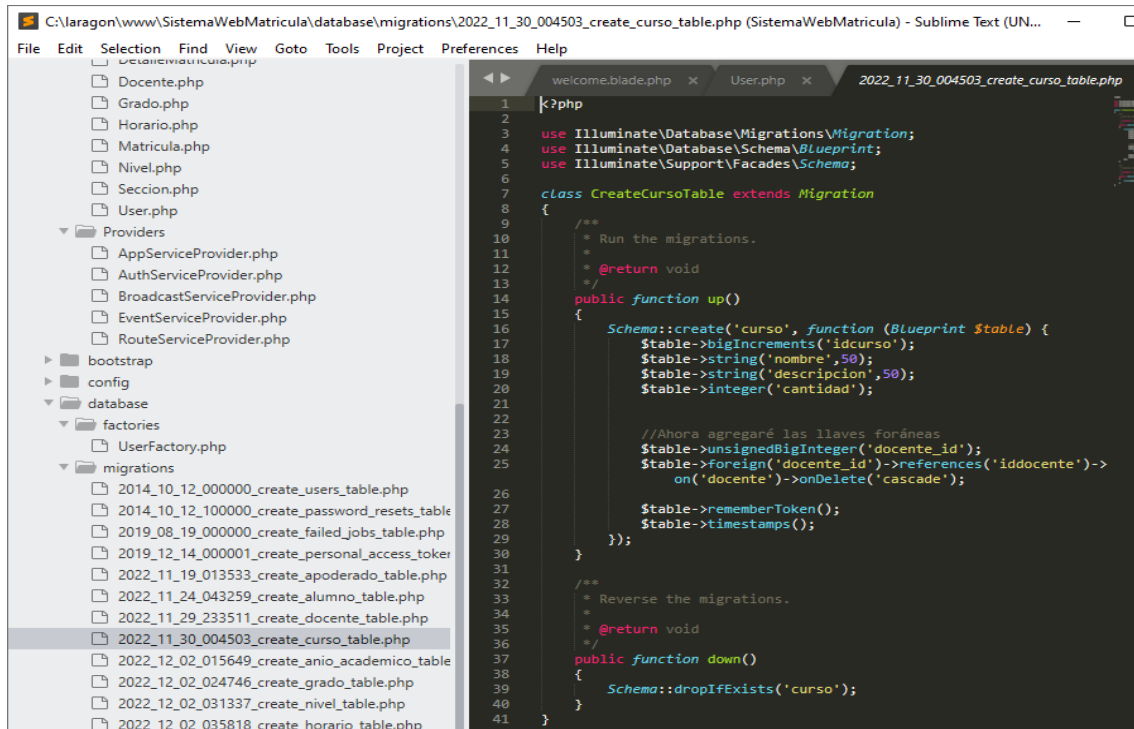
Figura 20: Migración de crear Docente.



```
1 k?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateDocenteTable extends Migration
8 {
9
10     /**
11      * Run the migrations.
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('docente', function (Blueprint $table) {
17             $table->bigIncrements('iddocente');
18             $table->string('nombre',50);
19             $table->string('apellido',50);
20             $table->enum('sexo', ['M', 'F']);
21             $table->string('email')->unique();
22             $table->string('descripcion',50);
23             $table->string('celular',15);
24
25             $table->rememberToken();
26             $table->timestamps();
27         });
28     }
29
30     /**
31      * Reverse the migrations.
32      * @return void
33      */
34     public function down()
35     {
36         Schema::dropIfExists('docente');
37     }
38 }
39
40
```

Nota. Código PHP de la migración crear Docente (Registrar Docente).

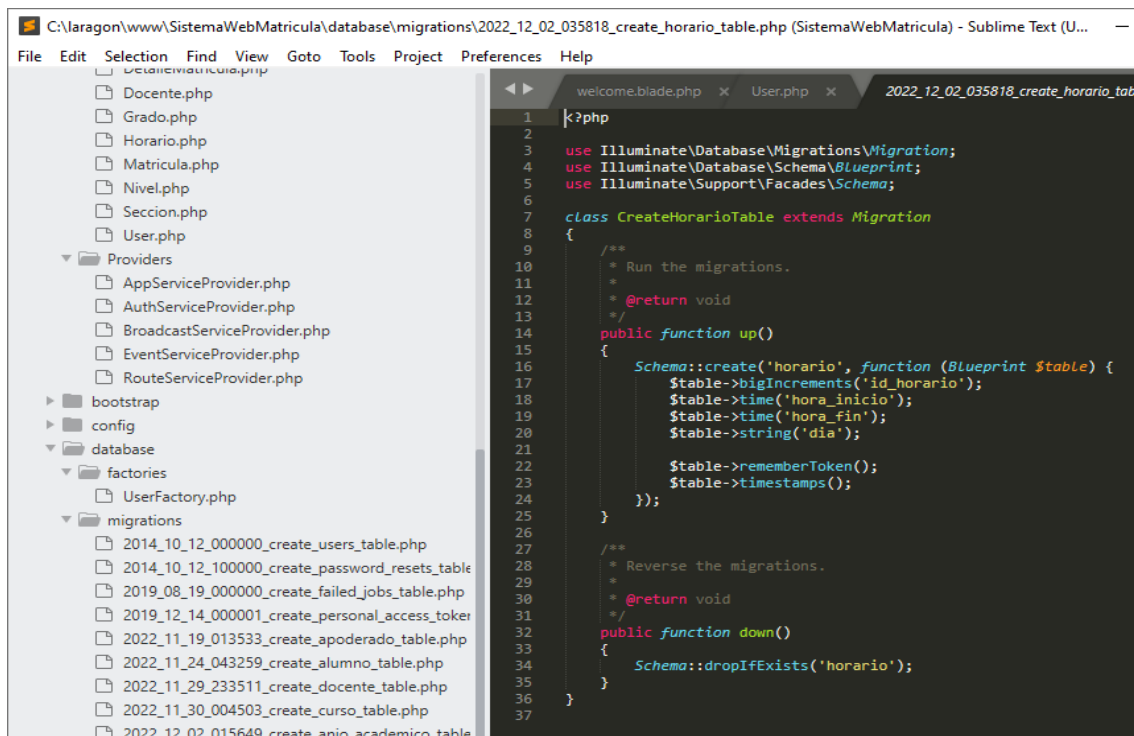
Figura 21: Migración de crear Cursos.



```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateCursoTable extends Migration
8 {
9
10     /**
11      * Run the migrations.
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('curso', function (Blueprint $table) {
17             $table->bigIncrements('idcurso');
18             $table->string('nombre',50);
19             $table->string('descripcion',50);
20             $table->integer('cantidad');
21
22
23             //Ahora agregaré las llaves foráneas
24             $table->unsignedBigInteger('docente_id');
25             $table->foreign('docente_id')->references('iddocente')->
26                 on('docente')->onDelete('cascade');
27
28             $table->rememberToken();
29             $table->timestamps();
30         });
31
32     /**
33      * Reverse the migrations.
34      * @return void
35      */
36     public function down()
37     {
38         Schema::dropIfExists('curso');
39     }
40 }
41
```

Nota. Código PHP de la migración crear Cursos (Registrar Cursos).

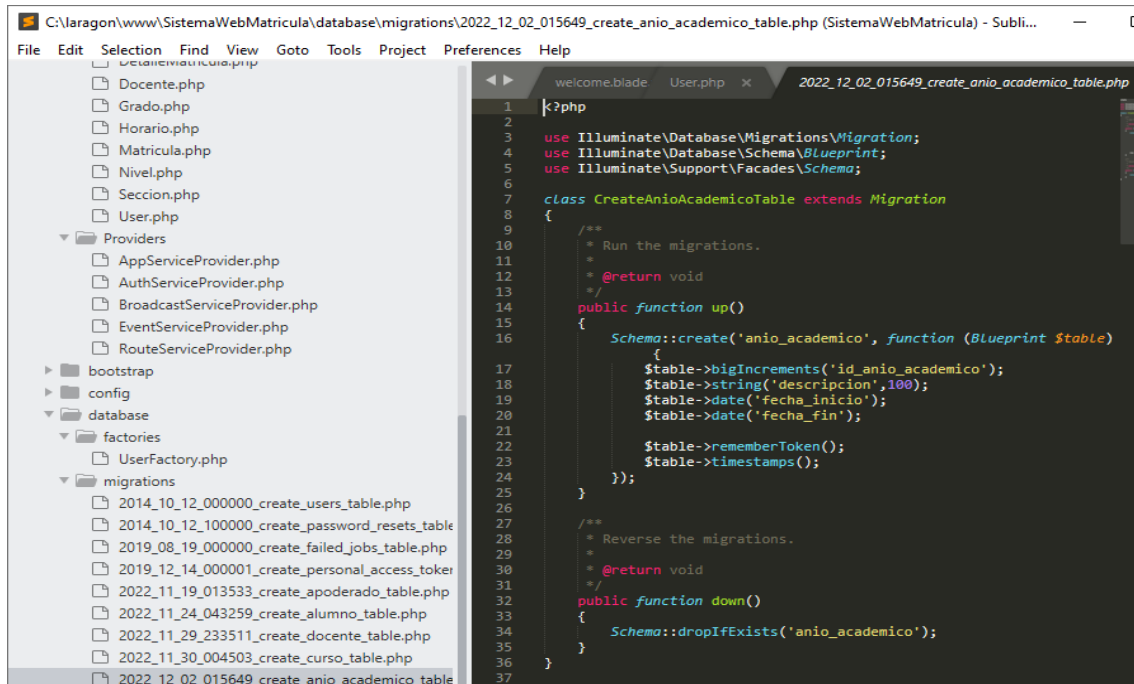
Figura 22: Migración de crear Horario.



```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateHorarioTable extends Migration
8 {
9
10     /**
11      * Run the migrations.
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('horario', function (Blueprint $table) {
17             $table->bigIncrements('id_horario');
18             $table->time('hora_inicio');
19             $table->time('hora_fin');
20             $table->string('dia');
21
22             $table->rememberToken();
23             $table->timestamps();
24         });
25
26     /**
27      * Reverse the migrations.
28      * @return void
29      */
30     public function down()
31     {
32         Schema::dropIfExists('horario');
33     }
34 }
35
36
37
```

Nota. Código PHP de la migración crear Horario (Registrar Horario).

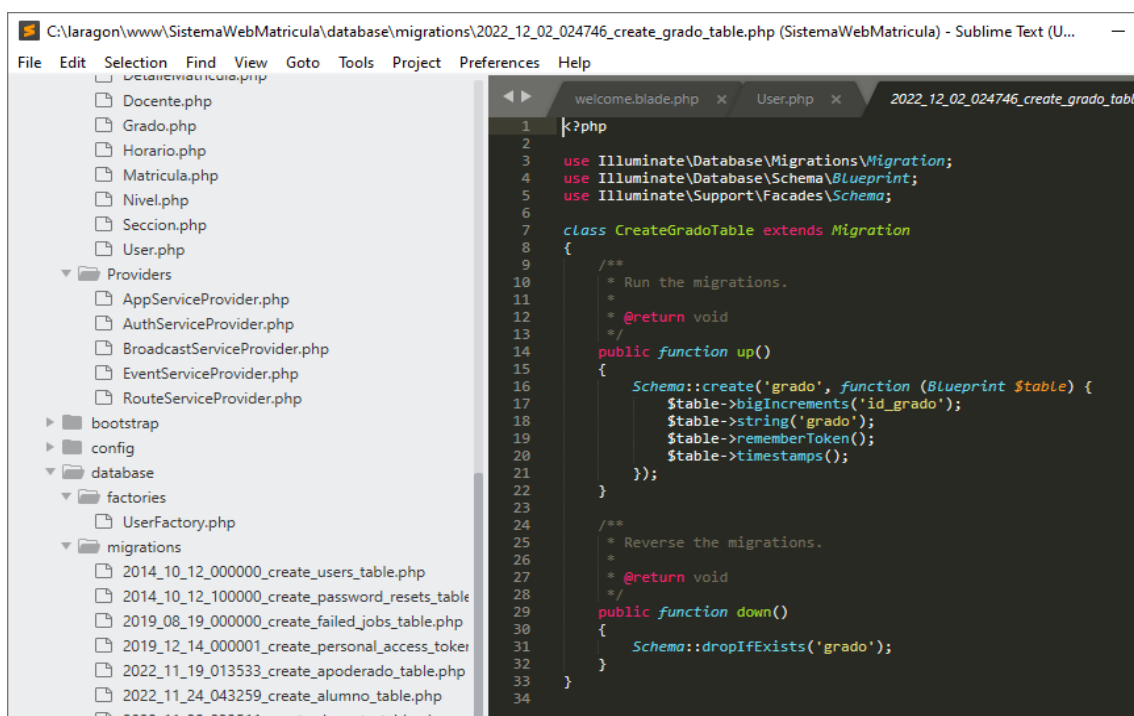
Figura 23: Migración de crear Año Académico.



```
1 |<?php
2 |
3 | use Illuminate\Database\Migrations\Migration;
4 | use Illuminate\Database\Schema\Blueprint;
5 | use Illuminate\Support\Facades\Schema;
6 |
7 | class CreateAnioAcademicoTable extends Migration
8 | {
9 |     /**
10 |      * Run the migrations.
11 |      *
12 |      * @return void
13 |      */
14 |     public function up()
15 |     {
16 |         Schema::create('anio_academico', function (Blueprint $table)
17 |         {
18 |             $table->bigIncrements('id_anio_academico');
19 |             $table->string('descripcion',100);
20 |             $table->date('fecha_inicio');
21 |             $table->date('fecha_fin');
22 |
23 |             $table->rememberToken();
24 |             $table->timestamps();
25 |         });
26 |     }
27 |
28 |     /**
29 |      * Reverse the migrations.
30 |      *
31 |      * @return void
32 |      */
33 |     public function down()
34 |     {
35 |         Schema::dropIfExists('anio_academico');
36 |     }
37 | }
```

Nota. Código PHP migración crear Año Académico (Gestionar Año Académico).

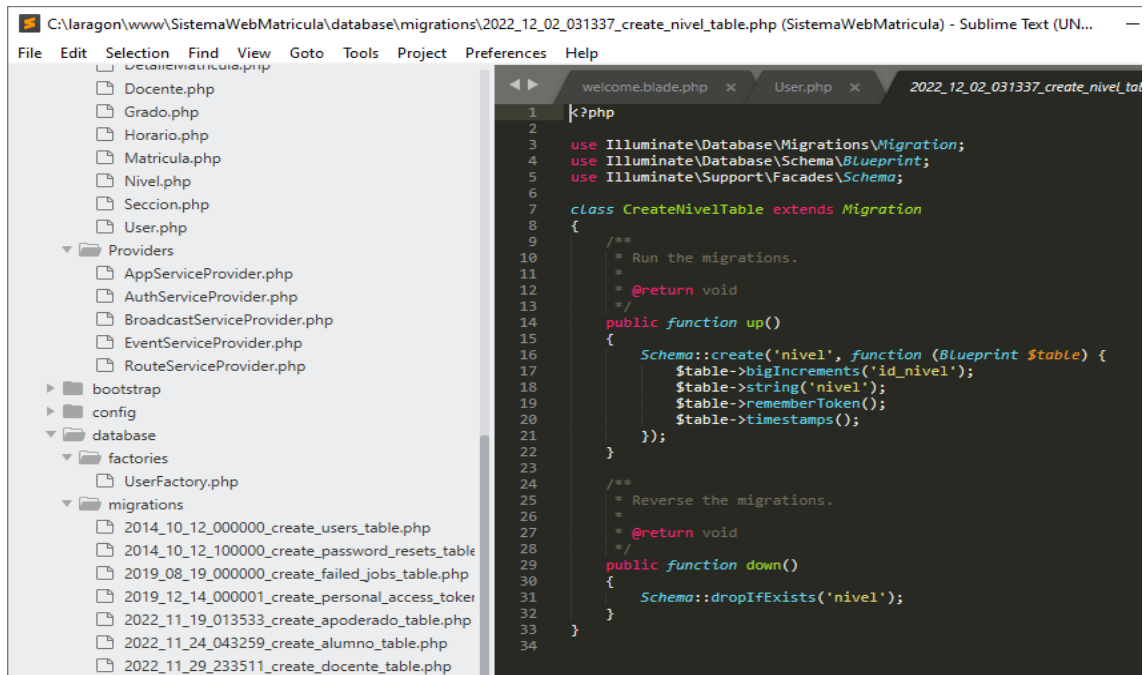
Figura 24: Migración de crear Secciones (Registrar Grados).



```
1 |<?php
2 |
3 | use Illuminate\Database\Migrations\Migration;
4 | use Illuminate\Database\Schema\Blueprint;
5 | use Illuminate\Support\Facades\Schema;
6 |
7 | class CreateGradoTable extends Migration
8 | {
9 |     /**
10 |      * Run the migrations.
11 |      *
12 |      * @return void
13 |      */
14 |     public function up()
15 |     {
16 |         Schema::create('grado', function (Blueprint $table) {
17 |             $table->bigIncrements('id_grado');
18 |             $table->string('grado');
19 |             $table->rememberToken();
20 |             $table->timestamps();
21 |         });
22 |     }
23 |
24 |     /**
25 |      * Reverse the migrations.
26 |      *
27 |      * @return void
28 |      */
29 |     public function down()
30 |     {
31 |         Schema::dropIfExists('grado');
32 |     }
33 | }
```

Nota. Código PHP de la migración crear Secciones (Registrar Grados).

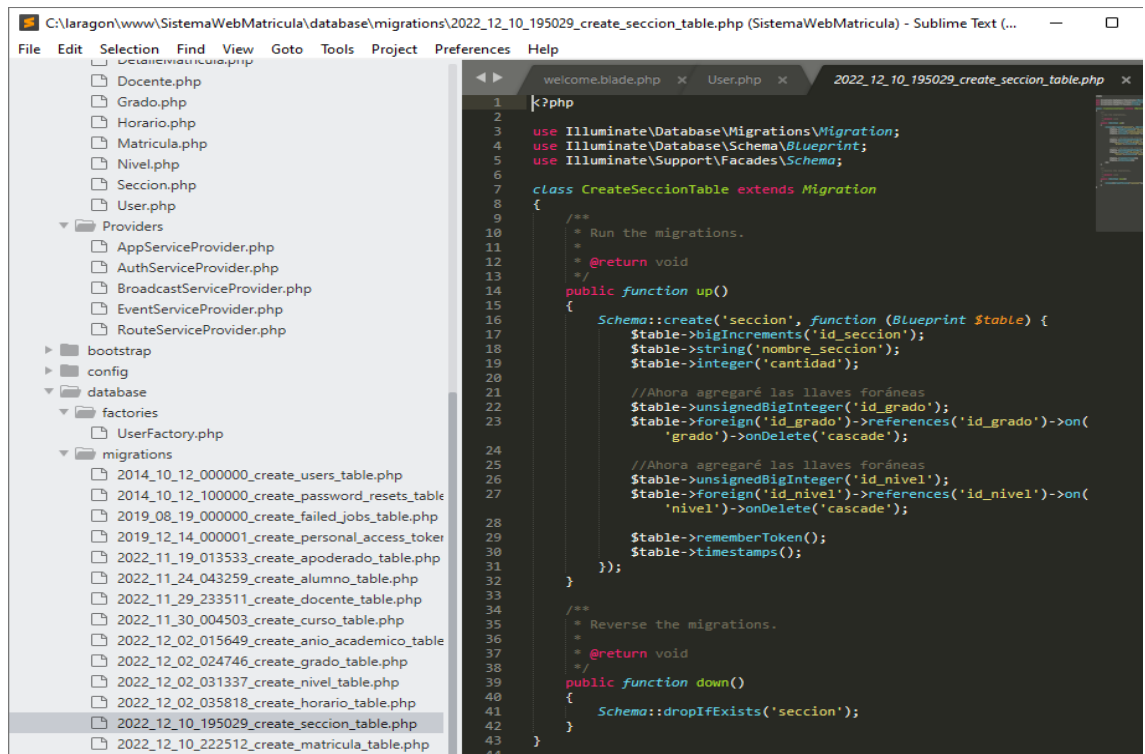
Figura 25: Migración de crear Secciones (Registrar Nivel).



```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateNivelTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      * @return void
12      */
13     public function up()
14     {
15         Schema::create('nivel', function (Blueprint $table) {
16             $table->bigIncrements('id_nivel');
17             $table->string('nombre');
18             $table->rememberToken();
19             $table->timestamps();
20         });
21     }
22
23     /**
24      * Reverse the migrations.
25      * @return void
26      */
27     public function down()
28     {
29         Schema::dropIfExists('nivel');
30     }
31 }
32
33
34
```

Nota. Código PHP de la migración crear Secciones (Registrar Nivel).

Figura 26: Migración de crear Secciones (Registrar Sección).



```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateSeccionTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      * @return void
12      */
13     public function up()
14     {
15         Schema::create('seccion', function (Blueprint $table) {
16             $table->bigIncrements('id_seccion');
17             $table->string('nombre');
18             $table->integer('cantidad');
19
20             //Ahora agregaré las llaves foráneas
21             $table->unsignedBigInteger('id_grado');
22             $table->foreign('id_grado')->references('id_grado')->on(
23                 'grado')->onDelete('cascade');
24
25             //Ahora agregaré las llaves foráneas
26             $table->unsignedBigInteger('id_nivel');
27             $table->foreign('id_nivel')->references('id_nivel')->on(
28                 'nivel')->onDelete('cascade');
29
30             $table->rememberToken();
31             $table->timestamps();
32         });
33     }
34
35     /**
36      * Reverse the migrations.
37      * @return void
38      */
39     public function down()
40     {
41         Schema::dropIfExists('seccion');
42     }
43 }
44
```

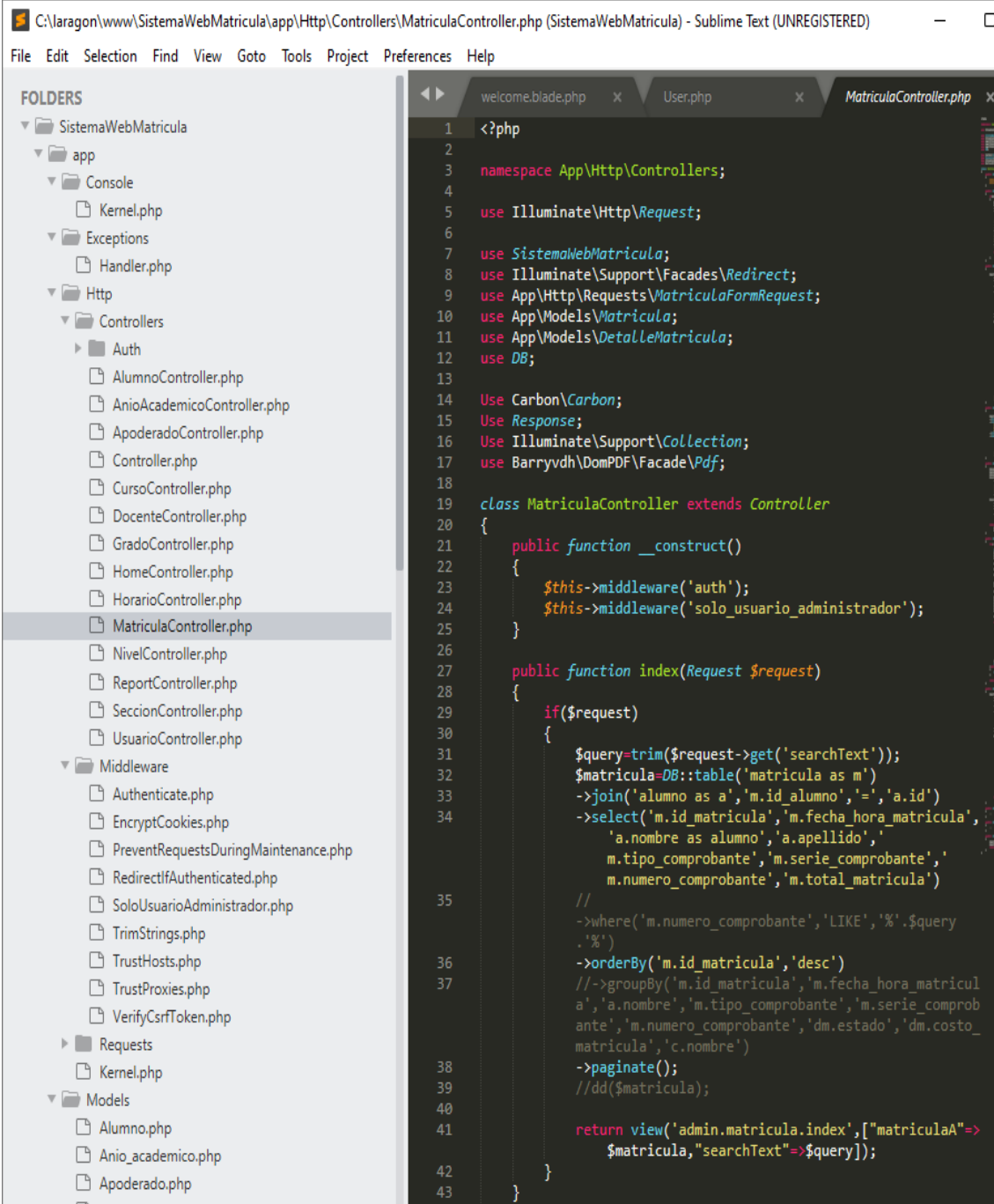
Nota. Código PHP de la migración crear Secciones (Registrar Sección).

Figura 27: Controladores.

```
C:\laragon\www\SistemaWebMatricula>
C:\laragon\www\SistemaWebMatricula>php artisan make:controller UsuarioController
```

Nota. Código en Sublim Text de cada migración hecha anteriormente.

Figura 28: Controlador de Matrícula (Registrar Matrícula).



```
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;

use SistemaWebMatricula;
use Illuminate\Support\Facades\Redirect;
use App\Http\Requests\MatriculaFormRequest;
use App\Models\Matricula;
use App\Models\DetalleMatricula;
use DB;

Use Carbon\Carbon;
Use Response;
Use Illuminate\Support\Collection;
use Barryvdh\DomPDF\Facade\Pdf;

class MatriculaController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
        $this->middleware('solo_usuario_administrador');
    }

    public function index(Request $request)
    {
        if($request)
        {
            $query=trim($request->get('searchText'));
            $matricula=DB::table('matricula as m')
            ->join('alumno as a','m.id_alumno','=','a.id')
            ->select('m.id_matricula','m.fecha_hora_matricula',
            'a.nombre as alumno','a.apellido','
            m.tipo_comprobante','m.serie_comprobante','
            m.numero_comprobante','m.total_matricula')

            //
            ->where('m.numero_comprobante','LIKE','%'.$query
            .'%')
            ->orderBy('m.id_matricula','desc')
            //->groupBy('m.id_matricula','m.fecha_hora_matricula',
            'a.nombre','m.tipo_comprobante','m.serie_comprobante',
            'm.numero_comprobante','dm.estado','dm.costo_matricula',
            'c.nombre')
            ->paginate();
            //dd($matricula);

            return view('admin.matricula.index',['matriculaA'=>
            $matricula,"searchText"=>$query]);
        }
    }
}
```

Nota. Código PHP del controlador Matrícula (Registrar Matrícula).

Figura 29: Controlador de Matrícula (Registrar Matrícula).

```

45 public function pdf($matriculas)
46 {
47     $matricula=DB::table('matricula as m')
48     ->join('alumno as a','m.id_alumno','=', 'a.id')
49     ->join('detalle_matricula as dm','m.id_matricula','
50     =','dm.id_matricula')
51     ->join('apoderado as ap','ap.id','=', 'a.apoderado')
52     ->select('m.id_matricula','m.fecha_hora_matricula',
53     'a.nombre','a.apellido','m.tipo_comprobante',
54     'm.serie_comprobante','m.numero_comprobante',
55     'ap.nombre as apoderado','ap.apellido as
56     apoderadoB','m.total_matricula','
57     m.fecha_hora_matricula','a.email','a.DNI','
58     a.fecha_nac','a.sexo','a.direccion','a.celular'
59     )
60     ->where('m.id_matricula','=',$matriculas)
61     ->first();
62     $detalles=DB::table('detalle_matricula as dm')
63     ->join('curso as cr','dm.id_curso','=', '
64     cr.id_curso')
65     ->join('anio_academico as aa','
66     dm.id_anio_academico','=', '
67     aa.id_anio_academico')
68     ->join('seccion as sec','dm.id_seccion','=', '
69     sec.id_seccion')
70     ->join('horario as hor','dm.id_horario','=', '
71     hor.id_horario')
72     ->join('grado as gr','gr.id_grado','=', '
73     sec.id_grado')
74     ->join('nivel as nv','nv.id_nivel','=', '
75     sec.id_nivel')
76     ->select('cr.nombre','dm.estado',
77     'dm.costo_matricula','aa.descripcion',
78     'sec.nombre_seccion','hor.hora_inicio',
79     'hor.hora_fin','hor.dia','gr.grado',
80     'nv.nivel')
81     ->where('dm.id_matricula','=',$matriculas)
82     ->get();
83     //dd($detalles);
84     $pdf = PDF::loadView('admin.matricula.pdf',['
85     matriculaA=>$matricula','detallesA=>$
86     detalles']);
87     return $pdf->stream();
88     //return view("admin.matricula.show",["matriculaA"=
89     >$matricula,"detallesA">$detalles]);
90 }

```

Nota. Código PHP del controlador Matrícula (Registrar Matrícula).

Figura 30: Controlador de Matrícula (Registrar Matrícula).

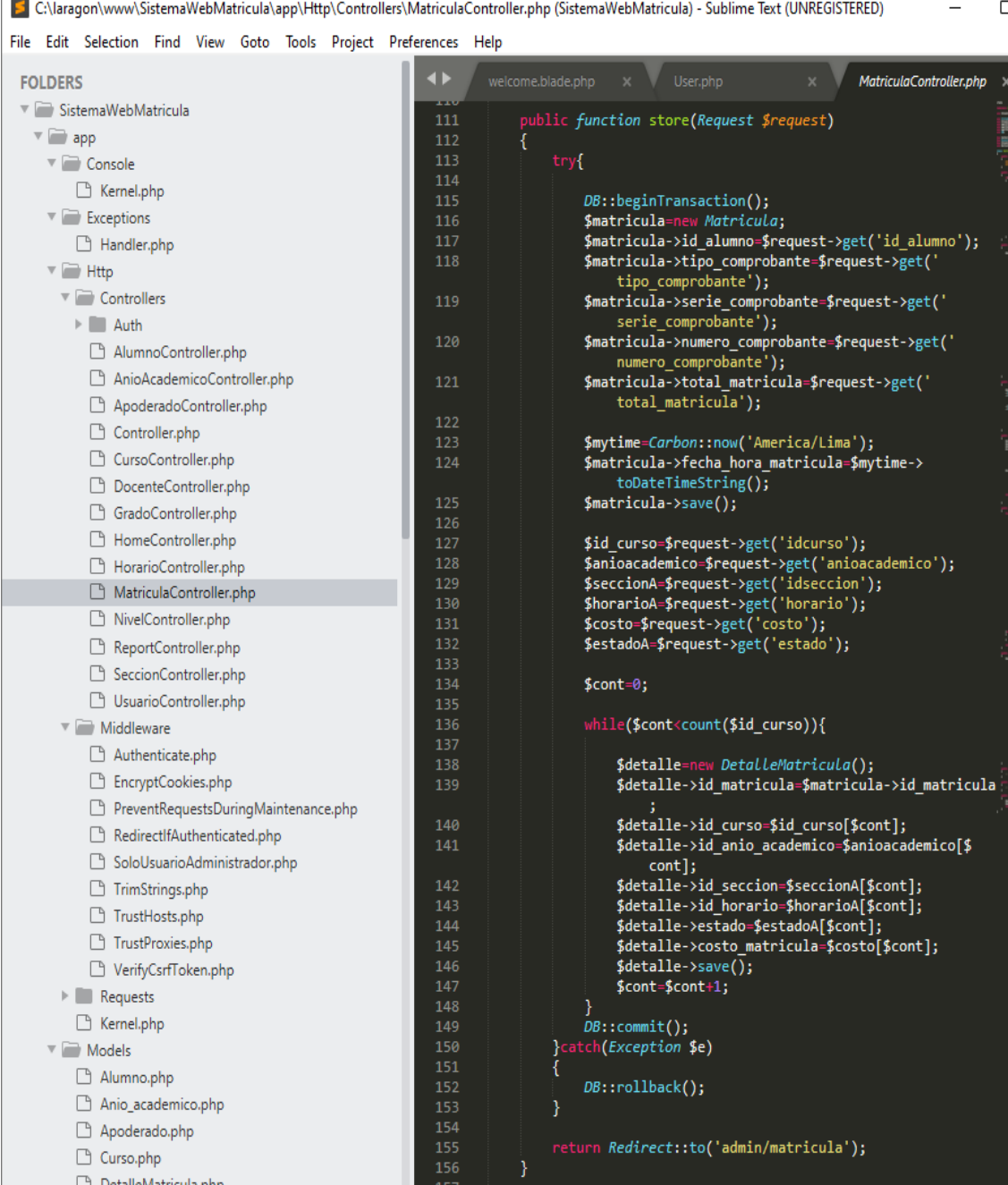
```

73 public function pdfbc()
74 {
75     $matricula=DB::table('matricula as m')
76     ->join('alumno as a','m.id_alumno','=', 'a.id')
77     ->select('m.id_matricula','m.fecha_hora_matricula',
78     'tipo_comprobante','m.serie_comprobante',
79     'm.numero_comprobante','a.nombre','a.apellido',
80     'm.total_matricula')
81     ->orderBy('m.id_matricula','asc')
82     ->paginate(20);
83     $pdf = PDF::loadView('admin.matricula.pdfbc',['
84     matriculaB=>$matricula']);
85     return $pdf->stream();
86     //return view("admin.matricula.show",["matriculaA"=
87     >$matricula,"detallesA">$detalles]);
88 }
89 public function create()
90 {
91     $alumno=DB::table('alumno')->get();
92     $anio=DB::table('anio_academico')->get();
93     $horario=DB::table('horario')->get();
94     $curso=DB::table('curso as c')
95     ->select('c.nombre as curso','c.idcurso','
96     c.cantidad')
97     //->where('cr.estado','=','Activo')
98     //->where('s.cantidad','>','0')
99     ->groupBy('curso','cr.id','s.cantidad')
100     ->get();
101     $seccion=DB::table('seccion as s')
102     ->join('nivel as n','n.id_nivel','=', 's.id_nivel')
103     ->join('grado as g','g.id_grado','=', 's.id_grado')
104     ->select('s.id_seccion','n.nivel','g.grado',
105     's.nombre_seccion')
106     ->get();
107     return view("admin.matricula.create",["alumnoA"=>$
108     alumno,"cursoA"=>$curso,"anioA"=>$anio,"horarioA"=>
109     $horario,"seccionA"=>$seccion]);
110 }

```

Nota. Código PHP del controlador Matrícula (Registrar Matrícula).

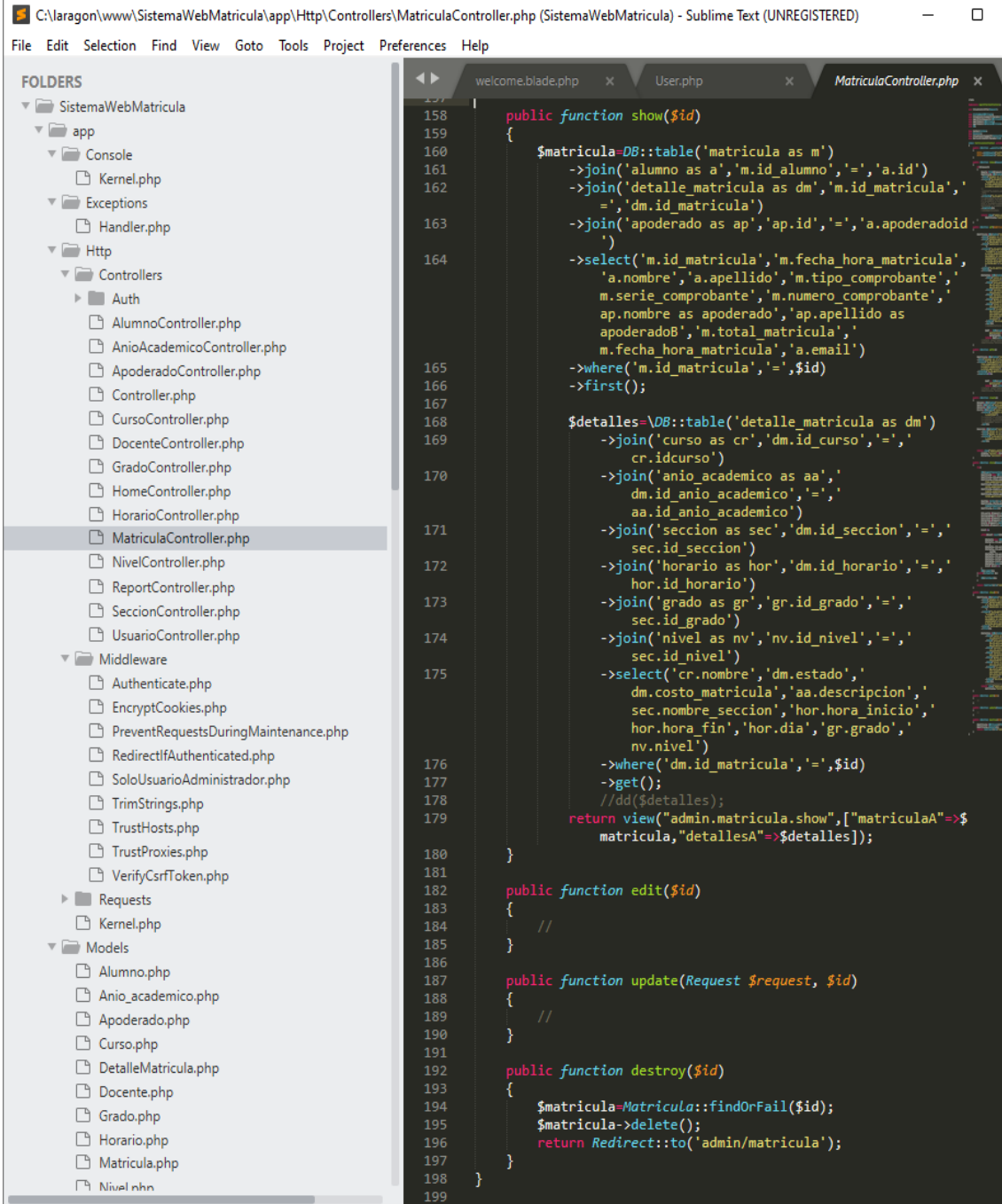
Figura 31: Controlador de Matrícula (Registrar Matrícula).



```
110
111     public function store(Request $request)
112     {
113         try{
114
115             DB::beginTransaction();
116             $matricula=new Matricula;
117             $matricula->id_alumno=$request->get('id_alumno');
118             $matricula->tipo_comprobante=$request->get('
119                 tipo_comprobante');
120             $matricula->serie_comprobante=$request->get('
121                 serie_comprobante');
122             $matricula->numero_comprobante=$request->get('
123                 numero_comprobante');
124             $matricula->total_matricula=$request->get('
125                 total_matricula');
126
127             $mytime=Carbon::now('America/Lima');
128             $matricula->fecha_hora_matricula=$mytime->
129                 toDateTimeString();
130             $matricula->save();
131
132             $id_curso=$request->get('idcurso');
133             $anioacademico=$request->get('anioacademico');
134             $seccionA=$request->get('idseccion');
135             $horarioA=$request->get('horario');
136             $costo=$request->get('costo');
137             $estadoA=$request->get('estado');
138
139             $cont=0;
140             while($cont<count($id_curso)){
141                 $detalle=new DetalleMatricula();
142                 $detalle->id_matricula=$matricula->id_matricula
143                     ;
144                 $detalle->id_curso=$id_curso[$cont];
145                 $detalle->id_anio_academico=$anioacademico[$
146                     cont];
147                 $detalle->id_seccion=$seccionA[$cont];
148                 $detalle->id_horario=$horarioA[$cont];
149                 $detalle->estado=$estadoA[$cont];
150                 $detalle->costo_matricula=$costo[$cont];
151                 $detalle->save();
152                 $cont=$cont+1;
153             }
154             DB::commit();
155         }catch(Exception $e)
156         {
157             DB::rollback();
158         }
159         return Redirect::to('admin/matricula');
```

Nota. Código PHP del controlador Matrícula (Registrar Matrícula).

Figura 32: Controlador de Matrícula (Registrar Matrícula).



```
157
158 public function show($id)
159 {
160     $matricula=DB::table('matricula as m')
161     ->join('alumno as a','m.id_alumno','=', 'a.id')
162     ->join('detalle_matricula as dm','m.id_matricula','
163     =','dm.id_matricula')
164     ->join('apoderado as ap','ap.id','=', 'a.apoderadoi
165     ')
166     ->select('m.id_matricula','m.fecha_hora_matricula',
167     'a.nombre','a.apellido','m.tipo_comprobante','
168     m.serie_comprobante','m.numero_comprobante','
169     ap.nombre as apoderado','ap.apellido as
170     apoderadoB','m.total_matricula','
171     m.fecha_hora_matricula','a.email')
172     ->where('m.id_matricula','=',$id)
173     ->first();
174
175     $detalles=DB::table('detalle_matricula as dm')
176     ->join('curso as cr','dm.id_curso','=','
177     cr.idcurso')
178     ->join('anio_academico as aa','
179     dm.id_anio_academico','=','
180     aa.id_anio_academico')
181     ->join('seccion as sec','dm.id_seccion','=','
182     sec.id_seccion')
183     ->join('horario as hor','dm.id_horario','=','
184     hor.id_horario')
185     ->join('grado as gr','gr.id_grado','=','
186     sec.id_grado')
187     ->join('nivel as nv','nv.id_nivel','=','
188     sec.id_nivel')
189     ->select('cr.nombre','dm.estado','
190     dm costo_matricula','aa.descripcion','
191     sec.nombre seccion','hor.hora_inicio','
192     hor.hora_fin','hor.dia','gr.grado','
193     nv.nivel')
194     ->where('dm.id_matricula','=',$id)
195     ->get();
196     //dd($detalles);
197     return view("admin.matricula.show",["matriculaA"=>$
198     matricula,"detallesA"=>$detalles]);
199
200 }
201
202 public function edit($id)
203 {
204     //
205 }
206
207 public function update(Request $request, $id)
208 {
209     //
210 }
211
212 public function destroy($id)
213 {
214     $matricula=Matricula::findOrFail($id);
215     $matricula->delete();
216     return Redirect::to('admin/matricula');
217 }
218
219 }
```

Nota. Código PHP del controlador Matrícula (Registrar Matrícula).

Figura 33: Controlador de Usuario (Perfil de usuario).

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 use App\Models\User; //Indicar la clase
8 use Illuminate\Support\Facades\Redirect;
9 use App\Http\Requests\UsuarioRequest; //=====USAMOS
10 LA VALIDACION DEL Request=====
11
12 class UsuarioController extends Controller
13 {
14     //DECLARAMOS UN CONSTRUCTOR QUE GESTIONE EL ACCESO POR
15     //USUARIOS
16     public function __construct()
17     {
18         $this->middleware('auth');
19         $this->middleware('solo_usuario_administrador');
20     }
21
22     public function index(Request $request)
23     {
24         if($request)
25         {
26             $query=trim($request->get('searchText'));
27             $usuario=\DB::table('users')->where('name','LIKE','%
28             $query.%')
29             ->orderBy('id','desc')
30             ->paginate(7);
31             $userCount = User::count();
32
33             return view('admin.usuario.index',['usuarioA'=>$
34             usuario,"usuariob"=>$userCount,"searchText"=>$
35             query]);
36         }
37     }
38
39     public function create()
40     {
41         return view("admin.usuario.create");
42     }
43
44     public function store(UsuarioRequest $request)
45     {
46         $usuario = new User;
47         $usuario->name=$request->get('name');
48         $usuario->email=$request->get('email');
49         $usuario->password=bcrypt($request->get('password'));
50         $usuario->type=$request->get('type');
51         $usuario->save();
52
53         //flash('Se ha registrado el usuario '. $usuario->name
54         . ' de forma exitos!')->success();
55         return Redirect::to('admin/usuario');
```

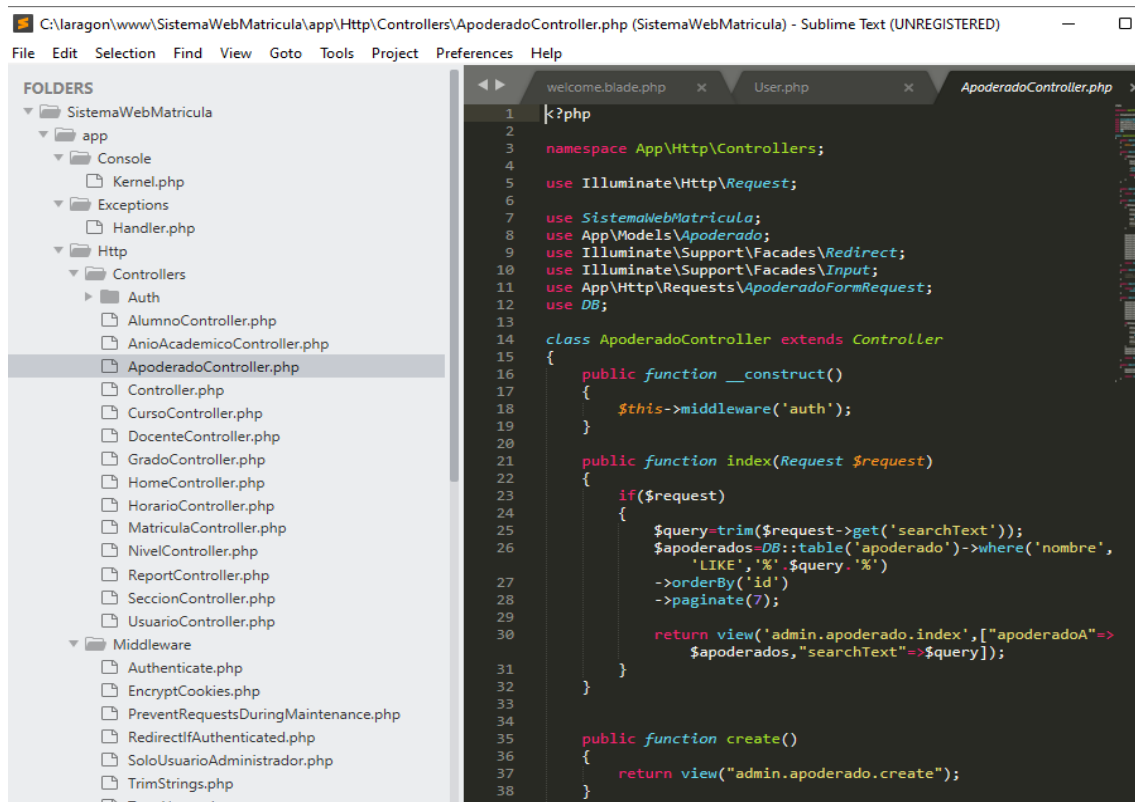
Nota. Código PHP del controlador Usuario (Perfil de Usuario).

Figura 34: Controlador de Usuario (Perfil de usuario).

```
52
53 public function show($id)
54 {
55     //
56 }
57
58 public function edit($id)
59 {
60     return view('admin.usuario.edit',['usuario'=>User::
61     findOrFail($id)];
62 }
63
64 public function update(Request $request, $id)
65 {
66     $usuario = User::findOrFail($id);
67     $usuario->name=$request->get('name');
68     $usuario->email=$request->get('email');
69     $usuario->password=bcrypt($request->get('password'));
70     $usuario->type=$request->get('type');
71     $usuario->update();
72
73     //flash('El usuario '. $usuario->name . ' ha sido
74     editado de forma exitos!')->warning();
75     return Redirect::to('admin/usuario');
76 }
77
78 public function destroy($id)
79 {
80     $usuario = \DB::table('users')->where('id','=', $id)->
81     delete();
82
83     //ahora generamos un mensaje con la clase flash
84     //flash('El usuario '. $usuario->name . ' ha sido
85     borrado de forma exitos!')->error();
86     return Redirect::to('admin/usuario');
```

Nota. Código PHP del controlador Usuario (Perfil de Usuario).

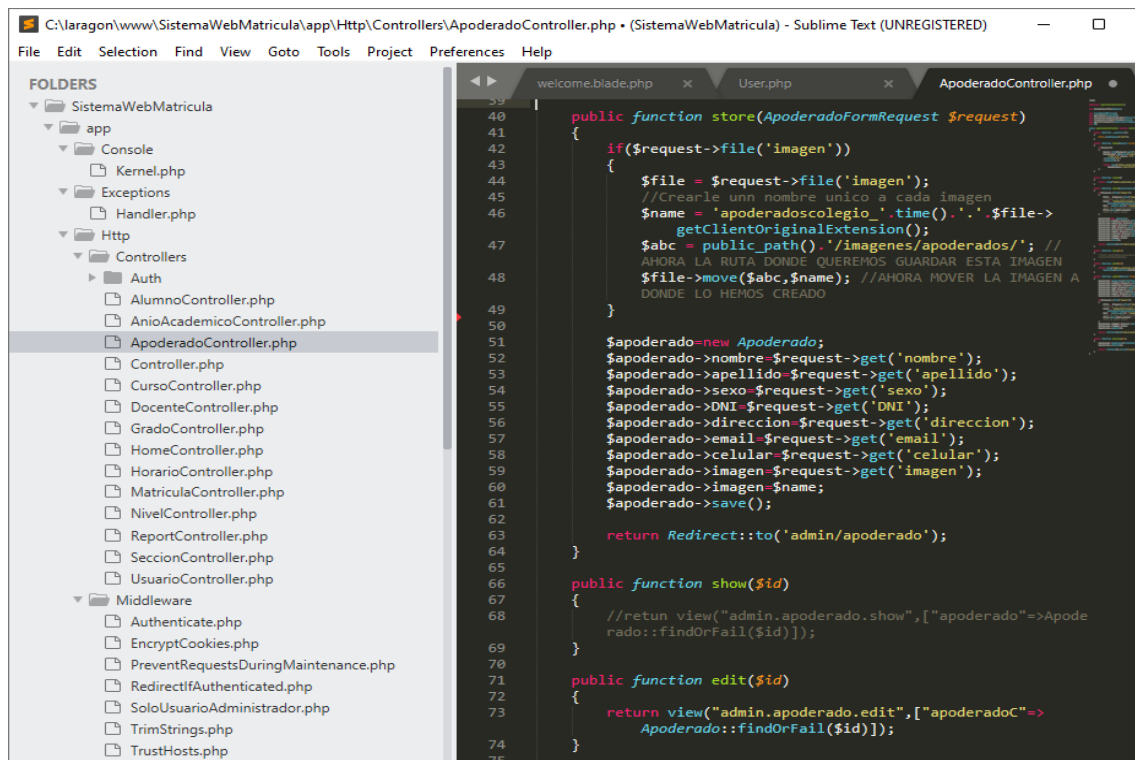
Figura 35: Controlador de Apoderado (Registrar Apoderado).



```
1 |<?php
2
3 | namespace App\Http\Controllers;
4
5 | use Illuminate\Http\Request;
6
7 | use SistemaWebMatricula;
8 | use App\Models\Apoderado;
9 | use Illuminate\Support\Facades\Redirect;
10 | use Illuminate\Support\Facades\Input;
11 | use App\Http\Requests\ApoderadoFormRequest;
12 | use DB;
13
14 | class ApoderadoController extends Controller
15 | {
16 |     public function __construct()
17 |     {
18 |         $this->middleware('auth');
19 |     }
20
21 |     public function index(Request $request)
22 |     {
23 |         if($request)
24 |         {
25 |             $query=trim($request->get('searchText'));
26 |             $apoderados=DB::table('apoderado')->where('nombre',
27 |                 'LIKE','%'.$query.'%')
28 |                 ->orderBy('id')
29 |                 ->paginate(7);
30
31 |             return view('admin.apoderado.index',['apoderadoA'=>
32 |                 $apoderados,"searchText"=>$query]);
33 |         }
34 |     }
35
36 |     public function create()
37 |     {
38 |         return view("admin.apoderado.create");
39 |     }
40 | }
```

Nota. Código PHP del controlador Apoderado (Registrar Apoderado).

Figura 36: Controlador de Apoderado (Registrar Apoderado).



```
40 |
41 |     public function store(ApoderadoFormRequest $request)
42 |     {
43 |         if($request->file('imagen'))
44 |         {
45 |             $file = $request->file('imagen');
46 |             //Crearle un nombre unico a cada imagen
47 |             $name = 'apoderadoscolegio_' .time().'-'. $file->
48 |                 getClientOriginalExtension();
49 |             $abc = public_path().'imagenes/apoderados/'; //
50 |             AHORA LA RUTA DONDE QUEREMOS GUARDAR ESTA IMAGEN
51 |             $file->move($abc,$name); //AHORA MOVER LA IMAGEN A
52 |             DONDE LO HEMOS CREADO
53 |         }
54
55 |         $apoderado=new Apoderado;
56 |         $apoderado->nombre-$request->get('nombre');
57 |         $apoderado->apellido-$request->get('apellido');
58 |         $apoderado->sexo-$request->get('sexo');
59 |         $apoderado->DNI-$request->get('DNI');
60 |         $apoderado->direccion-$request->get('direccion');
61 |         $apoderado->email-$request->get('email');
62 |         $apoderado->celular-$request->get('celular');
63 |         $apoderado->imagen-$request->get('imagen');
64 |         $apoderado->save();
65 |     }
66
67 |     public function show($id)
68 |     {
69 |         //return view("admin.apoderado.show",["apoderado"=>Apode
70 |         rado::findOrFail($id));
71 |     }
72
73 |     public function edit($id)
74 |     {
75 |         return view("admin.apoderado.edit",["apoderadoC"=>
76 |             Apoderado::findOrFail($id)];
77 |     }
78 | }
```

Nota. Código PHP del controlador Apoderado (Registrar Apoderado).

Figura 37: Controlador de Apoderado (Registrar Apoderado).

```
76 public function update(Request $request, $id)
77 {
78     $apoderado=Apoderado::findOrFail($id);
79     $apoderado->nombre=$request->get('nombre');
80     $apoderado->apellido=$request->get('apellido');
81     $apoderado->sexo=$request->get('sexo');
82     $apoderado->DNI=$request->get('DNI');
83     $apoderado->direccion=$request->get('direccion');
84     $apoderado->email=$request->get('email');
85     $apoderado->celular=$request->get('celular');
86
87     if($request->file('imagen'))
88     {
89         $file = $request->file('imagen');
90         //Crearle un nombre unico a cada imagen
91         $name = 'apoderadoscolegio_' . time() . '.' . $file->
92             getClientOriginalExtension();
93         $abc = public_path() . '/imagenes/apoderados/'; //
94             AHORA LA RUTA DONDE QUEREMOS GUARDAR ESTA IMAGEN
95             $file->move($abc,$name); //AHORA MOVER LA IMAGEN A
96             DONDE LO HEAMOS CREADO
97     }
98     $apoderado->imagen=$request->get('imagen');
99     $apoderado->imagen=$name;
100     $apoderado->update();
101     return Redirect::to('admin/apoderado');
102 }
103
104 public function destroy($id)
105 {
106     $apoderado=Apoderado::findOrFail($id);
107     $apoderado->delete();
108     return redirect()->route('apoderado.index');
109 }
110 }
```

Nota. Código PHP del controlador Apoderado (Registrar Apoderado).

Figura 38: Controlador de Alumno (Registrar Alumno).

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 use SistemaWebMatricula;
8 use App\Models\Alumno;
9 use App\Models\Apoderado;
10 use Illuminate\Support\Facades\Redirect;
11 use App\Http\Requests\AlumnoFormRequest;
12 use DB;
13 use Barryvdh\DomPDF\Facade\Pdf;
14
15 class AlumnoController extends Controller
16 {
17     public function __construct()
18     {
19         $this->middleware('auth');
20     }
21
22     public function index(Request $request)
23     {
24         if($request)
25         {
26             $query=trim($request->get('searchText'));
27             $alumno=DB::table('alumno as a')
28                 ->join('apoderado as ap','ap.id','=','a.apoderadoid')
29                 ->select('a.id','a.nombre','a.apellido','a.email','
30                     a.DNI','a.fecha_nac','a.sexo','a.direccion','
31                     a.celular','a.imagen','ap.nombre as apoderado')
32                 ->where('a.nombre','LIKE','%'.$query.'%')
33                 ->orderBy('a.id','asc')
34                 ->paginate(7);
35             //dd($alumno);
36             return view('admin.alumno.index',['alumnoA'=>$
37                 alumno,"searchText"=>$query]);
38         }
39     }
40
41     public function pdf()
42     {
43         $alumnos=DB::table('alumno as a')
44             ->join('apoderado as ap','ap.id','=','a.apoderadoid')
45             ->select('a.id','a.nombre','a.apellido','a.email','
46                 a.DNI','a.fecha_nac','a.sexo','a.direccion','
47                 a.celular','ap.nombre as apoderado')
48             ->orderBy('a.id','asc')
49             ->paginate(7);
50
51         $pdf = PDF::loadView('admin.alumno.pdf',['alumnoA'=>$
52             alumnos]);
53         return $pdf->stream();
54         //return
55         view("admin.alumno.pdf",["alumnoA"=>$alumnos]);
56     }
57 }
```

Nota. Código PHP del controlador Alumno (Registrar Alumno).

Figura 39: Controlador de Alumno (Registrar Alumno).

```

52 public function create()
53 {
54     $apoderado=DB::table('apoderado')->get();
55     // $apoderado=Apoderado::orderBy('id','ASC')->pluck('nombre','id');
56     //dd($apoderado);
57     return view("admin.alumno.create",["apoderadoA"=>$apoderado]);
58 }
59
60
61 public function store(AlumnoFormRequest $request)
62 {
63     // $alumno=new Alumno;
64     $alumno->apoderadoid=$request->get('id');
65     $alumno->nombre=$request->get('nombre');
66     $alumno->apellido=$request->get('apellido');
67     $alumno->email=$request->get('email');
68     $alumno->DNI=$request->get('DNI');
69     $alumno->fecha_nac=$request->get('fecha_nac');
70     $alumno->sexo=$request->get('sexo');
71     $alumno->direccion=$request->get('direccion');
72     $alumno->celular=$request->get('celular');
73
74     if(Input::hasFile('imagen')){
75         $file=Input::file('imagen');
76         $file->move(public_path().'/imagenes/
77             alumnos/', $file->getClientOriginalName());
78         $alumno->imagen=$file->getClientOriginalName();
79     }
80
81     $alumno->save();
82     return Redirect::to('admin/alumno');
83 }
84
85 public function file($request)
86 {
87     $file = $request->file('imagen');
88     $name = 'alumno.'.time().'.'.$file->
89         getClientOriginalExtension();
90     $abc = public_path().'/imagenes/alumnos/'; //AHORA LA
91         RUTA DONDE QUEREMOS GUARDAR ESTA IMAGEN
92     $file->move($abc,$name); //AHORA MOVER LA IMAGEN A
93         DONDE LO HEMOS CREADO
94 }
95
96 //CREAMOS UN OBJETO PARA GUARDAR EL "ARTICULO" Y LA IMAGEN
97 $alumno = new Alumno($request->all());
98 // $alumno->apoderadoid = Auth::apoderado()->id;
99 $alumno->imagen=$name;
100 $alumno->save();
101
102 //Falta que a la hora de guardar se vaya a la pagina
103 //principal
104 //Flash("El articulo ". $a->nombre . " ha sido creada con
105 //éxito!")->success();
106 //AHORA PODEMOS RETORNAR UNA REDIRECCIÓN A LA RUTA
107 SIGUIENTE:
108 //Flash("Se ha registrado el empleado ". $a->nombre . " de
109 //forma exitosa!")->success();
110 //dd($alumno);
111 return redirect()->route('alumno.index');
112 }
113
114 }
115
116 }
117
118 }
119
120 }
121
122 }
123
124 }
125
126 }
127
128 }
129
130 }
131
132 }
133
134 }
135
136 }
137
138 }
139
140 }
141
142 }
143
144 }
145
146 }
147
148 }
149
150 }
151
152 }
153
154 }
155
156 }
157
158 }
159
160 }
161
162 }
163
164 }
165
166 }
167
168 }
169
170 }
171
172 }
173
174 }
175
176 }
177
178 }
179
180 }
181
182 }
183
184 }
185
186 }
187
188 }
189
190 }
191
192 }
193
194 }
195
196 }
197
198 }
199
200 }
201
202 }
203
204 }
205
206 }
207
208 }
209
210 }
211
212 }
213
214 }
215
216 }
217
218 }
219
220 }
221
222 }
223
224 }
225
226 }
227
228 }
229
230 }
231
232 }
233
234 }
235
236 }
237
238 }
239
240 }
241
242 }
243
244 }
245
246 }
247
248 }
249
250 }
251
252 }
253
254 }
255
256 }
257
258 }
259
260 }
261
262 }
263
264 }
265
266 }
267
268 }
269
270 }
271
272 }
273
274 }
275
276 }
277
278 }
279
280 }
281
282 }
283
284 }
285
286 }
287
288 }
289
290 }
291
292 }
293
294 }
295
296 }
297
298 }
299
300 }
301
302 }
303
304 }
305
306 }
307
308 }
309
310 }
311
312 }
313
314 }
315
316 }
317
318 }
319
320 }
321
322 }
323
324 }
325
326 }
327
328 }
329
330 }
331
332 }
333
334 }
335
336 }
337
338 }
339
340 }
341
342 }
343
344 }
345
346 }
347
348 }
349
350 }
351
352 }
353
354 }
355
356 }
357
358 }
359
360 }
361
362 }
363
364 }
365
366 }
367
368 }
369
370 }
371
372 }
373
374 }
375
376 }
377
378 }
379
380 }
381
382 }
383
384 }
385
386 }
387
388 }
389
390 }
391
392 }
393
394 }
395
396 }
397
398 }
399
400 }
401
402 }
403
404 }
405
406 }
407
408 }
409
410 }
411
412 }
413
414 }
415
416 }
417
418 }
419
420 }
421
422 }
423
424 }
425
426 }
427
428 }
429
430 }
431
432 }
433
434 }
435
436 }
437
438 }
439
440 }
441
442 }
443
444 }
445
446 }
447
448 }
449
450 }
451
452 }
453
454 }
455
456 }
457
458 }
459
460 }
461
462 }
463
464 }
465
466 }
467
468 }
469
470 }
471
472 }
473
474 }
475
476 }
477
478 }
479
480 }
481
482 }
483
484 }
485
486 }
487
488 }
489
490 }
491
492 }
493
494 }
495
496 }
497
498 }
499
500 }
501
502 }
503
504 }
505
506 }
507
508 }
509
510 }
511
512 }
513
514 }
515
516 }
517
518 }
519
520 }
521
522 }
523
524 }
525
526 }
527
528 }
529
530 }
531
532 }
533
534 }
535
536 }
537
538 }
539
540 }
541
542 }
543
544 }
545
546 }
547
548 }
549
550 }
551
552 }
553
554 }
555
556 }
557
558 }
559
560 }
561
562 }
563
564 }
565
566 }
567
568 }
569
570 }
571
572 }
573
574 }
575
576 }
577
578 }
579
580 }
581
582 }
583
584 }
585
586 }
587
588 }
589
590 }
591
592 }
593
594 }
595
596 }
597
598 }
599
600 }
601
602 }
603
604 }
605
606 }
607
608 }
609
610 }
611
612 }
613
614 }
615
616 }
617
618 }
619
620 }
621
622 }
623
624 }
625
626 }
627
628 }
629
630 }
631
632 }
633
634 }
635
636 }
637
638 }
639
640 }
641
642 }
643
644 }
645
646 }
647
648 }
649
650 }
651
652 }
653
654 }
655
656 }
657
658 }
659
660 }
661
662 }
663
664 }
665
666 }
667
668 }
669
670 }
671
672 }
673
674 }
675
676 }
677
678 }
679
680 }
681
682 }
683
684 }
685
686 }
687
688 }
689
690 }
691
692 }
693
694 }
695
696 }
697
698 }
699
700 }
701
702 }
703
704 }
705
706 }
707
708 }
709
710 }
711
712 }
713
714 }
715
716 }
717
718 }
719
720 }
721
722 }
723
724 }
725
726 }
727
728 }
729
730 }
731
732 }
733
734 }
735
736 }
737
738 }
739
740 }
741
742 }
743
744 }
745
746 }
747
748 }
749
750 }
751
752 }
753
754 }
755
756 }
757
758 }
759
760 }
761
762 }
763
764 }
765
766 }
767
768 }
769
770 }
771
772 }
773
774 }
775
776 }
777
778 }
779
780 }
781
782 }
783
784 }
785
786 }
787
788 }
789
790 }
791
792 }
793
794 }
795
796 }
797
798 }
799
800 }
801
802 }
803
804 }
805
806 }
807
808 }
809
810 }
811
812 }
813
814 }
815
816 }
817
818 }
819
820 }
821
822 }
823
824 }
825
826 }
827
828 }
829
830 }
831
832 }
833
834 }
835
836 }
837
838 }
839
840 }
841
842 }
843
844 }
845
846 }
847
848 }
849
850 }
851
852 }
853
854 }
855
856 }
857
858 }
859
860 }
861
862 }
863
864 }
865
866 }
867
868 }
869
870 }
871
872 }
873
874 }
875
876 }
877
878 }
879
880 }
881
882 }
883
884 }
885
886 }
887
888 }
889
890 }
891
892 }
893
894 }
895
896 }
897
898 }
899
900 }
901
902 }
903
904 }
905
906 }
907
908 }
909
910 }
911
912 }
913
914 }
915
916 }
917
918 }
919
920 }
921
922 }
923
924 }
925
926 }
927
928 }
929
930 }
931
932 }
933
934 }
935
936 }
937
938 }
939
940 }
941
942 }
943
944 }
945
946 }
947
948 }
949
950 }
951
952 }
953
954 }
955
956 }
957
958 }
959
960 }
961
962 }
963
964 }
965
966 }
967
968 }
969
970 }
971
972 }
973
974 }
975
976 }
977
978 }
979
980 }
981
982 }
983
984 }
985
986 }
987
988 }
989
990 }
991
992 }
993
994 }
995
996 }
997
998 }
999
1000 }
1001
1002 }
1003
1004 }
1005
1006 }
1007
1008 }
1009
1010 }
1011
1012 }
1013
1014 }
1015
1016 }
1017
1018 }
1019
1020 }
1021
1022 }
1023
1024 }
1025
1026 }
1027
1028 }
1029
1030 }
1031
1032 }
1033
1034 }
1035
1036 }
1037
1038 }
1039
1040 }
1041
1042 }
1043
1044 }
1045
1046 }
1047
1048 }
1049
1050 }
1051
1052 }
1053
1054 }
1055
1056 }
1057
1058 }
1059
1060 }
1061
1062 }
1063
1064 }
1065
1066 }
1067
1068 }
1069
1070 }
1071
1072 }
1073
1074 }
1075
1076 }
1077
1078 }
1079
1080 }
1081
1082 }
1083
1084 }
1085
1086 }
1087
1088 }
1089
1090 }
1091
1092 }
1093
1094 }
1095
1096 }
1097
1098 }
1099
1100 }
1101
1102 }
1103
1104 }
1105
1106 }
1107
1108 }
1109
1110 }
1111
1112 }
1113
1114 }
1115
1116 }
1117
1118 }
1119
1120 }
1121
1122 }
1123
1124 }
1125
1126 }
1127
1128 }
1129
1129 }
1130
1130 }
1131
1131 }
1132
1132 }
1133
1133 }
1134
1134 }
1135
1135 }
1136
1136 }
1137
1137 }
1138
1138 }
1139
1139 }
1140
1140 }
1141
1141 }
1142
1142 }
1143
1143 }
1144
1144 }
1145
1145 }
1146
1146 }
1147
1147 }
1148
1148 }
1149
1149 }
1150
1150 }
1151
1151 }
1152
1152 }
1153
1153 }
1154
1154 }
1155
1155 }
1156
1156 }
1157
1157 }
1158
1158 }
1159
1159 }
1160
1160 }
1161
1161 }
1162
1162 }
1163
1163 }
1164
1164 }
1165
1165 }
1166
1166 }
1167
1167 }
1168
1168 }
1169
1169 }
1170
1170 }
1171
1171 }
1172
1172 }
1173
1173 }
1174
1174 }
1175
1175 }
1176
1176 }
1177
1177 }
1178
1178 }
1179
1179 }
1180
1180 }
1181
1181 }
1182
1182 }
1183
1183 }
1184
1184 }
1185
1185 }
1186
1186 }
1187
1187 }
1188
1188 }
1189
1189 }
1190
1190 }
1191
1191 }
1192
1192 }
1193
1193 }
1194
1194 }
1195
1195 }
1196
1196 }
1197
1197 }
1198
1198 }
1199
1199 }
1200
1200 }
1201
1201 }
1202
1202 }
1203
1203 }
1204
1204 }
1205
1205 }
1206
1206 }
1207
1207 }
1208
1208 }
1209
1209 }
1210
1210 }
1211
1211 }
1212
1212 }
1213
1213 }
1214
1214 }
1215
1215 }
1216
1216 }
1217
1217 }
1218
1218 }
1219
1219 }
1220
1220 }
1221
1221 }
1222
1222 }
1223
1223 }
1224
1224 }
1225
1225 }
1226
1226 }
1227
1227 }
1228
1228 }
1229
1229 }
1230
1230 }
1231
1231 }
1232
1232 }
1233
1233 }
1234
1234 }
1235
1235 }
1236
1236 }
1237
1237 }
1238
1238 }
1239
1239 }
1240
1240 }
1241
1241 }
1242
1242 }
1243
1243 }
1244
1244 }
1245
1245 }
1246
1246 }
1247
1247 }
1248
1248 }
1249
1249 }
1250
1250 }
1251
1251 }
1252
1252 }
1253
1253 }
1254
1254 }
1255
1255 }
1256
1256 }
1257
1257 }
1258
1258 }
1259
1259 }
1260
1260 }
1261
1261 }
1262
1262 }
1263
1263 }
1264
1264 }
1265
1265 }
1266
1266 }
1267
1267 }
1268
1268 }
1269
1269 }
1270
1270 }
1271
1271 }
1272
1272 }
1273
1273 }
1274
1274 }
1275
1275 }
1276
1276 }
1277
1277 }
1278
1278 }
1279
1279 }
1280
1280 }
1281
1281 }
1282
1282 }
1283
1283 }
1284
1284 }
1285
1285 }
1286
1286 }
1287
1287 }
1288
1288 }
1289
1289 }
1290
1290 }
1291
1291 }
1292
1292 }
1293
1293 }
1294
1294 }
1295
1295 }
1296
1296 }
1297
1297 }
1298
1298 }
1299
1299 }
1300
1300 }
1301
1301 }
1302
1302 }
1303
1303 }
1304
1304 }
1305
1305 }
1306
1306 }
1307
1307 }
1308
1308 }
1309
1309 }
1310
1310 }
1311
1311 }
1312
1312 }
1313
1313 }
1314
1314 }
1315
1315 }
1316
1316 }
1317
1317 }
1318
1318 }
1319
1319 }
1320
1320 }
1321
1321 }
1322
1322 }
1323
1323 }
1324
1324 }
1325
1325 }
1326
1326 }
1327
1327 }
1328
1328 }
1329
1329 }
1330
1330 }
1331
1331 }
1332
1332 }
1333
1333 }
1334
1334 }
1335
1335 }
1336
1336 }
1337
1337 }
1338
1338 }
1339
1339 }
1340
1340 }
1341
1341 }
1342
1342 }
1343
1343 }
1344
1344 }
1345
1345 }
1346
1346 }
1347
1347 }
1348
1348 }
1349
1349 }
1350
1350 }
1351
1351 }
1352
1352 }
1353
1353 }
1354
1354 }
1355
1355 }
1356
1356 }
1357
1357 }
1358
1358 }
1359
1359 }
1360
1360 }
1361
1361 }
1362
1362 }
1363
1363 }
1364
1364 }
1365
1365 }
1366
1366 }
1367
1367 }
1368
1368 }
1369
1369 }
1370
1370 }
1371
1371 }
1372
1372 }
1373
1373 }
1374
1374 }
1375
1375 }
1376
1376 }
1377
1377 }
1378
1378 }
1379
1379 }
1380
1380 }
1381
1381 }
1382
1382 }
1383
1383 }
1384
1384 }
1385
1385 }
1386
1386 }
1387
1387 }
1388
1388 }
1389
1389 }
1390
1390 }
1391
1391 }
1392
1392 }
1393
1393 }
1394
1394 }
1395
1395 }
1396
1396 }
1397
1397 }
1398
1398 }
1399
1399 }
1400
1400 }
1401
1401 }
1402
1402 }
1403
1403 }
1404
1404 }
1405
1405 }
1406
1406 }
1407
1407 }
1408
1408 }
1409
1409 }
1410
1410 }
1411
1411 }
1412
1412 }
1413
1413 }
1414
1414 }
1415
1415 }
1416
1416 }
1417
1417 }
1418
1418 }
1419
1419 }
1420
1420 }
1421
1421 }
1422
1422 }
1423
1423 }
1424
1424 }
1425
1425 }
1426
1426 }
1427
1427 }
1428
1428 }
1429
1429 }
1430
1430 }
1431
1431 }
1432
1432 }
1433
1433 }
1434
1434 }
1435
1435 }
1436
1436 }
1437
1437 }
1438
1438 }
1439
1439 }
1440
1440 }
1441
1441 }
1442
1442 }
1443
1443 }
1444
1444 }
1445
1445 }
1446
1446 }
1447
1447 }
1448
1448 }
1449
1449 }
1450
1450 }
1451
1451 }
1452
1452 }
1453
1453 }
1454
1454 }
1455
1455 }
1456
1456 }
1457
1457 }
1458
1458 }
1459
1459 }
1460
1460 }
1461
1461 }
1462
1462 }
1463
1463 }
1464
1464 }
1465
1465 }
1466
1466 }
1467
1467 }
1468
1468 }
1469
1469 }
1470
1470 }
1471
1471 }
1472
1472 }
1473
1473 }
1474
1474 }
1475
1475 }
1476
1476 }
1477
1477 }
1478
1478 }
1479
1479 }
1480
1480 }
1481
1481 }
1482
1482 }
1483
1483 }
1484
1484 }
1485
1485 }
1486
1486 }
1487
1487 }
1488
1488 }
1489
1489 }
1490
1490 }
1491
1491 }
1492
1492 }
1493
1493 }
1494
1494 }
1495
1495 }
1496
1496 }
1497
1497 }
1498
1498 }
1499
1499 }
1500
1500 }
1501
1501 }
1502
1502 }
1503
1503 }
1504
1504 }
1505
1505 }
1506
1506 }
1507
1507 }
1508
1508 }
1509
1509 }
1510
1510 }
1511
1511 }
1512
1512 }
1513
1513 }
1514
1514 }
1515
1515 }
1516
1516 }
1517
1517 }
1518
1518 }
1519
1519 }
1520
1520 }
1521
1521 }
1522
1522 }
1523
1523 }
1524
1524 }
1525
1525 }
1526
1526 }
1527
1527 }
1528
1528 }
1529
1529 }
1530
1530 }
1531
1531 }
1532
1532 }
1533
1533 }
1534
1534 }
1535
1535 }
1536
1536 }
1537
1537 }
1538
1538 }
1539
1539 }
1540
1540 }
1541
1541 }
1542
1542 }
1543
1543 }
1544
1544 }
1545
1545 }
1546
1546 }
1547
1547 }
1548
1548 }
1549
1549 }
1550
1550 }
1551
1551 }
1552
1552 }
1553
1553 }
1554
1554 }
1555
1555 }
1556
1556 }
1557
1557 }
1558
1558 }
1559
1559 }
1560
1560 }
1561
1561 }
1562
1562 }
1563
1563 }
1564
1564 }
1565
1565 }
1566
1566 }
1567
1567 }
1568
1568 }
1569
1569 }
1570
1570 }
1571
1571 }
1572
1572 }
1573
1573 }
1574
1574 }
1575
1575 }
1576
1576 }
1577
1577 }
1578
1578 }
1579
1579 }
1580
1580 }
1581
1581 }
1582
1582 }
1583
1583 }
1584
1584 }
1585
1585 }
1586
1586 }
1587
1587 }
1588
1588 }
1589
1589 }
1590
1590 }
1591
1591 }
1592
1592 }
1593
1593 }
1594
1594 }
1595
1595 }
1596
1596 }
1597
1597 }
1598
1598 }
1599
1599 }
1600
1600 }
1601
1601 }
1602
1602 }
1603
1603 }
1604
1604 }
1605
1605 }
1606
1606 }
1607
1607 }
1608
1608 }
1609
1609 }
1610
1610 }
1611
1611 }
1612
1612 }
1613
1613 }
1614
1614 }
1615
1615 }
1616
1616 }
1617
1617 }
1618
1618 }
1619
1619 }
1620
1620 }
1621
1621 }
1622
1622 }
1623
1623 }
1624
1624 }
1625
1625 }
1626
1626 }
1627
1627 }
1628
1628 }
1629
1629 }
1630
1630 }
1631
1631 }
1632
1632 }
1633
1633 }
1634
1634 }
1635
1635 }
1636
1636 }
1637
1637 }
1638
1638 }
1639
1639 }
1640
1640 }
1641
1641 }
1642
1642 }
1643
1643 }
1644
1644 }
1645
1645 }
1646
1646 }
1647
1647 }
1648
1648 }
1649
1649 }
1650
1650 }
1651
1651 }
1652
1652 }
1653
1653 }
1654
1654 }
1655
1655 }
1656
1656 }
1657
1657 }
1658
1658 }
1659
1659 }
1660
1660 }
1661
1661 }
1662
1662 }
1663
1663 }
1664
1664 }
1665
1665 }
1666
1666 }
1667
1667 }
1668
1668 }
1669
1669 }
1670
1670 }
1671
1671 }
1672
1672 }
1673
1673 }
1674
1674 }
1675
1675 }
1676
1676 }
1677
1677 }
1678
1678 }
1679
1679 }
1680
1680 }
1681
1681 }
1682
1682 }
1683
1683 }
1684
1684 }
1685
1685 }
1686
1686 }
1687
1687 }
1688
1688 }
1689
1689 }
1690
1690 }
1691
1691 }
1692
1692 }
1693
1693 }
1694
1694 }
1695
1695 }
1696
1696 }
1697
1697 }
1698
1698 }
1699
1699 }
1700
1700 }
1701
1701 }
1702
1702 }
1703
1703 }
1704
1704 }
1705
1705 }
1706
1706 }
1707
1707 }
1708
1708 }
1709
1709 }
1710
1710 }
1711
1711 }
1712
1712 }
1713
1713 }
1714
1714 }
1715
1715 }
1716
1716 }
1717
1717 }
1718
1718 }
1719
1719 }
1720
1720 }
1721
1721 }
1722
1722 }
1723
1723 }
1724
1724 }
1725
1725 }
1726
1726 }
1727
1727 }
1728
1728 }
1729
1729 }
1730
1730 }
1731
1731 }
1732
1732 }
1733
1733 }
1734
1734 }
1735
1735 }
1736
1736 }
1737
1737 }
1738
1738 }
1739
1739 }
1740
1740 }
1741
1741 }

```


Figura 41: Controlador de Docente (Registrar Docente).

```
1 <?php
2
3
4 namespace App\Http\Controllers;
5 use Illuminate\Http\Request;
6
7 use SistemaWebMatricula;
8 use App\Models\Docente;
9 use Illuminate\Support\Facades\Redirect;
10 use Illuminate\Support\Facades\Input;
11 use App\Http\Requests\DocenteFormRequest;
12 use DB;
13
14 class DocenteController extends Controller
15 {
16     public function __construct()
17     {
18         $this->middleware('auth');
19     }
20
21     public function index(Request $request)
22     {
23         $query=trim($request->get('searchText'));
24         $docentes=DB::table('docente')->where('nombre','LIKE','%
25             $query %');
26         ->orderBy('iddocente')
27         ->paginate(7);
28
29         return view('admin.docente.index',['docenteA'=>$docentes
30             ,"searchText"=>$query]);
31     }
32
33
34     public function create()
35     {
36         return view("admin.docente.create");
37     }
38
39     public function store(DocenteFormRequest $request)
40     {
41         $docentes=new Docente;
42         $docentes->nombre=$request->get('nombre');
43         $docentes->apellido=$request->get('apellido');
44         $docentes->sexo=$request->get('sexo');
45         $docentes->descripcion=$request->get('descripcion');
46         $docentes->email=$request->get('email');
47         $docentes->celular=$request->get('celular');
48         $docentes->save();
49
50         return Redirect::to('admin/docente');
51     }
52
53     public function show($id)
54     {
55         //
56     }
57 }
```

Nota. Código PHP del controlador Docente (Registrar Docente).

Figura 42: Controlador de Docente (Registrar Docente).

```
57
58     public function edit($id)
59     {
60         return view("admin.docente.edit",["docenteC"=>$Docente::
61             findOrFail($id)];
62     }
63
64     public function update(Request $request, $id)
65     {
66         $docentes=Docente::findOrFail($id);
67         $docentes->nombre=$request->get('nombre');
68         $docentes->apellido=$request->get('apellido');
69         $docentes->sexo=$request->get('sexo');
70         $docentes->descripcion=$request->get('descripcion');
71         $docentes->email=$request->get('email');
72         $docentes->celular=$request->get('celular');
73
74         $docentes->update();
75
76         return Redirect::to('admin/docente');
77     }
78
79     public function destroy($id)
80     {
81         $docente=Docente::findOrFail($id);
82         $docente->delete();
83
84         return redirect()->route('docente.index');
85     }
86 }
```

Nota. Código PHP del controlador Docente (Registrar Docente).

Figura 43: Controlador de Cursos (Registrar Cursos).

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  use SistemaWebMatricula;
8  use App\Models\Curso;
9  use App\Models\Docente;
10 use Illuminate\Support\Facades\Redirect;
11 use App\Http\Requests\CursoFormRequest;
12 use DB;
13
14 class CursoController extends Controller
15 {
16     public function __construct()
17     {
18         $this->middleware('auth');
19     }
20
21     public function index(Request $request)
22     {
23         if($request)
24         {
25             $query=trim($request->get('searchText'));
26             $curso=DB::table('curso as curs')
27                 ->join('docente as doc','curs.docente_id','=','doc.iddocente')
28                 ->select('curs.idcurso','curs.nombre','curs.descripcion',
29                     'doc.nombre as docente','curs.cantidad')
30                 ->where('curs.nombre','LIKE','%'.$query.'%')
31                 ->orderBy('curs.idcurso','asc')
32                 ->paginate(7);
33             //dd($curso);
34
35             return view('admin.curso.index',['cursoA'=>$curso,"
36                 searchText"=>$query]);
37         }
38     }
39
40     public function create()
41     {
42         $docente=DB::table('docente')->get();
43         //dd($docente);
44         //dd($apoderado);
45         return view("admin.curso.create",["docenteA"=>$docente]);
46     }
47
48     public function store(CursoFormRequest $request)
49     {
50         //dd($request);
51         //dd($request->all());
52         //dd($request->toArray());
53         $curso=new Curso;
54         $curso->nombre=$request->get('nombre');
55         $curso->cantidad=$request->get('cantidad');
56         $curso->docente_id=$request->get('docente_id');
57         $curso->descripcion=$request->get('descripcion');
58
59         $curso->save();
60
61         //dd($curso);
62         return redirect()->route('curso.index');
63     }
64
65     public function show($id)
66     {
67         //dd($id);
68     }
69
70     public function edit($id)
71     {
72         $curso=Curso::findOrFail($id);
73         $docentes=DB::table('docente')->get();
74         //dd($curso);
75         return view("admin.curso.edit",["cursoC"=>$curso,"docenteA"
76             =>$docentes]);
77     }
78
79     public function update(CursoFormRequest $request, $id)
80     {
81         $curso=Curso::findOrFail($id);
82         $curso->docente_id=$request->get('docente_id');
83         $curso->cantidad=$request->get('cantidad');
84         $curso->nombre=$request->get('nombre');
85         $curso->descripcion=$request->get('descripcion');
86
87         $curso->update();
88         //dd($curso);
89         return Redirect::to('admin/curso');
90     }
91
92     public function destroy($id)
93     {
94         $curso=Curso::findOrFail($id);
95         $curso->delete();
96
97         return redirect()->route('curso.index');
98     }
99 }

```

Nota. Código PHP del controlador Cursos (Registrar Cursos).

Figura 44: Controlador de Horario (Gestionar Horario).

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 use SistemaWebMatricula;
8 use App\Models\Horario;
9 use Illuminate\Support\Facades\Redirect;
10 use Illuminate\Support\Facades\Input;
11 use App\Http\Requests\HorarioFormRequest;
12 use DB;
13
14 use Carbon\Carbon;
15 use Response;
16 use Illuminate\Support\Collection;
17
18 class HorarioController extends Controller
19 {
20     public function __construct()
21     {
22         $this->middleware('auth');
23     }
24
25     public function index(Request $request)
26     {
27         if($request)
28         {
29             $query=trim($request->get('searchText'));
30             $horario=DB::table('horario')->where('dia','LIKE','%'.$
31             query.'%')
32             ->orderBy('id_horario')
33             ->paginate(7);
34
35             return view('admin.horario.index',['horarioA'=>$horario,
36             "searchText"=>$query]);
37         }
38     }
39
40     public function create()
41     {
42         return view("admin.horario.create");
43     }
44
45     public function store(HorarioFormRequest $request)
46     {
47         $horario=new Horario;
48         $horario->hora_inicio=$request->get('hora_inicio');
49         $horario->hora_fin=$request->get('hora_fin');
50         $horario->dia=$request->get('dia');
51         $horario->save();
52
53         return Redirect::to('admin/horario');
54     }
55
56     public function show($id)
57     {
58         //
59     }
60
61     public function edit($id)
62     {
63         return view("admin.horario.edit",["horarioC"=>$Horario::
64         findOrFail($id)];
65     }
66
67     public function update(Request $request, $id)
68     {
69         $horario=Horario::findOrFail($id);
70         $horario->hora_inicio=$request->get('hora_inicio');
71         $horario->hora_fin=$request->get('hora_fin');
72         $horario->dia=$request->get('dia');
73
74         $horario->update();
75
76         return Redirect::to('admin/horario');
77     }
78
79     public function destroy($id)
80     {
81         $horario=Horario::findOrFail($id);
82         $horario->delete();
83
84         return redirect()->route('horario.index');
```

Nota. Código PHP del controlador Horario (Registrar Horario).

Figura 45: Controlador de Año Académico (Gestionar Año Académico).

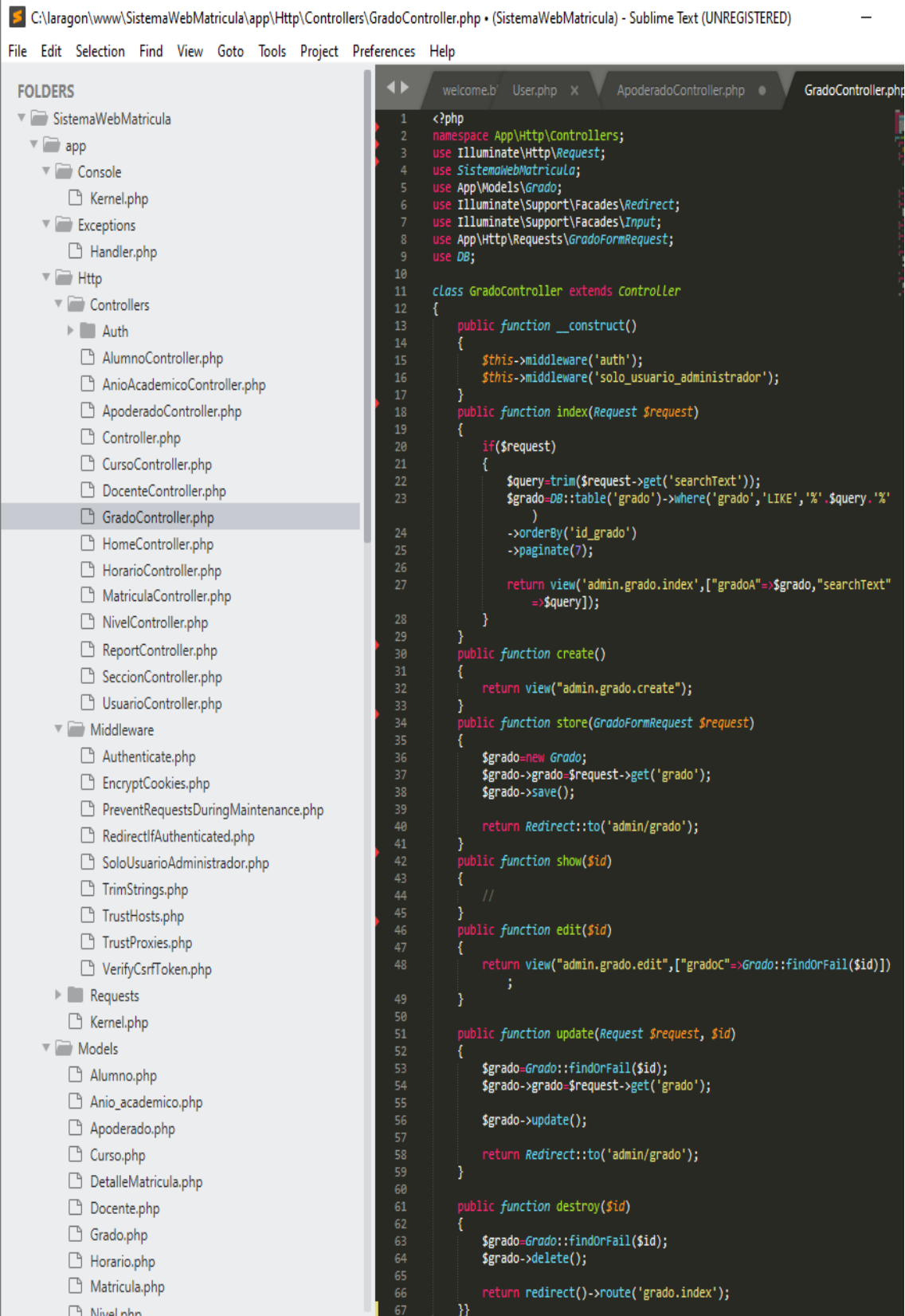
```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  use SistemaWebMatricula;
8  use App\Models\Anio_academico;
9  use Illuminate\Support\Facades\Redirect;
10 use Illuminate\Support\Facades\Input;
11 use App\Http\Requests\AnioAcademicoFormRequest;
12 use DB;
13
14 class AnioAcademicoController extends Controller
15 {
16     public function __construct()
17     {
18         $this->middleware('auth');
19         $this->middleware('solo_usuario_administrador');
20     }
21
22     public function index(Request $request)
23     {
24         if($request)
25         {
26             $query=trim($request->get('searchText'));
27             $anioacademico=DB::table('anio_academico')->where('
28                 descripcion','LIKE','%'.$query.'%')
29                 ->orderBy('id_anio_academico')
30                 ->paginate(7);
31
32             return view('admin.anioacademico.index',['anioA'=>$
33                 anioacademico,"searchText"=>$query]);
34         }
35     }
36
37     public function create()
38     {
39         return view("admin.anioacademico.create");
40     }
41
42     public function store(AnioAcademicoFormRequest $request)
43     {
44         $anio=new Anio_academico;
45         $anio->descripcion=$request->get('descripcion');
46         $anio->fecha_inicio=$request->get('fecha_inicio');
47         $anio->fecha_fin=$request->get('fecha_fin');
48         $anio->save();
49
50         return Redirect::to('admin/anioacademico');
51     }
52
53     public function show($id)
54     {
55         //
56     }
57
58     public function edit($id)
59     {
60         return view("admin.anioacademico.edit",["anioC"=>
61             Anio_academico::findOrFail($id)];
62     }
63
64     public function update(Request $request, $id)
65     {
66         $anio=Anio_academico::findOrFail($id);
67         $anio->descripcion=$request->get('descripcion');
68         $anio->fecha_inicio=$request->get('fecha_inicio');
69         $anio->fecha_fin=$request->get('fecha_fin');
70
71         $anio->update();
72
73         return Redirect::to('admin/anioacademico');
74     }
75
76     public function destroy($id)
77     {
78         $anio=Anio_academico::findOrFail($id);
79         $anio->delete();
80
81         return redirect()->route('anioacademico.index');
82     }
83 }

```

Nota. Código PHP del controlador Año Académico (Gestionar Año Académico).

Figura 46: Controlador de Secciones(Grado).



```
1 <?php
2 namespace App\Http\Controllers;
3 use Illuminate\Http\Request;
4 use SistemaWebMatricula;
5 use App\Models\Grado;
6 use Illuminate\Support\Facades\Redirect;
7 use Illuminate\Support\Facades\Input;
8 use App\Http\Requests\GradoFormRequest;
9 use DB;
10
11 class GradoController extends Controller
12 {
13     public function __construct()
14     {
15         $this->middleware('auth');
16         $this->middleware('solo_usuario_administrador');
17     }
18     public function index(Request $request)
19     {
20         if($request)
21         {
22             $query=trim($request->get('searchText'));
23             $grado=DB::table('grado')->where('grado','LIKE','%'.$query.'%')
24                 ->orderBy('id_grado')
25                 ->paginate(7);
26
27             return view('admin.grado.index',['gradoA'=>$grado,"searchText"
28                 =>$query]);
29         }
30     }
31     public function create()
32     {
33         return view("admin.grado.create");
34     }
35     public function store(GradoFormRequest $request)
36     {
37         $grado=new Grado;
38         $grado->grado=$request->get('grado');
39         $grado->save();
40
41         return Redirect::to('admin/grado');
42     }
43     public function show($id)
44     {
45         //
46     }
47     public function edit($id)
48     {
49         return view("admin.grado.edit",["gradoc"=>Grado::findOrFail($id)]);
50     }
51     }
52     public function update(Request $request, $id)
53     {
54         $grado=Grado::findOrFail($id);
55         $grado->grado=$request->get('grado');
56
57         $grado->update();
58
59         return Redirect::to('admin/grado');
60     }
61     }
62     public function destroy($id)
63     {
64         $grado=Grado::findOrFail($id);
65         $grado->delete();
66
67         return redirect()->route('grado.index');
```

Nota. Código PHP del controlador de Secciones (Grado).

Figura 47: Controlador de Secciones(Nivel).

```
1 <?php
2 namespace App\Http\Controllers;
3 use Illuminate\Http\Request;
4 use SistemaWebMatricula;
5 use App\Models\Nivel;
6 use Illuminate\Support\Facades\Redirect;
7 use Illuminate\Support\Facades\Input;
8 use App\Http\Requests\NivelFormRequest;
9 use DB;
10
11 class NivelController extends Controller
12 {
13     public function __construct()
14     {
15         $this->middleware('auth');
16         $this->middleware('solo_usuario_administrador');
17     }
18     public function index(Request $request)
19     {
20         if($request)
21         {
22             $query=trim($request->get('searchText'));
23             $nivel=DB::table('nivel')->where('nivel','LIKE','%'.$query.'%'
24             )
25             ->orderBy('id_nivel')
26             ->paginate(7);
27
28             return view('admin.nivel.index',['nivelA'=>$nivel,"searchText"
29             =>$query]);
30         }
31     }
32     public function create()
33     {
34         return view("admin.nivel.create");
35     }
36     public function store(NivelFormRequest $request)
37     {
38         $nivel=new Nivel;
39         $nivel->nivel-$request->get('nivel');
40         $nivel->save();
41
42         return Redirect::to('admin/nivel');
43     }
44     public function show($id)
45     {
46         //
47     }
48     public function edit($id)
49     {
50         return view("admin.nivel.edit",["nivelC"=>$nivel::findOrFail($id)]
51         );
52     }
53     public function update(Request $request, $id)
54     {
55         $nivel=Nivel::findOrFail($id);
56         $nivel->nivel-$request->get('nivel');
57
58         $nivel->update();
59
60         return Redirect::to('admin/nivel');
61     }
62     public function destroy($id)
63     {
64         $nivel=Nivel::findOrFail($id);
65         $nivel->delete();
66         return redirect()->route('nivel.index');
67     }
68 }
```

Nota. Código PHP del controlador de Secciones (Nivel).

Figura 48: Controlador de Secciones(Sección).

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 use SistemaWebMatricula;
8 use App\Models\Seccion;
9 use App\Models\Grado;
10 use App\Models\Nivel;
11 use Illuminate\Support\Facades\Redirect;
12 use App\Http\Requests\SeccionFormRequest;
13 use DB;
14
15 class SeccionController extends Controller
16 {
17     public function __construct()
18     {
19         $this->middleware('auth');
20         $this->middleware('solo_usuario_administrador');
21     }
22
23     public function index(Request $request)
24     {
25         if($request)
26         {
27             $query=trim($request->get('searchText'));
28             $seccion=DB::table('seccion as s')
29             ->join('grado as g','s.id_grado','=','g.id_grado')
30             ->join('nivel as n','s.id_nivel','=','n.id_nivel')
31             ->select('s.id_seccion','s.nombre_seccion','s.cantidad','
32             $grado as grado','n.nivel as nivel')
33             ->where('s.nombre_seccion','LIKE','%'.$query.'%')
34             ->orderBy('s.id_seccion','asc')
35             ->paginate();
36             //dd($curso);
37
38             return view('admin.seccion.index',['seccionA'=>$seccion,"
39             searchText"=>$query]);
40         }
41
42     }
43
44     public function create()
45     {
46         $grado=DB::table('grado')->get();
47         $nivel=DB::table('nivel')->get();
48         //dd($apoderado);
49         $apoderado=Apoderado::orderBy('id','ASC')->pluck('nombre','id');
50         //dd($apoderado);
51         return view('admin.seccion.create',['gradoA'=>$grado,"nivelA"=>$
52         nivel]);
53     }
54
55     public function store(SeccionFormRequest $request)
56     {
57         $seccion=new Seccion;
58         $seccion->nombre_seccion=$request->get('nombre_seccion');
59         $seccion->cantidad=$request->get('cantidad');
60         $seccion->id_grado=$request->get('id_grado');
61         $seccion->id_nivel=$request->get('id_nivel');
62
63         $seccion->save();
64
65         //dd($curso);
66         return redirect()->route('seccion.index');
```

Nota. Código PHP del controlador de Secciones (Sección).

Figura 49: Controlador de Secciones(Sección).

```
65 public function show($id)
66 {
67     //
68 }
69
70 public function edit($id)
71 {
72     $seccion=Seccion::findOrFail($id);
73     $grado=DB::table('grado')->get();
74     $nivel=DB::table('nivel')->get();
75     //dd($alumno);
76     return view("admin.seccion.edit",["seccionC"=>$seccion,"gradoA"=>$
77     $grado,"nivelA"=>$nivel]);
78 }
79
80 public function update(Request $request, $id)
81 {
82     $seccion=Seccion::findOrFail($id);
83     $seccion->nombre_seccion=$request->get('nombre_seccion');
84     $seccion->cantidad=$request->get('cantidad');
85     $seccion->id_grado=$request->get('id_grado');
86     $seccion->id_nivel=$request->get('id_nivel');
87
88     $seccion->update();
89     //dd($curso);
90     return Redirect::to('admin/seccion');
91 }
92
93 public function destroy($id)
94 {
95     $seccion=Seccion::findOrFail($id);
96     $seccion->delete();
97
98     return redirect()->route('seccion.index');
```

Nota. Código PHP del controlador de Secciones (Sección).

Figura 50: Controlador ReportController (Generar reportes)

```

1 <?php
2 namespace App\Http\Controllers;
3
4 use Illuminate\Http\Request;
5 use App\Models\Matricula;
6 use Carbon\Carbon;
7
8 use App\Models\Nivel;
9 use App\Models\Seccion;
10 use App\Models\ALumno;
11 use App\Models\Grado;
12 use App\Models\DetalleMatricula;
13 use Illuminate\Support\Facades\DB;
14 use Barryvdh\DomPDF\Facade\Pdf;
15 use Dompdf\Dompdf;
16
17
18 class ReportController extends Controller
19 {
20
21     public function __construct()
22     {
23         //
24     }
25
26     public function reports_day()
27     {
28         $matricula = Matricula::whereDate('fecha_hora_matricula', Carbon::today('America/
29             Lima'))->get();
30         $total = $matricula->sum('total_matricula');
31
32         //dd($matricula);
33         //return view('admin.report.reports_day', compact('matricula', 'total'));
34         return view('admin.report.reports_day', ["matriculaA"=>$matricula, "totalA"=>$total
35             ]);
36     }
37
38     public function reports_date()
39     {
40         $matricula = Matricula::whereDate('fecha_hora_matricula', Carbon::today('America/
41             Lima'))->get();
42         $total = $matricula->sum('total_matricula');
43
44         //dd($matricula);
45         return view('admin.report.reports_date', ["matriculaA"=>$matricula, "totalA"=>$
46             total]);
47     }
48
49     public function reports_results(Request $request)
50     {
51         $fi = $request->fecha_ini. ' 00:00:00';
52         $ff = $request->fecha_fin. ' 23:59:59';
53         $matricula = Matricula::whereBetween('fecha_hora_matricula', [$fi, $ff])->get();
54         $total = $matricula->sum('total_matricula');
55
56         //dd($matricula);
57         return view('admin.report.reports_date', ["matriculaA"=>$matricula, "totalA"=>$
58             total]);
59         //return Redirect::to('/sales/reports_date');
60     }
61
62     public function alumnosmatricula()
63     {
64         $matriculas = Matricula::all();
65         $alumno = Alumno::all();
66         $total = $matriculas->sum('total_matricula');
67         //dd($matriculas);
68     }
69 }
70
71 public function formularioBusqueda()
72 {
73     $alumno = DB::table('alumno')
74     ->get();
75     return view('admin.report.busqueda', ['alumnoA' => $alumno]);
76 }
77
78 public function buscar(Request $request)
79 {
80     $nombre = $request->input('nombre');
81     $alumno = Alumno::where('nombre', 'LIKE', "%$nombre%")->first();
82
83     if ($alumno) {
84         //with('seccion')->get()->pluck('id_seccion');
85         $secciones = $alumno->matriculas()->with('detallesMatricula', 'seccion')->get()
86         ->pluck('detallesMatricula')->collapse();
87         ->pluck('seccion.nombre_seccion')->unique();
88         //dd($nombre, $alumno, $secciones);
89
90         // Crear el contenido del PDF
91         $html = "<table style='text-align: center;'>Reporte de Matriculas de $nombre/<h1>";
92         $html .= "<table style='width: 100%; border-collapse: collapse; margin-top:
93             20px;'>";
94         $html .= "<tr style='background-color: #ccc; text-align: center;'>";
95         $html .= "<th style='border: 1px solid #000; padding: 10px;'>ID</th>";
96         $html .= "<th style='border: 1px solid #000; padding: 10px;'>Alumno</th>";
97         $html .= "<th style='border: 1px solid #000; padding: 10px;'>Apellidos</th>";
98         $html .= "<th style='border: 1px solid #000; padding: 10px;'>seccion</th>";
99         $html .= "</tr>";
100         foreach ($secciones as $matricula) {
101             $html .= "<tr style='text-align: center;'>";
102             $html .= "<td style='border: 1px solid #000; padding: 10px;'>" . $
103                 alumno->id . "</td>";
104             $html .= "<td style='border: 1px solid #000; padding: 10px;'>" . $
105                 alumno->nombre . "</td>";
106             $html .= "<td style='border: 1px solid #000; padding: 10px;'>" . $
107                 alumno->apellido . "</td>";
108             $html .= "<td style='border: 1px solid #000; padding: 10px;'>" . $
109                 matricula . "</td>";
110         }
111         $html .= "</table>";
112         // Crear una instancia de Dompdf
113         $dompdf = new Dompdf();
114         // Cargar el contenido HTML en Dompdf
115         $dompdf->loadHtml($html);
116         // Renderizar el PDF
117         $dompdf->render();
118         // Obtener el contenido del PDF como una cadena de texto
119         $pdf = $dompdf->output();
120         // Descargar el PDF
121         //return view('admin.report.grados', ['alumnoA' => $alumno]);
122         // Devolver la vista con el formulario y el contenido del pdf
123         $request->session()->put('pdfb', $pdf);
124         return view('admin.report.secciones', compact('alumno', 'secciones', 'pdfb'));
125     } else {
126         return view('admin.report.no_encontrado', compact('nombre'));
127     }
128 }
129
130 }
131

```

Nota. Código PHP del controlador de Reportes (Generar reportes).

Figura 51: Controlador ReportController (Generar reportes)

```

68     return view('admin.report.resultado', ["matriculaA"=>$matriculas, "totalA"=>$total,
69         "alumnoA"=>$alumno]);
70 }
71
72 public function formularioBusqueda()
73 {
74     $alumno = DB::table('alumno')
75     ->get();
76     return view('admin.report.busqueda', ['alumnoA' => $alumno]);
77 }
78
79 public function buscar(Request $request)
80 {
81     $nombre = $request->input('nombre');
82     $alumno = Alumno::where('nombre', 'LIKE', "%$nombre%")->first();
83
84     if ($alumno) {
85         //with('seccion')->get()->pluck('id_seccion');
86         $secciones = $alumno->matriculas()->with('detallesMatricula', 'seccion')->get()
87         ->pluck('detallesMatricula')->collapse();
88         ->pluck('seccion.nombre_seccion')->unique();
89         //dd($nombre, $alumno, $secciones);
90
91         // Crear el contenido del PDF
92         $html = "<table style='text-align: center;'>Reporte de Matriculas de $nombre/<h1>";
93         $html .= "<table style='width: 100%; border-collapse: collapse; margin-top:
94             20px;'>";
95         $html .= "<tr style='background-color: #ccc; text-align: center;'>";
96         $html .= "<th style='border: 1px solid #000; padding: 10px;'>ID</th>";
97         $html .= "<th style='border: 1px solid #000; padding: 10px;'>Alumno</th>";
98         $html .= "<th style='border: 1px solid #000; padding: 10px;'>Apellidos</th>";
99         $html .= "<th style='border: 1px solid #000; padding: 10px;'>seccion</th>";
100         $html .= "</tr>";
101         foreach ($secciones as $matricula) {
102             $html .= "<tr style='text-align: center;'>";
103             $html .= "<td style='border: 1px solid #000; padding: 10px;'>" . $
104                 alumno->id . "</td>";
105             $html .= "<td style='border: 1px solid #000; padding: 10px;'>" . $
106                 alumno->nombre . "</td>";
107             $html .= "<td style='border: 1px solid #000; padding: 10px;'>" . $
108                 alumno->apellido . "</td>";
109             $html .= "<td style='border: 1px solid #000; padding: 10px;'>" . $
110                 matricula . "</td>";
111         }
112         $html .= "</table>";
113         // Crear una instancia de Dompdf
114         $dompdf = new Dompdf();
115         // Cargar el contenido HTML en Dompdf
116         $dompdf->loadHtml($html);
117         // Renderizar el PDF
118         $dompdf->render();
119         // Obtener el contenido del PDF como una cadena de texto
120         $pdf = $dompdf->output();
121         // Descargar el PDF
122         //return view('admin.report.grados', ['alumnoA' => $alumno]);
123         // Devolver la vista con el formulario y el contenido del pdf
124         $request->session()->put('pdfb', $pdf);
125         return view('admin.report.secciones', compact('alumno', 'secciones', 'pdfb'));
126     } else {
127         return view('admin.report.no_encontrado', compact('nombre'));
128     }
129 }
130 }
131

```

Nota. Código PHP del controlador de Reportes (Generar reportes).

Figura 52: Controlador ReportController (Generar reportes)

```

133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Nota. Código PHP del controlador de Reportes (Generar reportes).

Figura 55: Vistas Registrar Matrícula (Create.blade.php).

```

131 <option value="{{ $hora->id_horario}}">{{ $hora->dia}} {{
132 $hora->hora_inicio}} - {{ $hora->hora_fin}}</option>
133 </select>
134 </div>
135 <div class="col-5">
136 <label for="estado">Estado:</label>
137 <select name="estado" class="form-control" id="pidestado">
138 <option value="Nuevo">Nuevo</option>
139 <option value="Antiguo">Antiguo</option>
140 </select>
141 </div>
142 </div>
143 <div class="row">
144 <div class="col-3">
145 <label>Sección:</label>
146 <select name="id_seccion" class="form-control selectpicker" id="
147 pidseccion" data-live-search="true">
148 <option value="">Seleccione una opción</option>
149 @foreach($seccionA as $scurs)
150 <option value="{{ $scurs->id_seccion}}_{{ $scurs->grado}}_{{
151 $scurs->nivel}}">{{ $scurs->grado}} - {{ $scurs->nombre_seccion}}
152 </option>
153 @endforeach
154 </select>
155 </div>
156 <div class="col-4">
157 <label>Grado:</label>
158 <input type="text" disabled name="pidgrado" class="form-control" id="
159 pidgrado" placeholder="Grado">
160 </div>
161 <div class="col-5">
162 <label>Nivel:</label>
163 <input type="text" disabled name="pidnivel" class="form-control" id="
164 pidnivel" placeholder="Nivel">
165 </div>
166 <div class="col-md-6">
167 <div class="form-group">
168 <label for="costo_matricula">Costo:</label>
169 <input type="number" name="costo_matricula" class="form-control
170 id="pidcosto" placeholder="Ingrese el costo a pagar $/.">
171 </div>
172 </div>
173 </div>
174 </div>
175 <!-- BOTÓN AGREGAR -->
176 <div class="col-lg-2 col-sm-2 col-md-2 col-xs-12">
177 <div class="form-group">
178 <button type="button" id="bt_add" class="btn btn-primary">Agregar</
179 button>
180 </div>
181 </div>
182 <!-- ÁCA ES PARA EL MAESTRO DETALLE -->
183 <table id="detalles" class="table table-striped shadow-lg mt-4" style="
184 width:100%; font-size:14px;">
185 <thead class="bg-primary text-white">
186 <tr>
187 <th>Opciones</th>
188 <th>Curso</th>
189 <th>Año Académico</th>
190 <th>Seccion</th>
191 <th>Horario</th>
192 <th>Estado</th>
193 <th>Costo</th>

```

Nota. Código PHP de vista Matrícula (Create.blade.php).

Figura 56: Vistas Registrar Matrícula (Create.blade.php).

```
192         <th>SubTotal</th>
193     </tr>
194 </thead>
195 <tfoot>
196     <th>Total</th>
197 </tr>
198 </tfoot>
199 </table>
200 </div>
201 <div class="col-md-6 col-sm-6 col-md-6 col-xs-12" id="guardar">
202     <div class="form-group">
203         <input name="_token" value="{{ csrf_token() }}" type="hidden">
204         <button class="btn btn-primary" type="submit"> Guardar </button>
205         <button class="btn btn-danger" type="reset"> Cancelar </button>
206     </div>
207 </div>
208 <!-- FIN DEL MAESTRO DETALLE-->
209 </div>
210 <!-- /.card-body ---->
211 <!-- /card ---->
212 </div>
213 <!-- /.content -->
214 <!-- Form::close() !!>
215 @stop
216 @section('css')
217     <link rel="stylesheet" href="/css/admin_custom.css">
218 @stop
219 @section('js')
220     <script> console.log('Hi!'); </script>
221     <script>
222     $(document).ready(function() {
223         $('#btnOpenSalt').click(function() {
224             Swal.fire(
225                 'Buen trabajo!',
226                 'La matrícula ha sido registrada con éxito!',
227                 'success'
228             );
229         });
230     });
231     var Toast = Swal.mixin({
232         toast: true,
233         position: 'top-end',
234         showConfirmButton: false,
235         timer: 3000
236     });
237     $('#btnOpenSalt').click(function() {
238         Toast.fire({
239             icon: 'success',
240             title: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr.'
241         });
242     });
243 </script>
244 <!-- TODO ESTO ES PARA EL MAESTRO DETALLE-->
245 <script>
246     $(document).ready(function(){
```

Nota. Código PHP de vista Matrícula (Create.blade.php).

Figura 57: Vistas Registrar Matrícula (Create.blade.php).

```
262 $('#bt_add').click(function(){
263     agregar();
264 });
265 });
266
267 var cont=0;
268 total=0;
269 subtotal=[];
270 $('#guardar').hide();
271 $('#pidcurso').change(mostrarValores);
272 $('#pidseccion').change(mostrarValoresB);
273
274 function mostrarValores()
275 {
276     datosCurso=document.getElementById('pidcurso').value.split('_');
277     $('#pidcantidad').val(datosCurso[1]);
278 }
279
280 function mostrarValoresB()
281 {
282     datos=document.getElementById('pidseccion').value.split('_');
283     $('#pidnivel').val(datos[2]);
284     $('#pidgrado').val(datos[1]);
285 }
286
287 function agregar()
288 {
289     datosCurso=document.getElementById('pidcurso').value.split('_');
290
291     idcurso=datosCurso[0];
292     curso=$('#pidcurso option:selected').text();
293     anioacademico=$('#pidanioacademico').val();
294     anioacademicoB=$('#pidanioacademico option:selected').text();
295     datos=document.getElementById('pidseccion').value.split('_');
296     idseccion=datos[0];
297     seccion=$('#pidseccion option:selected').text();
298     horario=$('#pidhorario').val();
299     horarioB=$('#pidhorario option:selected').text();
300     estado=$('#pidestado').val();
301     costo=$('#pidcosto').val();
302     //cantidad=$('#pidcantidad').val();
303
304     if(idcurso!="" && curso!="" && costo!="" && costo>0 && anioacademico!="" && idseccion
305        !="" && seccion!="" && horario !="" && estado !="" )
306     {
307         subtotal[cont]=(1*costo);
308         total_total+=subtotal[cont];
309
310         var fila='<tr class="selected" id="fila'+cont+'><td><button type="button"
311             class="btn btn-warning" onclick="eliminar('+cont+'
312             );">X</button></td><td><input type="hidden" name="idcurso[]" value="'+idcurso+'
313             "></td><td><input type="hidden" name="anioacademico[]" value="'+
314             anioacademico+'"></td><td><input type="hidden"
315             name="idseccion[]" value="'+idseccion+'"></td><td><input
316             type="hidden" name="horario[]" value="'+horario+'"></td><td><input
317             name="estado[]" value="'+estado+'"></td><td><input name="costo[]" value="'+
318             costo+'</td><td><td>'+subtotal[cont]+'</td></tr>';
319         cont++;
320         limpiar();
321         $('#total').html("S/. "+total);
322         $('#total_matricula').val(total);
323         evaluar();
324         $('#detalles').append(fila);
325     }
326     else
327     {
328         alert("Error al ingresar el detalle de la matricula, revise los datos del curso");
329     }
330 }
```

Nota. Código PHP de vista Matrícula (Create.blade.php).

Figura 58: Vistas Registrar Matrícula (Create.blade.php).

```
325 function limpiar(){
326     $('#pidcurso').val("");
327     $('#pidanioacademico').val("");
328     $('#pidseccion').val("");
329     $('#pidhorario').val("");
330     $('#pidestado').val("");
331     $('#pidcosto').val("");
332     $('#pidcantidad').val("");
333     $('#pidgrado').val("");
334     $('#pidnivel').val("");
335 }
336
337 function evaluar()
338 {
339     if (total>0)
340     {
341         $('#guardar').show();
342     }
343     else
344     {
345         $('#guardar').hide();
346     }
347 }
348
349 function eliminar(index){
350     total_total-=subtotal[index];
351     $('#total').html("S/. " + total);
352     $('#total_matricula').val(total);
353     $('#fila'+ index).remove();
354     evaluar();
355 }
356
357 </script>
358 @stop
```

Nota. Código PHP de vista Matrícula (Create.blade.php).

Figura 59: Vistas Registrar Matrícula (edit.blade.php).

```
1 @extends('adminlte::page')
2
3 @section('title', 'Seccion')
4
5 @section('content_header')
6 <h1>Editar Seccion: {{ $seccion->nombre_seccion }}</h1>
7 @if (count($errors)>0)
8 <div class="alert alert-danger">
9 <ul>
10 @foreach($errors->all() as $errors)
11 <li>{{ $errors }}</li>
12 @endforeach
13 </ul>
14 </div>
15 @endif
16 @stop
17
18 @section('content')
19 <!-- Main Content -->
20 <section class="content">
21 <div class="container-fluid">
22 <div class="row">
23 <!-- left column -->
24 <div class="col-md-12">
25 <!-- jquery validation -->
26 <div class="card card-primary">
27 <div class="card-header">
28 <h3 class="card-title">Quick Example <small>jQuery Validation</small></h3>
29 </div>
30 <!-- /.card-header -->
31 <!-- form start -->
32 {!! Form::open(array('route' =>
33 ['seccion.update',$seccion->id_seccion], 'method' => 'PUT', 'files'=>true))
34 !!}
35 {!! Form::token() !!}
36
37 <form id="quickForm">
38 <div class="card-body">
39 <div class="form-group">
40 <label for="nombre_seccion">Nombre de la Seccion</label>
41 <input type="text" name="nombre_seccion" class="form-control" required
42 value="{{ $seccion->nombre_seccion }}">
43 </div>
44 <div class="form-group">
45 <label for="cantidad">Cantidad</label>
46 <input type="text" name="cantidad" class="form-control" required value
47="{{ $seccion->cantidad }}">
48 </div>
49 <div class="form-group">
50 <label for="id_grado">Grado</label>
51 <select name="id_grado" class="form-control">
52 @foreach($gradoA as $gr)
53 <option value="{{ $gr->id_grado }}" selected="{{ $gr->grado }}">
54 <option value="{{ $gr->id_grado }}">{{ $gr->grado}}</option>
55 @endforeach
56 </select>
57 </div>
58 <div class="form-group">
59 <label for="id_nivel">Nivel</label>
60 <select name="id_nivel" class="form-control">
61 @foreach($nivelA as $nv)
62 <option value="{{ $nv->id_nivel }}" selected="{{ $nv->nivel }}">
63 <option value="{{ $nv->id_nivel }}">{{ $nv->nivel}}</option>
64 @endforeach
65 </select>
66 </div>
67 </div>
68 </div>
69 <div class="card-footer">
70 <button class="btn btn-primary" type="submit" id="btnOpenSalTB">Guardar
71 </button>
72 <button class="btn btn-danger" type="reset">Cancelar</button>
73 </div>
74 </div>
75 </form>
76 </div>
77 <!-- /.card -->
78 </div>
79 <!-- /.col (left) -->
80 <!-- right column -->
81 <div class="col-md-6">
82 </div>
83 </div>
84 <!-- /.col (right) -->
85 </div>
86 <!-- /.row -->
87 </div><!-- /.container-fluid -->
88 </section>
89 <!-- /.content -->
90 {!! Form::close() !!}
91 @stop
92
93 @section('css')
94 <link rel="stylesheet" href="/css/admin_custom.css">
95 @stop
96
97 @section('js')
98 <script> console.log('Hi!'); </script>
99 @stop
```

Nota. Código PHP de vista Matrícula (edit.blade.php).

Figura 60: Vistas Registrar Matrícula (edit.blade.php).

```
66 @foreach
67 </select>
68 </div>
69 </div>
70 <!-- /.card-body -->
71 <div class="card-footer">
72 <button class="btn btn-primary" type="submit" id="btnOpenSalTB">Guardar
73 </button>
74 <button class="btn btn-danger" type="reset">Cancelar</button>
75 </div>
76 </form>
77 </div>
78 <!-- /.card -->
79 </div>
80 <!-- /.col (left) -->
81 <!-- right column -->
82 <div class="col-md-6">
83 </div>
84 </div>
85 <!-- /.col (right) -->
86 </div>
87 <!-- /.row -->
88 </div><!-- /.container-fluid -->
89 </section>
90 <!-- /.content -->
91 {!! Form::close() !!}
92 @stop
93
94 @section('css')
95 <link rel="stylesheet" href="/css/admin_custom.css">
96 @stop
97
98 @section('js')
99 <script> console.log('Hi!'); </script>
100 @stop
```

Nota. Código PHP de vista Matrícula (edit.blade.php).

Figura 61: Vistas Registrar Matrícula (index.blade.php).

```
C:\laragon\www\SistemaWebMatricula\resources\views\admin\matricula\index.blade.php (SistemaWebMatricula) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
w ReportController.php create.blade.php index.blade.php User.php
1 @extends('adminlte::page')
2
3 @section('title', 'Matricula')
4
5 @section('content_header')
6 <h1>Listado de Matriculas <!-- {{ route('apoderado/create')}} --><button class="btn
7 btn-success">Nueva Matricula</button></a--></h1>
8
9 @stop
10
11 @section('content')
12 <div class="card">
13 <div class="card-header">
14 <a href="matricula/create" class="btn btn-primary">Nueva Matricula</a>
15 <div style="text-align:right">
16 <a href="{{ route('matriculas.pdf') }}" class="btn btn-success">Generar
17 PDF</a>
18 </div>
19 </div>
20 <div class="card-body">
21 <table id="example" class="table table-striped shadow-lg mt-4" style="width:
22 100%">
23 <thead class="bg-primary text-white">
24 <tr>
25 <th>ID</th>
26 <th>Fecha - Hora</th>
27 <th>Alumno Matriculado</th>
28 <th>Tipo de Comprobante</th>
29 <th>Total de Matricula</th>
30 <th>Opciones</th>
31 </tr>
32 </thead>
33 <tbody>
34 @foreach($matriculaA as $matricula)
35 <tr>
36 <td>{{ $matricula->id_matricula }}</td>
37 <td>{{ $matricula->fecha_hora_matricula }}</td>
38 <td>{{ $matricula->alumno }} {{ $matricula->apellido }}</td>
39 <td>{{ $matricula->tipo_comprobante }}; {{ $matricula->serie_comprob
40 ante }}. {{ $matricula->numero_comprobante }}</td>
41 <td>$. {{ $matricula->total_matricula }}</td>
42 <td width="250px">
43 <a href="{{ route('matricula.pdf', $matricula->id_matricula) }}"
44 ><button class="btn btn-secondary btn-sm">Imprimir</button></
45 a>
46 <a href="{{ route('matricula.show', $matricula->id_matricula)
47 }}"><button class="btn btn-primary btn-sm">Detalles</button></
48 a>
49 <!-- Button trigger modal -->
50 <button type="button" class="btn btn-danger btn-sm"
51 data-bs-toggle="modal" data-bs-target="#modal-delete-{{
52 $matricula->id_matricula }}">
53 Anular
54 </button>
55 </td>
56 </tr>
57 </tbody>
58 </table>
59 @include('admin.matricula.modal')
60 @endforeach
61 </tbody>
62 </div>
63 </div>
64 @stop
65
66 @section('css')
67 <link rel="stylesheet" href="/css/admin_custom.css">
68 <link rel="stylesheet" href="https://cdn.datatables.net/1.13.1/css/
69 datatables.bootstrap5.min.css">
70
71 @stop
72
73 @section('js')
74 <script> console.log('Hi!'); </script>
75 <script src="https://code.jquery.com/jquery-3.5.1.js"></script>
76 <script src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>
77 <script src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap5.min.js"></
78 script>
79
80 <script>
81 $(document).ready(function () {
82 $('#example').DataTable({
83 "language":{
84 "search": "Buscar",
85 "lengthMenu": "Mostrar _MENU_ registros por página",
86 "info": "Mostrando página _PAGE_ de _PAGES_",
87 "paginate":{
88 "previous": "Anterior",
89 "next": "siguiente",
90 "first": "Primero",
91 "last": "Último"
92 }
93 }
94 });
95 </script>
96
97 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js
98 integrity="sha384-0BqDVMz9ATKXIEp9t1CKS/Z9fNFEXIDAYUJMAeBASjFucZsmkbsSunQ1mh/jp3"
99 crossorigin="anonymous"></script>
100 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
101 integrity="sha384-IDwe1+LCz02R09k972gdyv1+AESN10+x7T8kgC9I5HFtUNz0WmPc12o6p9vxnk"
102 crossorigin="anonymous"></script>
103
104 @stop
```

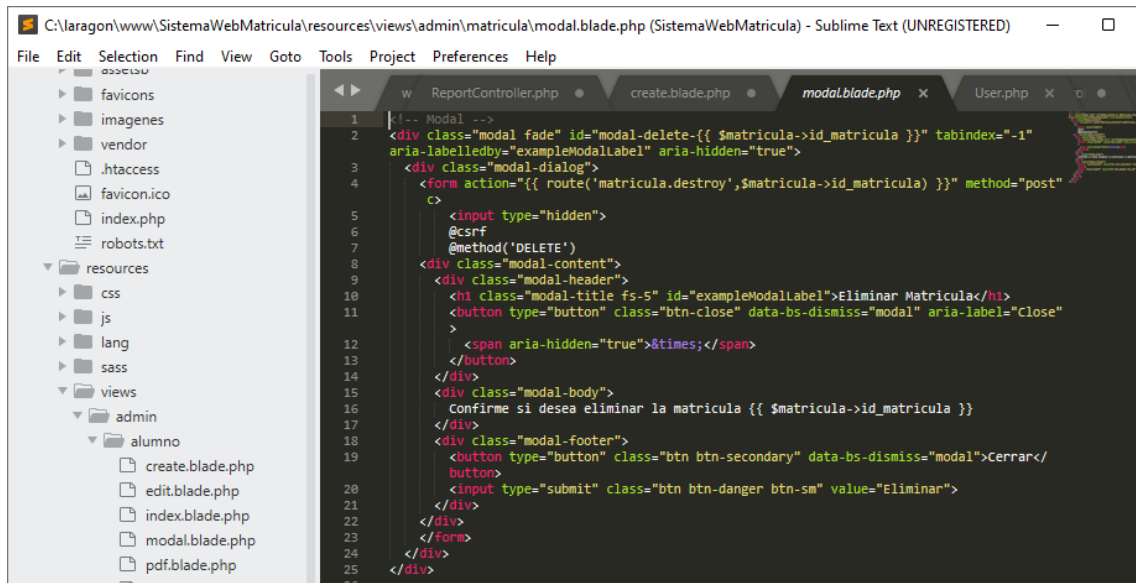
Nota. Código PHP de vista Matrícula (index.blade.php).

Figura 62: Vistas Registrar Matrícula (index.blade.php).

```
C:\laragon\www\SistemaWebMatricula\resources\views\admin\matricula\index.blade.php (SistemaWebMatricula) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
w ReportController.php create.blade.php index.blade.php User.php
60 @section('js')
61 <script> console.log('Hi!'); </script>
62 <script src="https://code.jquery.com/jquery-3.5.1.js"></script>
63 <script src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>
64 <script src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap5.min.js"></
65 script>
66
67 <script>
68 $(document).ready(function () {
69 $('#example').DataTable({
70 "language":{
71 "search": "Buscar",
72 "lengthMenu": "Mostrar _MENU_ registros por página",
73 "info": "Mostrando página _PAGE_ de _PAGES_",
74 "paginate":{
75 "previous": "Anterior",
76 "next": "siguiente",
77 "first": "Primero",
78 "last": "Último"
79 }
80 }
81 });
82 </script>
83
84 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js
85 integrity="sha384-0BqDVMz9ATKXIEp9t1CKS/Z9fNFEXIDAYUJMAeBASjFucZsmkbsSunQ1mh/jp3"
86 crossorigin="anonymous"></script>
87 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
88 integrity="sha384-IDwe1+LCz02R09k972gdyv1+AESN10+x7T8kgC9I5HFtUNz0WmPc12o6p9vxnk"
89 crossorigin="anonymous"></script>
90
91 @stop
```

Nota. Código PHP de vista Matrícula (index.blade.php).

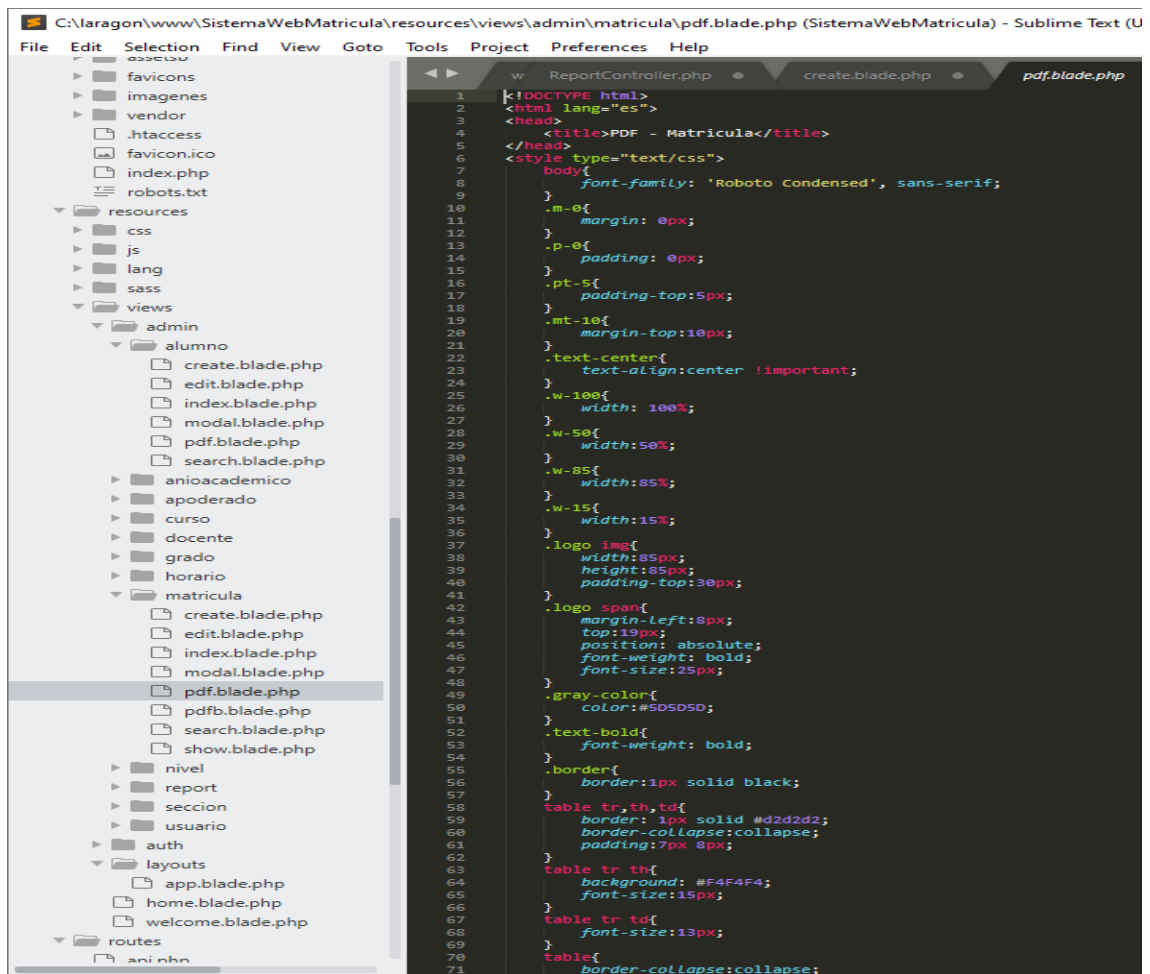
Figura 63: Vistas Registrar Matrícula (modal.blade.php).



```
1 <!-- Modal -->
2 <div class="modal fade" id="modal-delete-{{ $matricula->id_matricula }}" tabindex="-1"
3   aria-labelledby="exampleModalLabel" aria-hidden="true">
4   <div class="modal-dialog">
5     <form action="{{ route('matricula.destroy',$matricula->id_matricula) }}" method="post">
6       <input type="hidden">
7       @csrf
8       @method('DELETE')
9     <div class="modal-content">
10      <div class="modal-header">
11        <h1 class="modal-title fs-5" id="exampleModalLabel">Eliminar Matrícula</h1>
12        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close">
13          <span aria-hidden="true">&times;</span>
14        </button>
15      </div>
16      <div class="modal-body">
17        Confirme si desea eliminar la matricula {{ $matricula->id_matricula }}
18      </div>
19      <div class="modal-footer">
20        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</
21        button>
22        <input type="submit" class="btn btn-danger btn-sm" value="Eliminar">
23      </div>
24    </form>
25  </div>
26 </div>
```

Nota. Código PHP de vista Matrícula (modal.blade.php).

Figura 64: Vistas Registrar Matrícula (pdf.blade.php).



```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <title>PDF - Matrícula</title>
5 </head>
6 <style type="text/css">
7   body{
8     font-family: 'Roboto Condensed', sans-serif;
9   }
10   .m-0{
11     margin: 0px;
12   }
13   .p-0{
14     padding: 0px;
15   }
16   .pt-5{
17     padding-top:5px;
18   }
19   .mt-10{
20     margin-top:10px;
21   }
22   .text-center{
23     text-align:center !important;
24   }
25   .w-100{
26     width: 100%;
27   }
28   .w-50{
29     width:50%;
30   }
31   .w-85{
32     width:85%;
33   }
34   .w-15{
35     width:15%;
36   }
37   .logo_img{
38     width:85px;
39     height:85px;
40     padding-top:30px;
41   }
42   .logo_span{
43     margin-left:8px;
44     top:19px;
45     position: absolute;
46     font-weight: bold;
47     font-size:25px;
48   }
49   .gray-color{
50     color:#505050;
51   }
52   .text-bold{
53     font-weight: bold;
54   }
55   .border{
56     border:1px solid black;
57   }
58   table tr,th,td{
59     border: 1px solid #d2d2d2;
60     border-collapse:collapse;
61     padding:7px 8px;
62   }
63   table tr th{
64     background: #F4F4F4;
65     font-size:15px;
66   }
67   table tr td{
68     font-size:13px;
69   }
70   table{
71     border-collapse:collapse;
```

Nota. Código PHP de vista Matrícula (pdf.blade.php).

Figura 65: Vistas Registrar Matrícula (pdf.blade.php).

```

72  .box-text p{
73      Line-height:10px;
74  }
75  .float-left{
76      float:left;
77  }
78  .total-part{
79      font-size:16px;
80      Line-height:12px;
81  }
82  .total-right p{
83      padding-right:20px;
84  }
85  }
86  </style>
87  <body>
88  <div class="head-title">
89  <h1 class="text-center m-0 p-0">Reporte de Matricula de Alumnos</h1>
90  </div>
91  <div class="add-detail mt-10">
92  <div class="w-50 float-left mt-10">
93  <p class="m-0 pt-5 text-bold w-100">ID de Matricula : <span class="gray-color">#
94  <span class="m-0 pt-5 text-bold w-100">{{ $matriculaA->id_matricula}}</span></p>
95  <p class="m-0 pt-5 text-bold w-100">{{ $matriculaA->tipo_comprobante }} de Venta
96  <p class="m-0 pt-5 text-bold w-100">{{ $matriculaA->numero_comprobante }} </span></p>
97  <p class="m-0 pt-5 text-bold w-100">Fecha - Hora : <span class="gray-color">{{
98  $matriculaA->fecha_hora_matricula }}</span></p>
99  </div>
100 <div class="w-50 float-left logo mt-10">
101 </span> </div>
102 </div>
103 <div style="clear: both;"></div>
104 <div class="table-section bill-tbl w-100 mt-10">
105 <table class="table w-100 mt-10">
106 <thead>
107 <tr>
108 <th class="w-50">Alumno Matriculado</th>
109 <th class="w-50">Apoderado</th>
110 </tr>
111 </thead>
112 <tbody>
113 <tr>
114 <td>{{ $matriculaA->nombre }} {{ $matriculaA->apellido }}</td>
115 <td>{{ $matriculaA->apoderado }} {{ $matriculaA->apoderadoB }}</td>
116 </tr>
117 </tbody>
118 </table>
119 <div class="table-section bill-tbl w-100 mt-10">
120 <table class="table w-100 mt-10">
121 <thead>
122 <tr>
123 <th class="w-50">Información del Alumno</th>
124 <th class="w-50"></th>
125 </tr>
126 </thead>
127 <tbody>
128 <tr>
129 <td>
130 <div class="box-text">
131 <p>Email:</p>
132 <p>DNI:</p>
133 <p>Fecha de Nacimiento:</p>
134 <p>Sexo:</p>
135 <p>Dirección:</p>
136 <p>Celular:</p>
137 </div>
138 </td>
139 <td>
140 <div class="box-text">
141 <p>{{ $matriculaA->email }}</p>
142 <p>{{ $matriculaA->DNI }}</p>
143 <p>{{ $matriculaA->fecha_nac }}</p>
144 <p>{{ $matriculaA->sexo }}</p>
145 </div>
146 </td>
147 </tr>
148 </tbody>
149 </table>
150 </div>
151 <p class="m-0 pt-5 text-bold w-100">Curso al que está matriculado :<span class="
152 gray-color">{{ $matriculaA->curso }}</span></p>
153 <div class="table-section bill-tbl w-100 mt-10">
154 <table class="table w-100 mt-10">
155 <thead>
156 <tr>
157 <th class="w-50">Curso</th>
158 <th class="w-50">Año Académico</th>
159 <th class="w-50">Sección</th>
160 <th class="w-50">Horario</th>
161 <th class="w-50">Estado</th>
162 <th class="w-50">Costo</th>
163 <th class="w-50">Sub Total</th>
164 </tr>
165 </thead>
166 <tbody>
167 <tr>
168 <td colspan="7">
169 <div class="total-part">
170 <div class="total-right w-85 float-left align="right">
171 <p>Sub Total:</p>
172 <p>Total Pagable:</p>
173 </div>
174 <div class="total-right w-15 float-left text-bold align="right">
175 <p>S/. {{ $matriculaA->total_matricula }}</p>
176 <p>S/. {{ $matriculaA->total_matricula }}</p>
177 </div>
178 </div>
179 </td>
180 </tr>
181 </tbody>
182 </table>
183 </div>
184 </div>
185 </html>

```

Nota. Código PHP de vista Matrícula (pdf.blade.php).

Figura 66: Vistas Registrar Matrícula (pdf.blade.php).

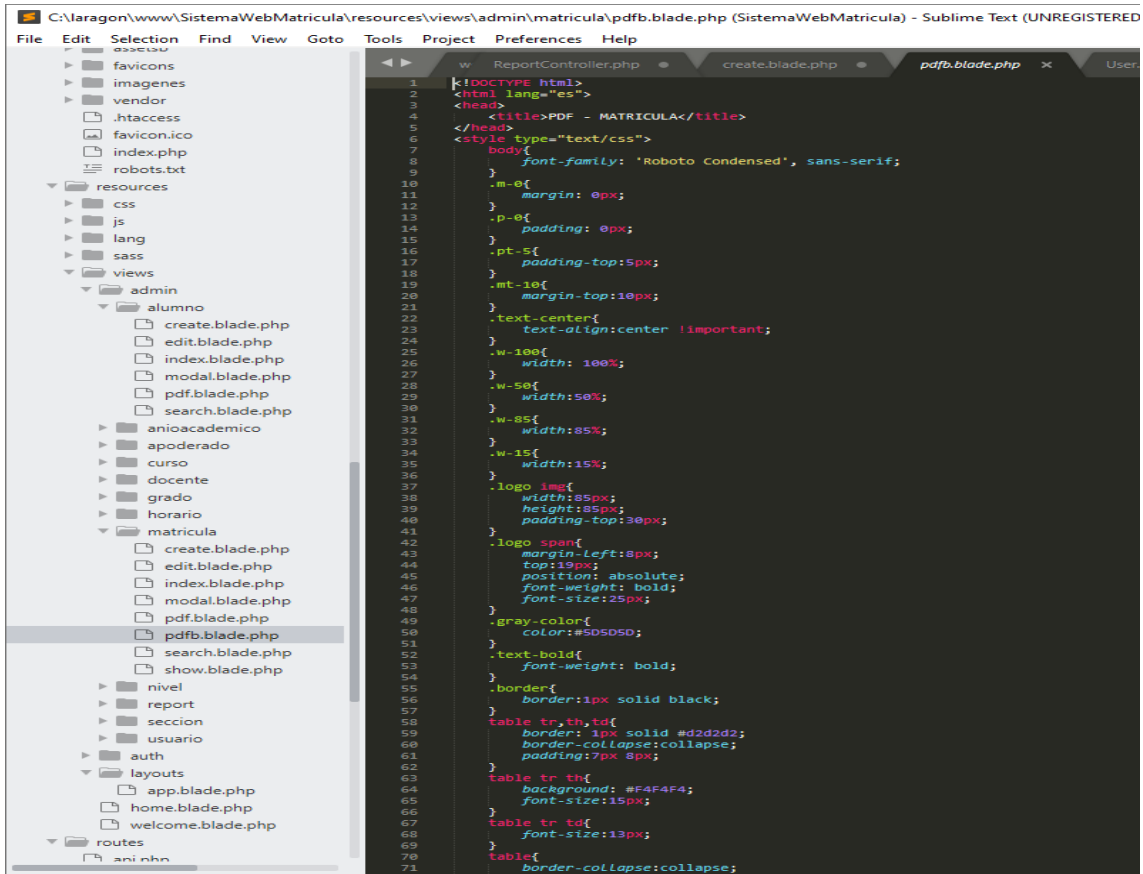
```

138 <p>{{ $matriculaA->direccion }}</p>
139 <p>{{ $matriculaA->celular }}</p>
140 </div>
141 </tr>
142 </tbody>
143 </table>
144 </div>
145 <p class="m-0 pt-5 text-bold w-100">Curso al que está matriculado :<span class="
146 gray-color">{{ $matriculaA->curso }}</span></p>
147 <div class="table-section bill-tbl w-100 mt-10">
148 <table class="table w-100 mt-10">
149 <thead>
150 <tr>
151 <th class="w-50">Curso</th>
152 <th class="w-50">Año Académico</th>
153 <th class="w-50">Sección</th>
154 <th class="w-50">Horario</th>
155 <th class="w-50">Estado</th>
156 <th class="w-50">Costo</th>
157 <th class="w-50">Sub Total</th>
158 </tr>
159 </thead>
160 <tbody>
161 <tr>
162 <td colspan="7">
163 <div class="total-part">
164 <div class="total-right w-85 float-left align="right">
165 <p>Sub Total:</p>
166 <p>Total Pagable:</p>
167 </div>
168 <div class="total-right w-15 float-left text-bold align="right">
169 <p>S/. {{ $matriculaA->total_matricula }}</p>
170 <p>S/. {{ $matriculaA->total_matricula }}</p>
171 </div>
172 </div>
173 </td>
174 </tr>
175 </tbody>
176 </table>
177 </div>
178 </div>
179 </html>

```

Nota. Código PHP de vista Matrícula (pdf.blade.php).

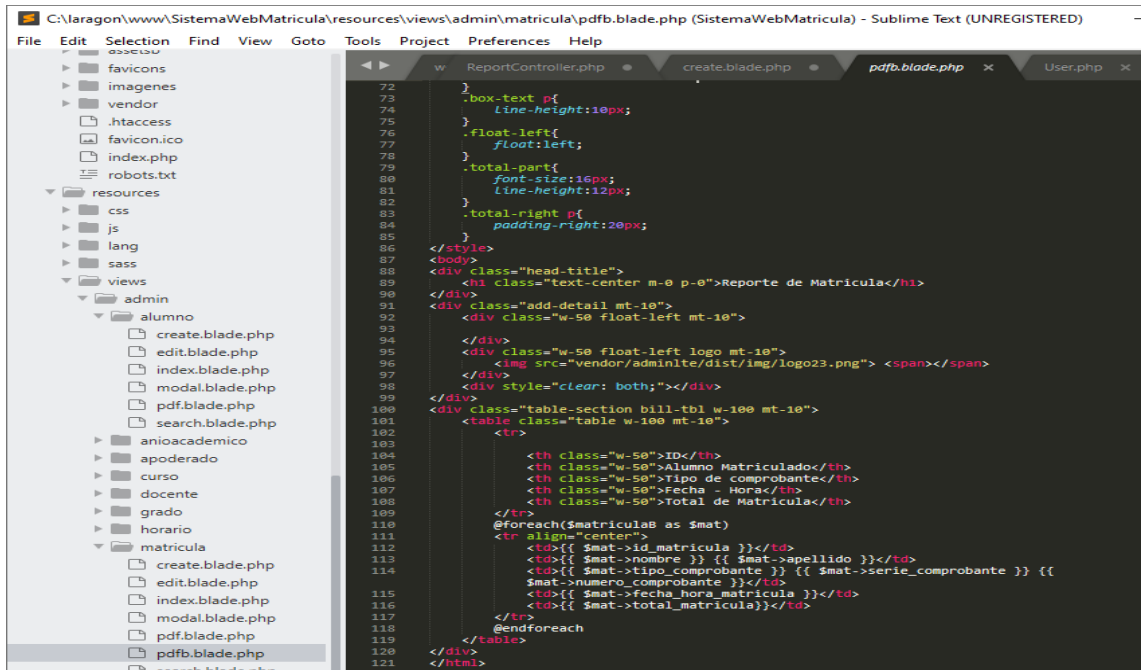
Figura 67: Vistas Registrar Matrícula (pdfb.blade.php).



```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <title>PDF - MATRICULA</title>
5 </head>
6 <style type="text/css">
7 body{
8     font-family: 'Roboto Condensed', sans-serif;
9 }
10
11 .m-0{
12     margin: 0px;
13 }
14
15 .p-0{
16     padding: 0px;
17 }
18
19 .pt-5{
20     padding-top:5px;
21 }
22
23 .mt-10{
24     margin-top:10px;
25 }
26
27 .text-center{
28     text-align:center !important;
29 }
30
31 .w-100{
32     width: 100%;
33 }
34
35 .w-50{
36     width:50%;
37 }
38
39 .w-85{
40     width:85%;
41 }
42
43 .w-15{
44     width:15%;
45 }
46
47 .logo_img{
48     width:85px;
49     height:85px;
50     padding-top:30px;
51 }
52
53 .logo_span{
54     margin-left:8px;
55     top:19px;
56     position: absolute;
57     font-weight: bold;
58     font-size:25px;
59 }
60
61 .gray-color{
62     color:#505050;
63 }
64
65 .text-bold{
66     font-weight: bold;
67 }
68
69 .border{
70     border:1px solid black;
71 }
72
73 table tr,th,td{
74     border: 1px solid #d2d2d2;
75     border-collapse:collapse;
76     padding:7px 8px;
77 }
78
79 table tr th{
80     background: #F4F4F4;
81     font-size:15px;
82 }
83
84 table tr td{
85     font-size:13px;
86 }
87
88 table{
89     border-collapse:collapse;
```

Nota. Código PHP de vista Matrícula (pdfb.blade.php).

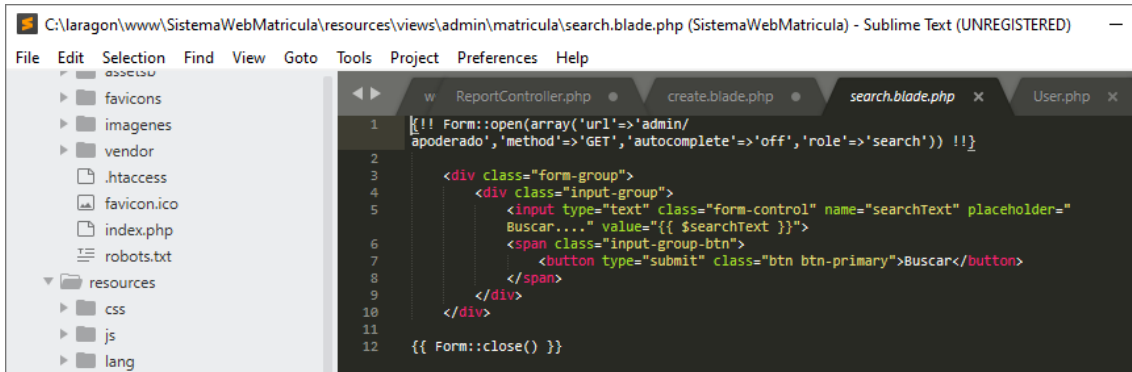
Figura 68: Vistas Registrar Matrícula (pdfb.blade.php).



```
72 }
73 .box-text p{
74     line-height:10px;
75 }
76
77 .float-left{
78     float:left;
79 }
80
81 .total-part{
82     font-size:15px;
83     line-height:12px;
84 }
85
86 .total-right p{
87     padding-right:20px;
88 }
89
90 </style>
91 <body>
92 <div class="head-title">
93 <h1 class="text-center m-0 p-0">Reporte de Matricula</h1>
94 </div>
95 <div class="add-detail mt-10">
96 <div class="w-50 float-left mt-10">
97 </div>
98 <div class="w-50 float-left logo mt-10">
99  <span></span>
100 </div>
101 <div style="clear: both;"></div>
102 </div>
103 <div class="table-section bill-tbl w-100 mt-10">
104 <table class="table w-100 mt-10">
105 <thead>
106 <tr>
107 <th class="w-50">ID</th>
108 <th class="w-50">Alumno Matriculado</th>
109 <th class="w-50">Tipo de comprobante</th>
110 <th class="w-50">Fecha - Hora</th>
111 <th class="w-50">Total de Matricula</th>
112 </tr>
113 </thead>
114 <tbody>
115 <tr>
116 <td align="center">
117 <td>{{ $mat->id_matricula }}</td>
118 <td>{{ $mat->nombre }} {{ $mat->apellido }}</td>
119 <td>{{ $mat->tipo_comprobante }} {{ $mat->serie_comprobante }}</td>
120 <td>{{ $mat->numero_comprobante }}</td>
121 <td>{{ $mat->fecha_hora_matricula }}</td>
122 <td>{{ $mat->total_matricula }}</td>
123 </tr>
124 </tbody>
125 </table>
126 </div>
127 </html>
```

Nota. Código PHP de vista Matrícula (pdfb.blade.php).

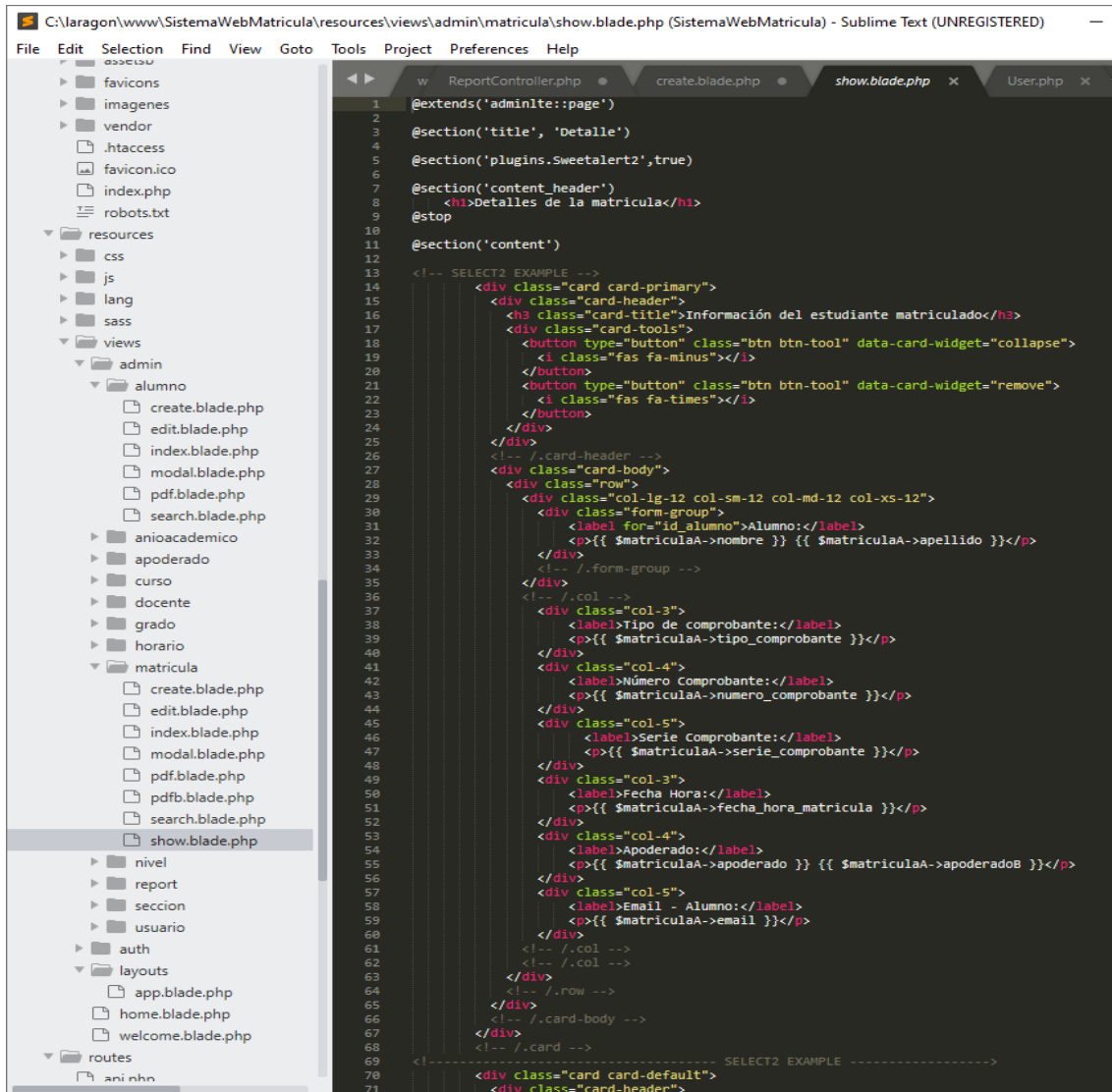
Figura 69: Vistas Registrar Matrícula (search.blade.php).



```
1 {!! Form::open(array('url'=>'admin/
apoderado','method'=>'GET','autocomplete'=>'off','role'=>'search')) !!}
2
3
4 <div class="form-group">
5 <div class="input-group">
6 <input type="text" class="form-control" name="searchText" placeholder="
Buscar..." value="{{ $searchText }}">
7 <span class="input-group-btn">
8 <button type="submit" class="btn btn-primary">Buscar</button>
9 </span>
10 </div>
11 </div>
12 {!! Form::close() !!}
```

Nota. Código PHP de vista Matrícula (search.blade.php).

Figura 70: Vistas Registrar Matrícula (show.blade.php).



```
1 @extends('adminlte:page')
2
3
4 @section('title', 'Detalle')
5
6 @section('plugins/sweetalert2',true)
7
8 @section('content_header')
9 <h1>Detalles de la matricula</h1>
10 @stop
11
12 @section('content')
13
14 <!-- SELECT2 EXAMPLE -->
15 <div class="card card-primary">
16 <div class="card-header">
17 <h3 class="card-title">Información del estudiante matriculado</h3>
18 <div class="card-tools">
19 <button type="button" class="btn btn-tool" data-card-widget="collapse">
20 <i class="fas fa-minus"></i>
21 </button>
22 <button type="button" class="btn btn-tool" data-card-widget="remove">
23 <i class="fas fa-times"></i>
24 </button>
25 </div>
26 </div>
27 <!-- /.card-header -->
28 <div class="card-body">
29 <div class="row">
30 <div class="col-lg-12 col-sm-12 col-md-12 col-xs-12">
31 <div class="form-group">
32 <label for="id_alumno">Alumno:</label>
33 <p>{{ $matriculaA->nombre }} {{ $matriculaA->apellido }}</p>
34 </div>
35 <!-- /.form-group -->
36 </div>
37 <!-- /.col -->
38 <div class="col-3">
39 <label>Tipo de comprobante:</label>
40 <p>{{ $matriculaA->tipo_comprobante }}</p>
41 </div>
42 <div class="col-4">
43 <label>Número Comprobante:</label>
44 <p>{{ $matriculaA->numero_comprobante }}</p>
45 </div>
46 <div class="col-5">
47 <label>Serie Comprobante:</label>
48 <p>{{ $matriculaA->serie_comprobante }}</p>
49 </div>
50 <div class="col-3">
51 <label>Fecha Hora:</label>
52 <p>{{ $matriculaA->fecha_hora_matricula }}</p>
53 </div>
54 <div class="col-4">
55 <label>Aporado:</label>
56 <p>{{ $matriculaA->apoderado }} {{ $matriculaA->apoderadoB }}</p>
57 </div>
58 <div class="col-5">
59 <label>Email - Alumno:</label>
60 <p>{{ $matriculaA->email }}</p>
61 </div>
62 <!-- /.col -->
63 <!-- /.col -->
64 <!-- /.row -->
65 </div>
66 </div>
67 </div>
68 <!-- /.card -->
69
70 <!-- SELECT2 EXAMPLE -->
71 <div class="card card-default">
72 <div class="card-header">
```

Nota. Código PHP de vista Matrícula (show.blade.php).

Figura 71: Vistas Registrar Matrícula (show.blade.php).

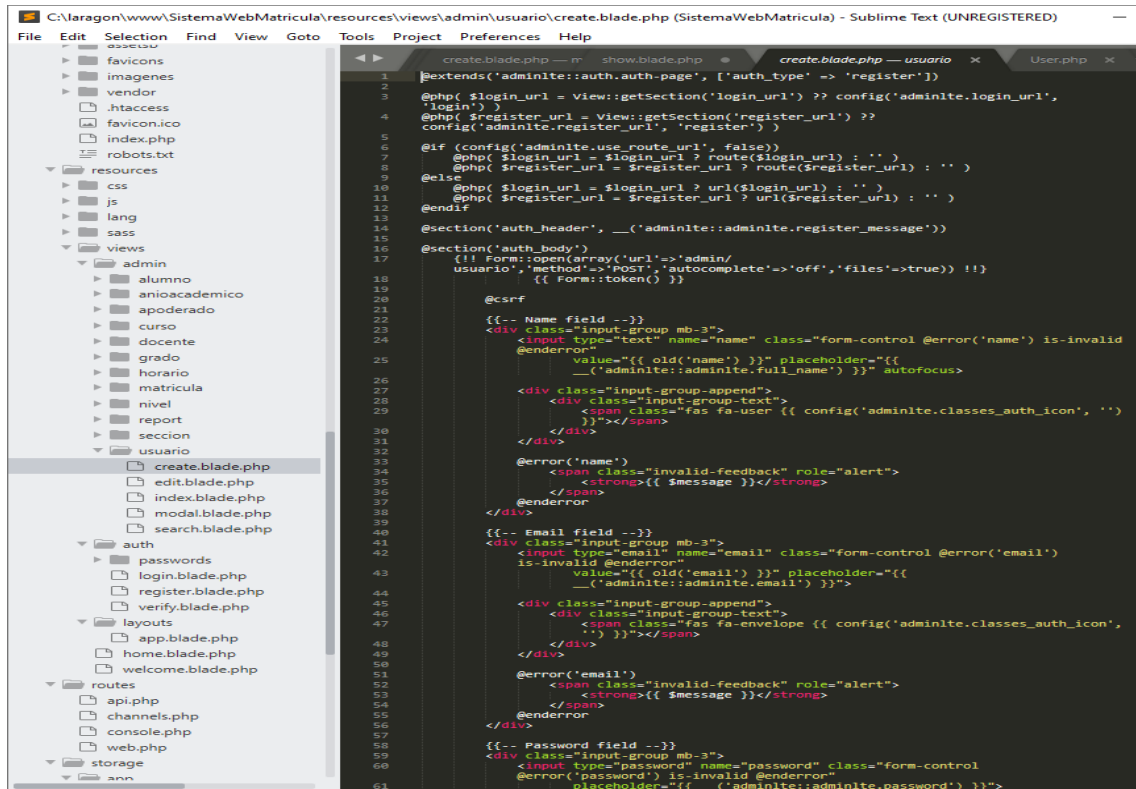
```

73 <h3 class="card-title">CURSOS AL QUE ESTÁ MATRICULADO</h3>
74
75 <div class="card-tools">
76 <button type="button" class="btn btn-tool" data-card-widget="collapse">
77 <i class="fas fa-minus"></i>
78 </button>
79 <button type="button" class="btn btn-tool" data-card-widget="remove">
80 <i class="fas fa-times"></i>
81 </button>
82 </div>
83 </div>
84 <!-- ÁCA ES PARA EL MAESTRO DETALLE-->
85 <div class="card-body">
86 <!-- ÁCA ES PARA EL MAESTRO DETALLE-->
87 <table id="detalles" class="table table-striped shadow-lg mt-4" style="
88 width:100%;">
89 <thead class="bg-primary text-white">
90 <tr>
91 <th>Curso</th>
92 <th>Año Académico</th>
93 <th>Seccion</th>
94 <th>Horario</th>
95 <th>Estado</th>
96 <th>Costo</th>
97 <th>SubTotal</th>
98 </tr>
99 </thead>
100 <tfoot>
101 <tr>
102 <th>Total:</th>
103 <td></td>
104 <th></th>
105 <th></th>
106 <th><h4 id="total">S/.{{ $matriculaA->total_matricula }}</h4></th>
107 </tr>
108 </tfoot>
109 <tbody>
110 @foreach($detallesA as $det)
111 <tr>
112 <td>{{ $det->nombre }}</td>
113 <td>{{ $det->descripcion }}</td>
114 <td>{{ $det->grado }} - {{ $det->nombre_seccion }} - {{
115 $det->nivel }}</td>
116 <td>{{ $det->dia }} - {{ $det->hora_inicio }} - {{
117 $det->hora_fin }}</td>
118 <td>{{ $det->estado }}</td>
119 <td>{{ $det->costo_matricula }}</td>
120 <td>{{ $det->costo_matricula }}</td>
121 </tr>
122 @endforeach
123 </tbody>
124 </table><br>
125 <!-- FIN DEL MAESTRO DETALLE-->
126 </div>
127 <!-- /.card-body ---->
128 <!-- FIN DE LO QUE HE CREADO YO ----->
129 </div>
130 <!-- /.content -->
@endsection

```

Nota. Código PHP de vista Matrícula (show.blade.php).

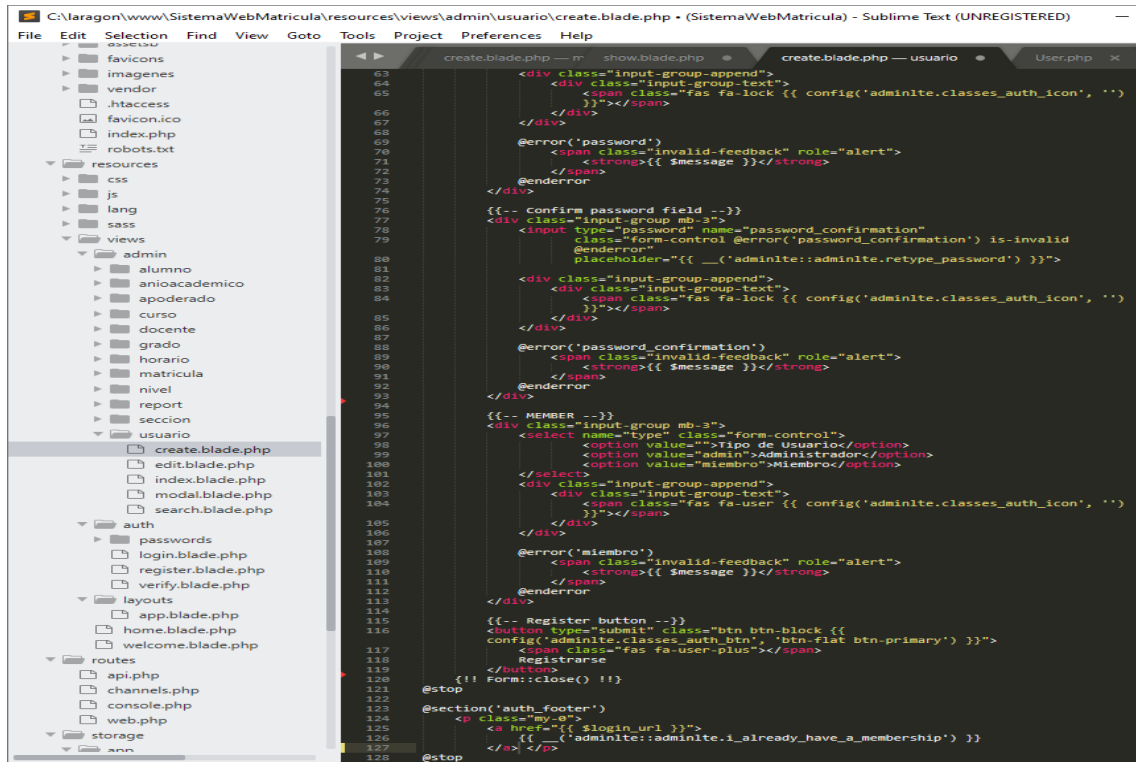
Figura 72: Vistas de Perfil Usuario (create.blade.php).



```
1 @extends('adminlte:auth.auth-page', ['auth_type' => 'register'])
2
3 @php($login_url = View::getSection('login_url')) ?? config('adminlte.login_url',
4 'login')
5 @php($register_url = View::getSection('register_url')) ??
6 config('adminlte.register_url', 'register')
7
8 if (config('adminlte.use_route_url', false))
9 @php($login_url = $login_url ? route($login_url) : '')
10 @php($register_url = $register_url ? route($register_url) : '')
11 else
12 @php($login_url = $login_url ? url($login_url) : '')
13 @php($register_url = $register_url ? url($register_url) : '')
14 @endif
15
16 @section('auth_header', __('adminlte:adminlte.register_message'))
17
18 @section('auth_body')
19 {!! Form::open(array('url' => 'admin/
20 usuario', 'method' => 'POST', 'autocomplete' => 'off', 'files' => true)) !!}
21 {{ Form::token() }}
22
23 @csrf
24
25 {{-- Name field --}}
26 <div class="input-group mb-3">
27 <input type="text" name="name" class="form-control @error('name') is-invalid
28 @enderror" value="{{ old('name') }}" placeholder="{{
29 __('adminlte:adminlte.full_name') }}" autofocus>
30
31 <div class="input-group-append">
32 <div class="input-group-text">
33 <span class="fas fa-user" {{ config('adminlte.classes_auth_icon', '')
34 }}></span>
35 </div>
36 </div>
37
38 @error('name')
39 <span class="invalid-feedback" role="alert">
40 <strong>{{ $message }}</strong>
41 </span>
42 @enderror
43
44 {{-- Email field --}}
45 <div class="input-group mb-3">
46 <input type="email" name="email" class="form-control @error('email')
47 is-invalid @enderror" value="{{ old('email') }}" placeholder="{{
48 __('adminlte:adminlte.email') }}">
49
50 <div class="input-group-append">
51 <div class="input-group-text">
52 <span class="fas fa-envelope" {{ config('adminlte.classes_auth_icon',
53 '') }}></span>
54 </div>
55 </div>
56
57 @error('email')
58 <span class="invalid-feedback" role="alert">
59 <strong>{{ $message }}</strong>
60 </span>
61 @enderror
62
63 {{-- Password field --}}
64 <div class="input-group mb-3">
65 <input type="password" name="password" class="form-control
66 @error('password') is-invalid @enderror"
67 placeholder="{{ __('adminlte:adminlte.password') }}">
68
69 <div class="input-group-append">
70 <div class="input-group-text">
71 <span class="fas fa-lock" {{ config('adminlte.classes_auth_icon', '')
72 }}></span>
73 </div>
74 </div>
75
76 @error('password')
77 <span class="invalid-feedback" role="alert">
78 <strong>{{ $message }}</strong>
79 </span>
80 @enderror
81
82 {{-- Confirm password field --}}
83 <div class="input-group mb-3">
84 <input type="password" name="password_confirmation" class="form-control
85 @error('password_confirmation') is-invalid
86 @enderror"
87 placeholder="{{ __('adminlte:adminlte.retype_password') }}">
88
89 <div class="input-group-append">
90 <div class="input-group-text">
91 <span class="fas fa-lock" {{ config('adminlte.classes_auth_icon', '')
92 }}></span>
93 </div>
94 </div>
95
96 @error('password_confirmation')
97 <span class="invalid-feedback" role="alert">
98 <strong>{{ $message }}</strong>
99 </span>
100 @enderror
101
102 {{-- MEMBER --}}
103 <div class="input-group mb-3">
104 <select name="type" class="form-control">
105 <option value="">Tipo de Usuario</option>
106 <option value="admin">Administrador</option>
107 <option value="miembro">Miembro</option>
108 </select>
109
110 <div class="input-group-append">
111 <div class="input-group-text">
112 <span class="fas fa-user" {{ config('adminlte.classes_auth_icon', '')
113 }}></span>
114 </div>
115 </div>
116
117 @error('miembro')
118 <span class="invalid-feedback" role="alert">
119 <strong>{{ $message }}</strong>
120 </span>
121 @enderror
122
123 {{-- Register button --}}
124 <button type="submit" class="btn btn-block {{
125 config('adminlte.classes_auth_btn', 'btn-flat btn-primary') }}">
126 Registrar
127 </button>
128
129 @stop
130
131 @section('auth_footer')
132 <p class="my-0">
133 <a href="{{ $login_url }}">
134 {{ __('adminlte:adminlte.i_already_have_a_membership') }}
135 </a>
136 </p>
137 @stop
```

Nota. Código PHP de vista Usuario (create.blade.php).

Figura 73: Vistas de Perfil Usuario (create.blade.php).



```
63 <div class="input-group-append">
64 <div class="input-group-text">
65 <span class="fas fa-lock" {{ config('adminlte.classes_auth_icon', '')
66 }}></span>
67 </div>
68 </div>
69
70 @error('password')
71 <span class="invalid-feedback" role="alert">
72 <strong>{{ $message }}</strong>
73 </span>
74 @enderror
75
76 {{-- Confirm password field --}}
77 <div class="input-group mb-3">
78 <input type="password" name="password_confirmation" class="form-control
79 @error('password_confirmation') is-invalid
80 @enderror"
81 placeholder="{{ __('adminlte:adminlte.retype_password') }}">
82
83 <div class="input-group-append">
84 <div class="input-group-text">
85 <span class="fas fa-lock" {{ config('adminlte.classes_auth_icon', '')
86 }}></span>
87 </div>
88 </div>
89
90 @error('password_confirmation')
91 <span class="invalid-feedback" role="alert">
92 <strong>{{ $message }}</strong>
93 </span>
94 @enderror
95
96 {{-- MEMBER --}}
97 <div class="input-group mb-3">
98 <select name="type" class="form-control">
99 <option value="">Tipo de Usuario</option>
100 <option value="admin">Administrador</option>
101 <option value="miembro">Miembro</option>
102 </select>
103
104 <div class="input-group-append">
105 <div class="input-group-text">
106 <span class="fas fa-user" {{ config('adminlte.classes_auth_icon', '')
107 }}></span>
108 </div>
109 </div>
110
111 @error('miembro')
112 <span class="invalid-feedback" role="alert">
113 <strong>{{ $message }}</strong>
114 </span>
115 @enderror
116
117 {{-- Register button --}}
118 <button type="submit" class="btn btn-block {{
119 config('adminlte.classes_auth_btn', 'btn-flat btn-primary') }}">
120 Registrar
121 </button>
122
123 @stop
124
125 @section('auth_footer')
126 <p class="my-0">
127 <a href="{{ $login_url }}">
128 {{ __('adminlte:adminlte.i_already_have_a_membership') }}
129 </a>
130 </p>
131 @stop
```

Nota. Código PHP de vista Usuario (create.blade.php).

Figura 74: Vistas de Perfil Usuario (edit.blade.php).

```

1 @extends('adminlte::page')
2
3 @section('title', 'usuario')
4
5 @section('content_header')
6 <h2>Editar Usuario: {{ $usuario->name }}</h2>
7 @if (count($errors)>0)
8 <div class="alert alert-danger">
9 <ul>
10 @foreach($errors->all() as $errors)
11 <li>{{ $errors }}</li>
12 @endforeach
13 </ul>
14 </div>
15 @endif
16 @stop
17
18 @section('content')
19 <div class="container">
20 <div class="card">
21 <div class="card-header">
22 <h3 class="card-title"><small> En este formulario se podrán editar a
23 los usuarios registrados</small></h3>
24 </div>
25 <div class="card-body">
26 <div class="form-group row">
27 <label for="name" class="col-md-4 col-form-label
28 text-md-right">{{ __('Nombre') }}</label>
29
30 <input id="name" type="text" class="form-control
31 @error('name') is-invalid @enderror" name="name" value=
32 "{{ $usuario->name }}" required autocomplete="name"
33 autofocus>
34
35 @error('name')
36 <span class="invalid-feedback" role="alert">
37 <strong>{{ $message }}</strong>
38 </span>
39 @enderror
40 </div>
41
42 <div class="form-group row">
43 <label for="email" class="col-md-4 col-form-label
44 text-md-right">{{ __('E-Mail Address') }}</label>
45
46 <input id="email" type="email" class="form-control
47 @error('email') is-invalid @enderror" name="email"
48 value="{{ $usuario->email }}" required autocomplete="
49 email">
50
51 @error('email')
52 <span class="invalid-feedback" role="alert">
53 <strong>{{ $message }}</strong>
54 </span>
55 @enderror
56
57
58
59
60
61

```

Nota. Código PHP de vista Usuario (edit.blade.php).

Figura 75: Vistas de Perfil Usuario (edit.blade.php).

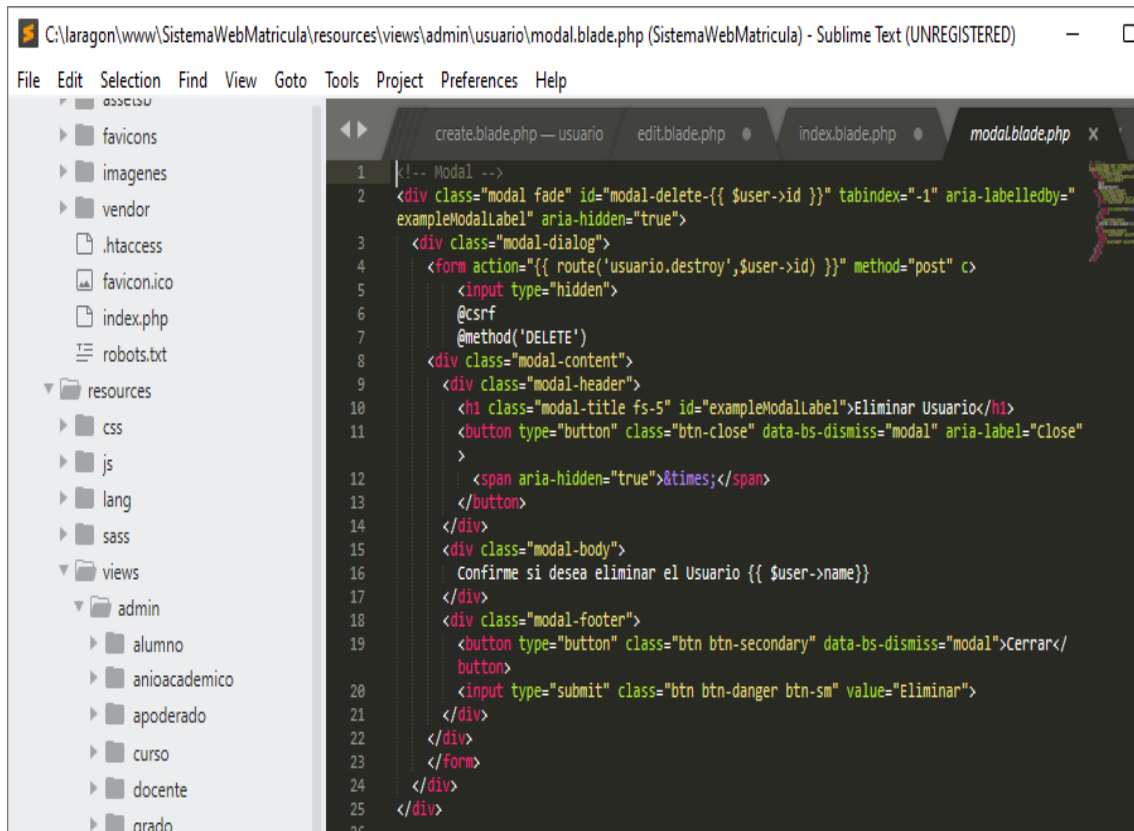
```

62 </div>
63 <div class="form-group row">
64 <label for="password" class="col-md-4 col-form-label
65 text-md-right">{{ __('Password') }}</label>
66
67 <input id="password" type="password" class="
68 form-control @error('password') is-invalid @enderror"
69 name="password" required autocomplete="new-password">
70
71 @error('password')
72 <span class="invalid-feedback" role="alert">
73 <strong>{{ $message }}</strong>
74 </span>
75 @enderror
76 </div>
77
78 <div class="form-group row">
79 <label for="password-confirm" class="col-md-4
80 col-form-label text-md-right">{{ __('Confirmar Password')
81 }}</label>
82
83 <input id="password-confirm" type="password" class="
84 form-control" name="password_confirmation" required
85 autocomplete="new-password">
86 </div>
87
88 <div class="form-group row">
89 <label for="type" class="col-md-4 col-form-label
90 text-md-right">{{ __('Tipo de Usuario') }}</label>
91
92 <select name="type" class="form-control">
93 <option value="admin">Administrador</option>
94 <option value="miembro">Miembro</option>
95 </select>
96 </div>
97 </div>
98 <div class="card-footer">
99 <button class="btn btn-primary" type="submit">Guardar</button>
100 <button class="btn btn-danger" type="reset">Cancelar</button>
101 </div>
102 </div>
103 </div>
104 <div class="row">
105 <div class="col-md-6">
106
107 </div>
108 <div class="col-md-6">
109
110 </div>
111 </div>
112 </div>
113 </div>
114 <div class="row">
115 <div class="col">
116 <div class="card">
117 <div class="card-body">
118 <div class="card">
119 <div class="card-body">
120 <div class="card">
121 <div class="card">
122 <div class="card">
123 <div class="card">

```

Nota. Código PHP de vista Usuario (edit.blade.php).

Figura 78: Vistas de Perfil Usuario (*modal.blade.php*).



```
1 |<!-- Modal -->
2 <div class="modal fade" id="modal-delete-{{ $user->id }}" tabindex="-1" aria-labelledby="
exampleModallabel" aria-hidden="true">
3 <div class="modal-dialog">
4 <form action="{{ route('usuario.destroy',$user->id) }}" method="post" c>
5 <input type="hidden">
6 @csrf
7 @method('DELETE')
8 <div class="modal-content">
9 <div class="modal-header">
10 <h1 class="modal-title fs-5" id="exampleModallabel">Eliminar Usuario</h1>
11 <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"
>
12 <span aria-hidden="true">&times;</span></button>
13 </div>
14 <div class="modal-body">
15 Confirme si desea eliminar el Usuario {{ $user->name}}
16 </div>
17 <div class="modal-footer">
18 <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</
button>
19 <input type="submit" class="btn btn-danger btn-sm" value="Eliminar">
20 </div>
21 </div>
22 </form>
23 </div>
24 </div>
25 </div>
26
```

Nota. Código PHP de vista Usuario (*modal.blade.php*).

Figura 79: Vistas de Perfil Usuario (*search.blade.php*).



```
1 {!! Form::open(array('url'=>'admin/
apoderado','method'=>'GET','autocomplete'=>'off','role'=>'search')) !!}
2
3 <div class="form-group">
4 <div class="input-group">
5 <input type="text" class="form-control" name="searchText" placeholder="
Buscar..." value="{{ $searchText }}">
6 <span class="input-group-btn">
7 <button type="submit" class="btn btn-primary">Buscar</button>
8 </span>
9 </div>
10 </div>
11
12 {!! Form::close() !!}
```

Nota. Código PHP de vista Usuario (*search.blade.php*).

Figura 80: Vistas de Registrar Apoderado (create.blade.php).

```

1 @extends('adminlte::page')
2
3 @section('title', 'Apoderado')
4
5 @section('plugins.SweetAlert2', true)
6
7 @section('content_header')
8 <h1>Nuevo Apoderado</h1>
9 @if (count($errors))
10 <div class="alert alert-danger">
11 <ul>
12 @foreach($errors->all() as $errors)
13 <li>{{ $errors }}</li>
14 @endforeach
15 </ul>
16 </div>
17 @endif
18 @stop
19
20 @section('content')
21 <!-- main content -->
22 <section class="content">
23 <div class="container-fluid">
24 <div class="row">
25 <!-- left column -->
26 <div class="col-md-12">
27 <!-- jquery validation -->
28 <div class="card card-primary">
29 <div class="card-header">
30 <h3 class="card-title"><small></small> En este formulario se registrarán los
31 padres de los distintos alumnos</small></h3>
32 </div>
33 <!-- card header -->
34 <!-- form start -->
35 {!! Form::open(array('url' => 'admin/apoderado', 'method' => 'POST', 'autocomplete' => 'off', 'files' => true)) !!}
36 {!! Form::token() !!}
37 <form id="quickForm">
38 <div class="card-body">
39 <div class="form-group">
40 <label for="nombre">Nombre:</label>
41 <input type="text" name="nombre" class="form-control" id="
42 exampleInputEmail" placeholder="Ingrese nombre">
43 </div>
44 <div class="form-group">
45 <label for="apellido">Apellido:</label>
46 <input type="text" name="apellido" class="form-control" id="
47 exampleInputEmail" placeholder="Ingrese apellido">
48 </div>
49 <div class="form-group">
50 <label for="sexo">Sexo:</label>
51 <select name="sexo" class="form-control">
52 <option value="Masculino">Masculino</option>
53 <option value="Femenino">Femenino</option>
54 </select>
55 </div>
56 <div class="form-group">
57 <label for="dni">DNI:</label>
58 <input type="text" name="dni" class="form-control" id="
59 exampleInputEmail" placeholder="Ingrese DNI">
60 </div>
61 <div class="form-group">
62 <label for="direccion">Dirección:</label>
63 <input type="text" name="direccion" class="form-control" id="
64 exampleInputEmail" placeholder="Ingrese direccion">
65 </div>
66 <div class="form-group">
67 <label for="email">Email address:</label>
68 <input type="email" name="email" class="form-control" id="
69 exampleInputEmail" placeholder="Enter email">
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 <div class="card-footer">
79 <button class="btn btn-primary" type="submit" id="btnOpenSalt" >
80 guardar</button>
81 <button class="btn btn-danger" type="reset">Cancelar</button>
82 </div>
83 </form>
84 </div>
85 </div>
86 <!-- /card -->
87 </div>
88 <!-- /col (left) -->
89 <!-- right column -->
90 <div class="col-md-6">
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 <!-- /content -->
97 {!! Form::close() !!}
98 @stop
99
100 @section('css')
101 <link rel="stylesheet" href="/css/admin_custom.css">
102 @stop
103
104 @section('js')
105 <script> console.log('Hi!'); </script>
106 <script>
107 (document).ready(function() {
108 $(&#34;#btnOpenSalt&#34;).click(function() {
109 Swal.fire({
110 title: '¡Bien trabajo!',
111 text: 'El apoderado ha sido registrado con éxito!',
112 icon: 'success'
113 });
114 });
115 var Toast = Swal.mixin({
116 toast: true,
117 position: 'top-end',
118 showConfirmButton: false,
119 timer: 3000
120 });
121 $(&#34;#btnOpenSalt&#34;).click(function() {
122 Toast.fire({
123 icon: 'success',
124 title: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr.'
125 });
126 });
127 </script>
128 </div>
129 @stop
130
131

```

Nota. Código PHP de vista Registrar Apoderado (create.blade.php).

Figura 81: Vistas de Registrar Apoderado (create.blade.php).

```

66 </div>
67 <div class="form-group">
68 <label for="celular">Celular:</label>
69 <input type="text" name="celular" class="form-control" id="
70 exampleInputPassword" placeholder="Ingrese celular">
71 </div>
72 <!-- Para agregar imagenes -->
73 <div class="form-group">
74 {!! Form::label('Imagen', 'Imagen') !!}
75 {!! Form::file('imagen') !!}
76 </div>
77 </div>
78 <div class="card-footer">
79 <button class="btn btn-primary" type="submit" id="btnOpenSalt" >
80 guardar</button>
81 <button class="btn btn-danger" type="reset">Cancelar</button>
82 </div>
83 </form>
84 </div>
85 </div>
86 <!-- /card -->
87 </div>
88 <!-- /col (left) -->
89 <!-- right column -->
90 <div class="col-md-6">
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 <!-- /content -->
97 {!! Form::close() !!}
98 @stop
99
100 @section('css')
101 <link rel="stylesheet" href="/css/admin_custom.css">
102 @stop
103
104 @section('js')
105 <script> console.log('Hi!'); </script>
106 <script>
107 (document).ready(function() {
108 $(&#34;#btnOpenSalt&#34;).click(function() {
109 Swal.fire({
110 title: '¡Bien trabajo!',
111 text: 'El apoderado ha sido registrado con éxito!',
112 icon: 'success'
113 });
114 });
115 var Toast = Swal.mixin({
116 toast: true,
117 position: 'top-end',
118 showConfirmButton: false,
119 timer: 3000
120 });
121 $(&#34;#btnOpenSalt&#34;).click(function() {
122 Toast.fire({
123 icon: 'success',
124 title: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr.'
125 });
126 });
127 </script>
128 </div>
129 @stop
130
131

```

Nota. Código PHP de vista Registrar Apoderado (create.blade.php).

Figura 82: Vistas de Registrar Apoderado (edit.blade.php).

```
1 @extends('adminlte::page')
2
3 @section('title', 'Apoderado')
4
5 @section('content_header')
6 <!-- Editor Apoderado: {{ $apoderadoC->nombre }}</!-->
7 @if (count($errors)>0)
8 <div class="alert alert-danger">
9 <ul>
10 @foreach($errors->all() as $errors)
11 <li>{{ $errors }}</li>
12 @endforeach
13 </ul>
14 </div>
15 @endif
16 @stop
17
18 @section('content')
19 <!-- Main content -->
20 <section class="content">
21 <div class="container-fluid">
22 <div class="row">
23 <div class="col-md-12">
24 <!-- jQuery validation -->
25 <div class="card card-primary">
26 <!-- card header -->
27 <!-- form start -->
28 {!! Form::open(array('route' =>
29 'apoderado.update', $apoderadoC->id), 'method' => 'PUT', 'files' => true)) !!}
30 {!! Form::token() !!}
31
32 <form id="quickForm">
33 <div class="card-body">
34 <div class="form-group">
35 <label for="nombre">Nombre:</label>
36 <input type="text" name="nombre" class="form-control" value="{{
37 $apoderadoC->nombre }}" id="exampleInputEmail" placeholder="Ingrese
38 nombre">
39 </div>
40 <div class="form-group">
41 <label for="apellido">Apellido:</label>
42 <input type="text" name="apellido" class="form-control" value="{{
43 $apoderadoC->apellido }}" id="exampleInputEmail" placeholder="
44 Ingrese apellido">
45 </div>
46 <div class="form-group">
47 <label for="sexo">Sexo:</label>
48 <input type="text" name="sexo" class="form-control" value="{{
49 $apoderadoC->sexo }}" id="exampleInputEmail" placeholder="Ingrese
50 sexo">
51 </div>
52 <div class="form-group">
53 <label for="DNI">DNI:</label>
54 <input type="text" name="DNI" class="form-control" value="{{
55 $apoderadoC->DNI }}" id="exampleInputEmail" placeholder="Ingrese DNI">
56 </div>
57 <div class="form-group">
58 <label for="direccion">Dirección:</label>
59 <input type="text" name="direccion" class="form-control" value="{{
60 $apoderadoC->direccion }}" id="exampleInputEmail" placeholder="
61 Ingrese direccion">
62 </div>
63 <div class="form-group">
64 <label for="exampleInputEmail">Email address:</label>
65 <input type="email" name="email" class="form-control" value="{{
66 $apoderadoC->email }}" id="exampleInputEmail" placeholder="Enter
67 email">
68 </div>
69 </div>
70 </div>
71 <div class="card-footer">
72 <button class="btn btn-primary" type="submit">Guardar</button>
73 <button class="btn btn-danger" type="reset">Cancelar</button>
74 </div>
75 </form>
76 </div>
77 </div>
78 </div>
79 </div>
80 <!-- /col (left) -->
81 <!-- right column -->
82 <div class="col-md-6">
83
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 <!-- /content -->
91 {!! Form::close() !!}
92 @stop
93
94 @section('css')
95 <link rel="stylesheet" href="/css/admin_custom.css">
96 @stop
97
98 @section('js')
99 <script> console.log('Hi!'); </script>
100 @stop
```

Nota. Código PHP de vista Registrar Apoderado (edit.blade.php).

Figura 83: Vistas de Registrar Apoderado (edit.blade.php).

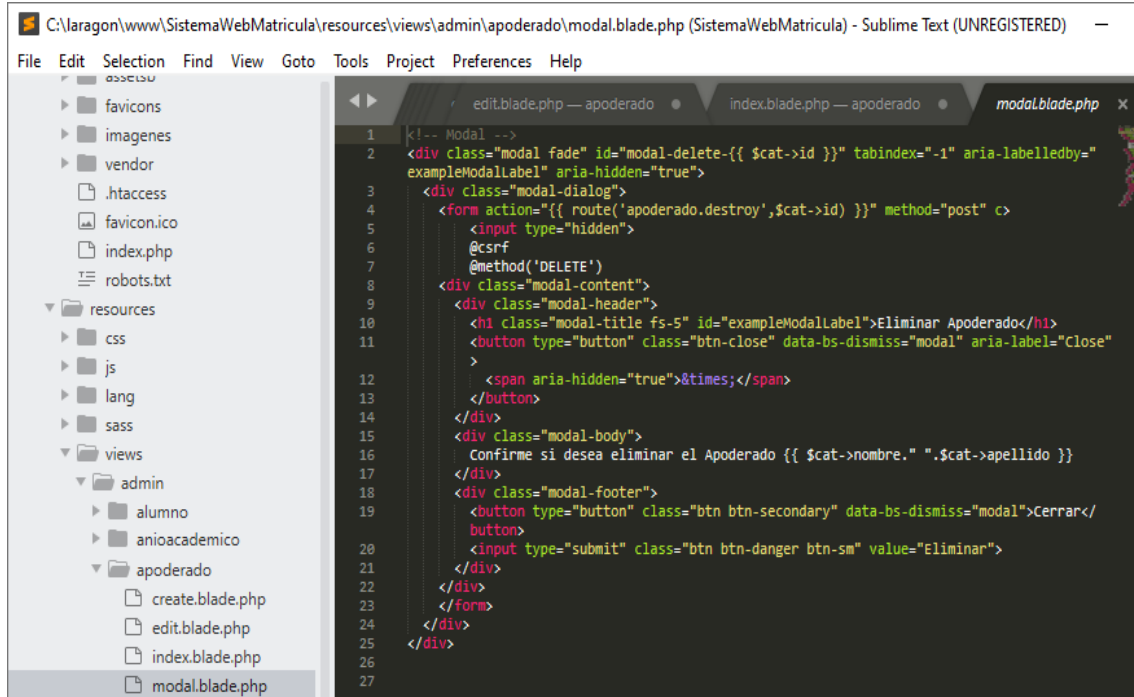
```
60 <label for="celular">celular:</label>
61 <input type="text" name="celular" class="form-control" value="{{
62 $apoderadoC->celular }}" id="exampleInputPassword1" placeholder="
63 Ingrese celular">
64 </div>
65 <div class="form-group">
66 <label for="imagen">Imagen:</label>
67 <input type="file" name="imagen" class="form-control">
68 @if (($apoderadoC->imagen))
69 
71 @endif
72 </div>
73 </div>
74 <div class="card-footer">
75 <button class="btn btn-primary" type="submit">Guardar</button>
76 <button class="btn btn-danger" type="reset">Cancelar</button>
77 </div>
78 </form>
79 </div>
80 </div>
81 </div>
82 <div class="col-md-6">
83
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 <!-- /content -->
91 {!! Form::close() !!}
92 @stop
93
94 @section('css')
95 <link rel="stylesheet" href="/css/admin_custom.css">
96 @stop
97
98 @section('js')
99 <script> console.log('Hi!'); </script>
100 @stop
```

Nota. Código PHP de vista Registrar Apoderado (edit.blade.php).

Figura 84: Vistas de Registrar Apoderado (index.blade.php).

```
1 @extends('adminite::page')
2
3 @section('title', 'Apoderado')
4
5 @section('content_header')
6 <h1>Listado de apoderados <!--<a href="apoderado/create"><button class="btn
7 btn-success">Nuevo</button></a--></h1>
8
9 @stop
10
11 @section('content')
12 <div class="card">
13 <div class="card-header">
14 <a href="apoderado/create" class="btn btn-primary">Nuevo Apoderado</a>
15 </div>
16 <div class="card-body">
17 <table id="example" class="table table-striped shadow-lg mt-4" style="width:
18 100%;>
19 <thead class="bg-primary text-white">
20 <tr>
21 <th>Nombre</th>
22 <th>Sexo</th>
23 <th>DNI</th>
24 <th>Dirección</th>
25 <th>Email</th>
26 <th>Celular</th>
27 <th>Imagen</th>
28 <th>Opciones</th>
29 </tr>
30 </thead>
31 <tbody>
32 @foreach($apoderadoA as $cat)
33 <tr>
34 <td>{{ $cat->nombre }}{{ $cat->apellido }}</td>
35 <td>{{ $cat->sexo }}</td>
36 <td>{{ $cat->DNI }}</td>
37 <td>{{ $cat->direccion }}</td>
38 <td>{{ $cat->email }}</td>
39 <td>{{ $cat->celular }}</td>
40 <td>
41 nombre }}" height="120px" width="120px" class="
43 img-thumbnail">
44 <a target="blank" href="/imagenes/apoderados/{{
45 $cat->imagen }}"> Abrir Foto</a>
46 </td>
47 <td>
48 <button width="250px">
49 <a href="{{ route('apoderado.edit',$cat->id) }}"><button
50 class="btn btn-primary btn-sm">Editar</button></a>
51 <!-- Button trigger modal -->
52 <button type="button" class="btn btn-danger btn-sm"
53 data-bs-toggle="modal" data-bs-target="#modal-delete-{{
54 $cat->id }}">
55 Eliminar
56 </button>
57 </td>
58 </tr>
59 @include('admin.apoderado.modal')
60 @endforeach
61 </tbody>
62 </table>
63 </div>
64 </div>
65 </tbody>
66 </table>
67 </div>
68 @stop
69
70 @section('css')
71 <link rel="stylesheet" href="/css/admin_custom.css">
72 <link rel="stylesheet" href="https://cdn.datatables.net/1.13.1/css/
73 datatables.bootstrap.min.css">
74 @stop
75
76 @section('js')
77 <script> console.log('Hi!'); </script>
78 <script src="https://code.jquery.com/jquery-3.5.1.js"></script>
79 <script src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>
80 <script src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap.min.js"></
81 script>
82
83 <script>
84 $(document).ready(function () {
85 <table>
86 <thead>
87 <tr>
88 <th>Search</th>
89 <th>lengthMenu</th>
90 <th>info</th>
91 <th>paginate</th>
92 </tr>
93 </thead>
94 </table>
95
96 <script>
97 <script>
98 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/
99 popper.min.js" integrity="sha384-8bqvmz9ATk1ep91kx5/
100 z9iffx0D9VJW8eAsJfucsm80SSq01mVjpa" crossorigin="anonymous"></script>
101 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
102 integrity="sha384-IDwe1+Lcz0S0q9k97z9dyv1+AES10+x7tBKgc9ISHTF9tZ0wNfPcl20v9vxnk"
103 crossorigin="anonymous"></script>
104 @stop
105
106 </script>
107
108 </script>
109
110 </script>
111
112 </script>
113
114 </script>
115
116 </script>
117
118 </script>
119
120 </script>
121
122 </script>
123
124 </script>
125
126 </script>
127
128 </script>
129
130 </script>
131
132 </script>
133
134 </script>
135
136 </script>
137
138 </script>
139
140 </script>
141
142 </script>
143
144 </script>
145
146 </script>
147
148 </script>
149
150 </script>
151
152 </script>
153
154 </script>
155
156 </script>
157
158 </script>
159
160 </script>
161
162 </script>
163
164 </script>
165
166 </script>
167
168 </script>
169
170 </script>
171
172 </script>
173
174 </script>
175
176 </script>
177
178 </script>
179
180 </script>
181
182 </script>
183
184 </script>
185
186 </script>
187
188 </script>
189
190 </script>
191
192 </script>
193
194 </script>
195
196 </script>
197
198 </script>
199
200 </script>
201
202 </script>
203
204 </script>
205
206 </script>
207
208 </script>
209
210 </script>
211
212 </script>
213
214 </script>
215
216 </script>
217
218 </script>
219
220 </script>
221
222 </script>
223
224 </script>
225
226 </script>
227
228 </script>
229
230 </script>
231
232 </script>
233
234 </script>
235
236 </script>
237
238 </script>
239
240 </script>
241
242 </script>
243
244 </script>
245
246 </script>
247
248 </script>
249
250 </script>
251
252 </script>
253
254 </script>
255
256 </script>
257
258 </script>
259
260 </script>
261
262 </script>
263
264 </script>
265
266 </script>
267
268 </script>
269
270 </script>
271
272 </script>
273
274 </script>
275
276 </script>
277
278 </script>
279
280 </script>
281
282 </script>
283
284 </script>
285
286 </script>
287
288 </script>
289
290 </script>
291
292 </script>
293
294 </script>
295
296 </script>
297
298 </script>
299
300 </script>
301
302 </script>
303
304 </script>
305
306 </script>
307
308 </script>
309
310 </script>
311
312 </script>
313
314 </script>
315
316 </script>
317
318 </script>
319
320 </script>
321
322 </script>
323
324 </script>
325
326 </script>
327
328 </script>
329
330 </script>
331
332 </script>
333
334 </script>
335
336 </script>
337
338 </script>
339
340 </script>
341
342 </script>
343
344 </script>
345
346 </script>
347
348 </script>
349
350 </script>
351
352 </script>
353
354 </script>
355
356 </script>
357
358 </script>
359
360 </script>
361
362 </script>
363
364 </script>
365
366 </script>
367
368 </script>
369
370 </script>
371
372 </script>
373
374 </script>
375
376 </script>
377
378 </script>
379
380 </script>
381
382 </script>
383
384 </script>
385
386 </script>
387
388 </script>
389
390 </script>
391
392 </script>
393
394 </script>
395
396 </script>
397
398 </script>
399
400 </script>
401
402 </script>
403
404 </script>
405
406 </script>
407
408 </script>
409
410 </script>
411
412 </script>
413
414 </script>
415
416 </script>
417
418 </script>
419
420 </script>
421
422 </script>
423
424 </script>
425
426 </script>
427
428 </script>
429
430 </script>
431
432 </script>
433
434 </script>
435
436 </script>
437
438 </script>
439
440 </script>
441
442 </script>
443
444 </script>
445
446 </script>
447
448 </script>
449
450 </script>
451
452 </script>
453
454 </script>
455
456 </script>
457
458 </script>
459
460 </script>
461
462 </script>
463
464 </script>
465
466 </script>
467
468 </script>
469
470 </script>
471
472 </script>
473
474 </script>
475
476 </script>
477
478 </script>
479
480 </script>
481
482 </script>
483
484 </script>
485
486 </script>
487
488 </script>
489
490 </script>
491
492 </script>
493
494 </script>
495
496 </script>
497
498 </script>
499
500 </script>
501
502 </script>
503
504 </script>
505
506 </script>
507
508 </script>
509
510 </script>
511
512 </script>
513
514 </script>
515
516 </script>
517
518 </script>
519
520 </script>
521
522 </script>
523
524 </script>
525
526 </script>
527
528 </script>
529
530 </script>
531
532 </script>
533
534 </script>
535
536 </script>
537
538 </script>
539
540 </script>
541
542 </script>
543
544 </script>
545
546 </script>
547
548 </script>
549
550 </script>
551
552 </script>
553
554 </script>
555
556 </script>
557
558 </script>
559
560 </script>
561
562 </script>
563
564 </script>
565
566 </script>
567
568 </script>
569
570 </script>
571
572 </script>
573
574 </script>
575
576 </script>
577
578 </script>
579
580 </script>
581
582 </script>
583
584 </script>
585
586 </script>
587
588 </script>
589
590 </script>
591
592 </script>
593
594 </script>
595
596 </script>
597
598 </script>
599
600 </script>
601
602 </script>
603
604 </script>
605
606 </script>
607
608 </script>
609
610 </script>
611
612 </script>
613
614 </script>
615
616 </script>
617
618 </script>
619
620 </script>
621
622 </script>
623
624 </script>
625
626 </script>
627
628 </script>
629
630 </script>
631
632 </script>
633
634 </script>
635
636 </script>
637
638 </script>
639
640 </script>
641
642 </script>
643
644 </script>
645
646 </script>
647
648 </script>
649
650 </script>
651
652 </script>
653
654 </script>
655
656 </script>
657
658 </script>
659
660 </script>
661
662 </script>
663
664 </script>
665
666 </script>
667
668 </script>
669
670 </script>
671
672 </script>
673
674 </script>
675
676 </script>
677
678 </script>
679
680 </script>
681
682 </script>
683
684 </script>
685
686 </script>
687
688 </script>
689
690 </script>
691
692 </script>
693
694 </script>
695
696 </script>
697
698 </script>
699
700 </script>
701
702 </script>
703
704 </script>
705
706 </script>
707
708 </script>
709
710 </script>
711
712 </script>
713
714 </script>
715
716 </script>
717
718 </script>
719
720 </script>
721
722 </script>
723
724 </script>
725
726 </script>
727
728 </script>
729
730 </script>
731
732 </script>
733
734 </script>
735
736 </script>
737
738 </script>
739
740 </script>
741
742 </script>
743
744 </script>
745
746 </script>
747
748 </script>
749
750 </script>
751
752 </script>
753
754 </script>
755
756 </script>
757
758 </script>
759
760 </script>
761
762 </script>
763
764 </script>
765
766 </script>
767
768 </script>
769
770 </script>
771
772 </script>
773
774 </script>
775
776 </script>
777
778 </script>
779
780 </script>
781
782 </script>
783
784 </script>
785
786 </script>
787
788 </script>
789
790 </script>
791
792 </script>
793
794 </script>
795
796 </script>
797
798 </script>
799
800 </script>
801
802 </script>
803
804 </script>
805
806 </script>
807
808 </script>
809
810 </script>
811
812 </script>
813
814 </script>
815
816 </script>
817
818 </script>
819
820 </script>
821
822 </script>
823
824 </script>
825
826 </script>
827
828 </script>
829
830 </script>
831
832 </script>
833
834 </script>
835
836 </script>
837
838 </script>
839
840 </script>
841
842 </script>
843
844 </script>
845
846 </script>
847
848 </script>
849
850 </script>
851
852 </script>
853
854 </script>
855
856 </script>
857
858 </script>
859
860 </script>
861
862 </script>
863
864 </script>
865
866 </script>
867
868 </script>
869
870 </script>
871
872 </script>
873
874 </script>
875
876 </script>
877
878 </script>
879
880 </script>
881
882 </script>
883
884 </script>
885
886 </script>
887
888 </script>
889
890 </script>
891
892 </script>
893
894 </script>
895
896 </script>
897
898 </script>
899
900 </script>
901
902 </script>
903
904 </script>
905
906 </script>
907
908 </script>
909
910 </script>
911
912 </script>
913
914 </script>
915
916 </script>
917
918 </script>
919
920 </script>
921
922 </script>
923
924 </script>
925
926 </script>
927
928 </script>
929
930 </script>
931
932 </script>
933
934 </script>
935
936 </script>
937
938 </script>
939
940 </script>
941
942 </script>
943
944 </script>
945
946 </script>
947
948 </script>
949
950 </script>
951
952 </script>
953
954 </script>
955
956 </script>
957
958 </script>
959
960 </script>
961
962 </script>
963
964 </script>
965
966 </script>
967
968 </script>
969
970 </script>
971
972 </script>
973
974 </script>
975
976 </script>
977
978 </script>
979
980 </script>
981
982 </script>
983
984 </script>
985
986 </script>
987
988 </script>
989
990 </script>
991
992 </script>
993
994 </script>
995
996 </script>
997
998 </script>
999
1000 </script>
1001
1002 </script>
1003
1004 </script>
1005
1006 </script>
1007
1008 </script>
1009
1010 </script>
1011
1012 </script>
1013
1014 </script>
1015
1016 </script>
1017
1018 </script>
1019
1020 </script>
1021
1022 </script>
1023
1024 </script>
1025
1026 </script>
1027
1028 </script>
1029
1030 </script>
1031
1032 </script>
1033
1034 </script>
1035
1036 </script>
1037
1038 </script>
1039
1040 </script>
1041
1042 </script>
1043
1044 </script>
1045
1046 </script>
1047
1048 </script>
1049
1050 </script>
1051
1052 </script>
1053
1054 </script>
1055
1056 </script>
1057
1058 </script>
1059
1060 </script>
1061
1062 </script>
1063
1064 </script>
1065
1066 </script>
1067
1068 </script>
1069
1070 </script>
1071
1072 </script>
1073
1074 </script>
1075
1076 </script>
1077
1078 </script>
1079
1080 </script>
1081
1082 </script>
1083
1084 </script>
1085
1086 </script>
1087
1088 </script>
1089
1090 </script>
1091
1092 </script>
1093
1094 </script>
1095
1096 </script>
1097
1098 </script>
1099
1100 </script>
1101
1102 </script>
1103
1104 </script>
1105
1106 </script>
1107
1108 </script>
1109
1110 </script>
1111
1112 </script>
1113
1114 </script>
1115
1116 </script>
1117
1118 </script>
1119
1120 </script>
1121
1122 </script>
1123
1124 </script>
1125
1126 </script>
1127
1128 </script>
1129
1130 </script>
1131
1132 </script>
1133
1134 </script>
1135
1136 </script>
1137
1138 </script>
1139
1140 </script>
1141
1142 </script>
1143
1144 </script>
1145
1146 </script>
1147
1148 </script>
1149
1150 </script>
1151
1152 </script>
1153
1154 </script>
1155
1156 </script>
1157
1158 </script>
1159
1160 </script>
1161
1162 </script>
1163
1164 </script>
1165
1166 </script>
1167
1168 </script>
1169
1170 </script>
1171
1172 </script>
1173
1174 </script>
1175
1176 </script>
1177
1178 </script>
1179
1180 </script>
1181
1182 </script>
1183
1184 </script>
1185
1186 </script>
1187
1188 </script>
1189
1190 </script>
1191
1192 </script>
1193
1194 </script>
1195
1196 </script>
1197
1198 </script>
1199
1200 </script>
1201
1202 </script>
1203
1204 </script>
1205
1206 </script>
1207
1208 </script>
1209
1210 </script>
1211
1212 </script>
1213
1214 </script>
1215
1216 </script>
1217
1218 </script>
1219
1220 </script>
1221
1222 </script>
1223
1224 </script>
1225
1226 </script>
1227
1228 </script>
1229
1230 </script>
1231
1232 </script>
1233
1234 </script>
1235
1236 </script>
1237
1238 </script>
1239
1240 </script>
1241
1242 </script>
1243
1244 </script>
1245
1246 </script>
1247
1248 </script>
1249
1250 </script>
1251
1252 </script>
1253
1254 </script>
1255
1256 </script>
1257
1258 </script>
1259
1260 </script>
1261
1262 </script>
1263
1264 </script>
1265
1266 </script>
1267
1268 </script>
1269
1270 </script>
1271
1272 </script>
1273
1274 </script>
1275
1276 </script>
1277
1278 </script>
1279
1280 </script>
1281
1282 </script>
1283
1284 </script>
1285
1286 </script>
1287
1288 </script>
1289
1290 </script>
1291
1292 </script>
1293
1294 </script>
1295
1296 </script>
1297
1298 </script>
1299
1300 </script>
1301
1302 </script>
1303
1304 </script>
1305
1306 </script>
1307
1308 </script>
1309
1310 </script>
1311
1312 </script>
1313
1314 </script>
1315
1316 </script>
1317
1318 </script>
1319
1320 </script>
1321
1322 </script>
1323
1324 </script>
1325
1326 </script>
1327
1328 </script>
1329
1330 </script>
1331
1332 </script>
1333
1334 </script>
1335
1336 </script>
1337
1338 </script>
1339
1340 </script>
1341
1342 </script>
1343
1344 </script>
1345
1346 </script>
1347
1348 </script>
1349
1350 </script>
1351
1352 </script>
1353
1354 </script>
1355
1356 </script>
1357
1358 </script>
1359
1360 </script>
1361
1362 </script>
1363
1364 </script>
1365
1366 </script>
1367
1368 </script>
1369
1370 </script>
1371
1372 </script>
1373
1374 </script>
1375
1376 </script>
1377
1378 </script>
1379
1380 </script>
1381
1382 </script>
1383
1384 </script>
1385
1386 </script>
1387
1388 </script>
1389
1390 </script>
1391
1392 </script>
1393
1394 </script>
1395
1396 </script>
1397
1398 </script>
1399
1400 </script>
1401
1402 </script>
1403
1404 </script>
1405
1406 </script>
1407
1408 </script>
1409
1410 </script>
1411
1412 </script>
1413
1414 </script>
1415
1416 </script>
1417
1418 </script>
1419
1420 </script>
1421
1422 </script>
1423
1424 </script>
1425
1426 </script>
1427
1428 </script>
1429
1430 </script>
1431
1432 </script>
1433
1434 </script>
1435
1436 </script>
1437
1438 </script>
1439
1440 </script>
1441
1442 </script>
1443
1444 </script>
1445
1446 </script>
1447
1448 </script>
1449
1450 </script>
1451
1452 </script>
1453
1454 </script>
1455
1456 </script>
1457
1458 </script>
1459
1460 </script>
1461
1462 </script>
1463
1464 </script>
1465
1466 </script>
1467
1468 </script>
1469
1470 </script>
1471
1472 </script>
1473
1474 </script>
1475
1476 </script>
1477
1478 </script>
1479
1480 </script>
1481
1482 </script>
1483
1484 </script>
1485
1486 </script>
1487
1488 </script>
1489
1490 </script>
1491
1492 </script>
1493
1494 </script>
1495
1496 </script>
1497
1498 </script>
1499
1500 </script>
1501
1502 </script>
1503
1504 </script>
1505
1506 </script>
1507
1508 </script>
1509
1510 </script>
1511
1512 </script>
1513
1514 </script>
1515
1516 </script>
1517
1518 </script>
1519
1520 </script>
1521
1522 </script>
1523
1524 </script>
1525
1526 </script>
1527
1528 </script>
1529
1530 </script>
1531
1532 </script>
1533
1534 </script>
1535
1536 </script>
1537
1538 </script>
1539
1540 </script>
1541
1542 </script>
1543
1544 </script>
1545
1546 </script>
1547
1548 </script>
1549
1550 </script>
1551
1552 </script>
```

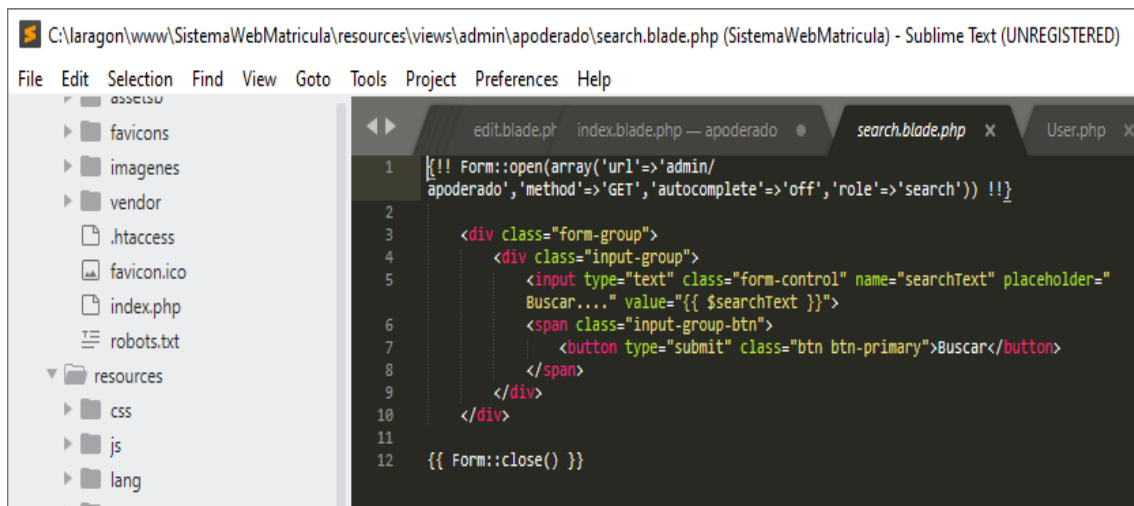

Figura 86: Vistas de Registrar Apoderado (*modal.blade.php*).



```
1 <!-- Modal -->
2 <div class="modal fade" id="modal-delete-{{ $cat->id }}" tabindex="-1" aria-labelledby="
exampleModallabel" aria-hidden="true">
3 <div class="modal-dialog">
4 <form action="{{ route('apoderado.destroy',$cat->id) }}" method="post" c>
5 <input type="hidden">
6 @csrf
7 @method('DELETE')
8 <div class="modal-content">
9 <div class="modal-header">
10 <h1 class="modal-title fs-5" id="exampleModallabel">Eliminar Apoderado</h1>
11 <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"
>
12 <span aria-hidden="true">&times;</span>
13 </button>
14 </div>
15 <div class="modal-body">
16 Confirme si desea eliminar el Apoderado {{ $cat->nombre." ".$cat->apellido }}
17 </div>
18 <div class="modal-footer">
19 <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</
button>
20 <input type="submit" class="btn btn-danger btn-sm" value="Eliminar">
21 </div>
22 </div>
23 </form>
24 </div>
25 </div>
26
27
```

Nota. Código PHP de vista Registrar Apoderado (*modal.blade.php*).

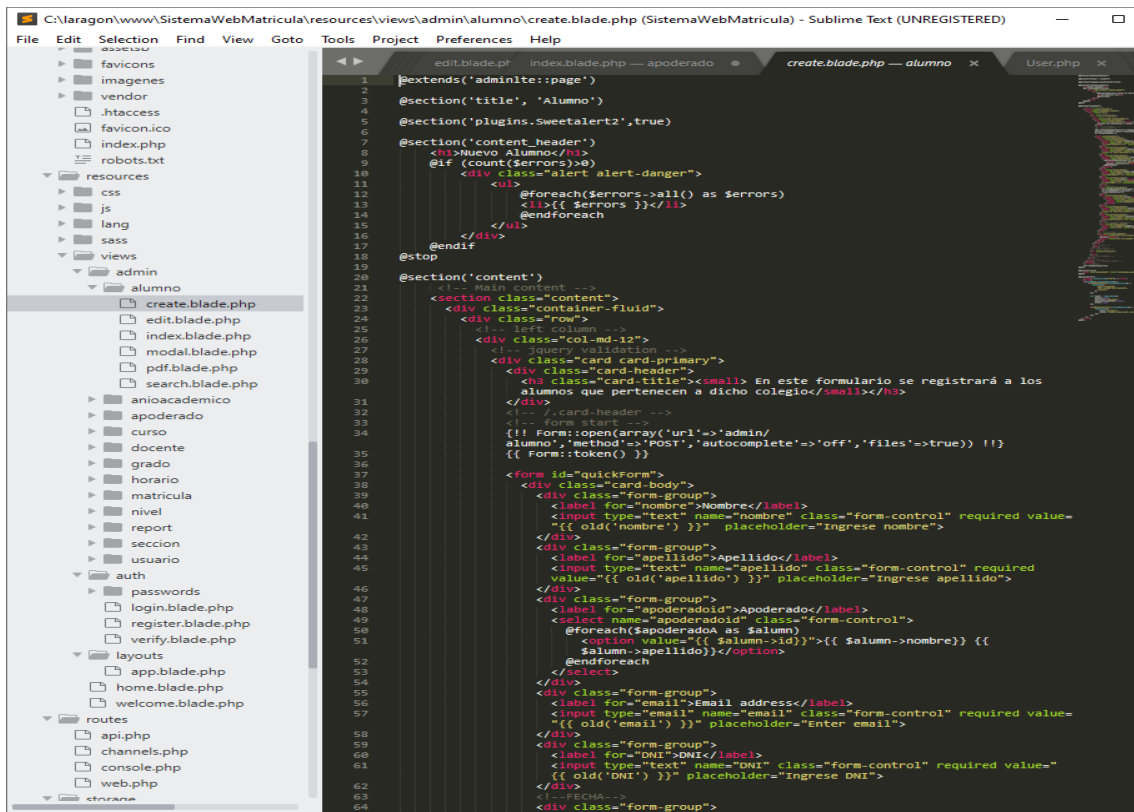
Figura 87: Vistas de Registrar Apoderado (*search.blade.php*)



```
1 {!! Form::open(array('url'=>'admin/
apoderado', 'method'=>'GET', 'autocomplete'=>'off', 'role'=>'search')) !!}
2
3 <div class="form-group">
4 <div class="input-group">
5 <input type="text" class="form-control" name="searchText" placeholder="
Buscar..." value="{{ $searchText }}">
6 <span class="input-group-btn">
7 <button type="submit" class="btn btn-primary">Buscar</button>
8 </span>
9 </div>
10 </div>
11
12 {!! Form::close() !!}
```

Nota. Código PHP de vista Registrar Apoderado (*search.blade.php*).

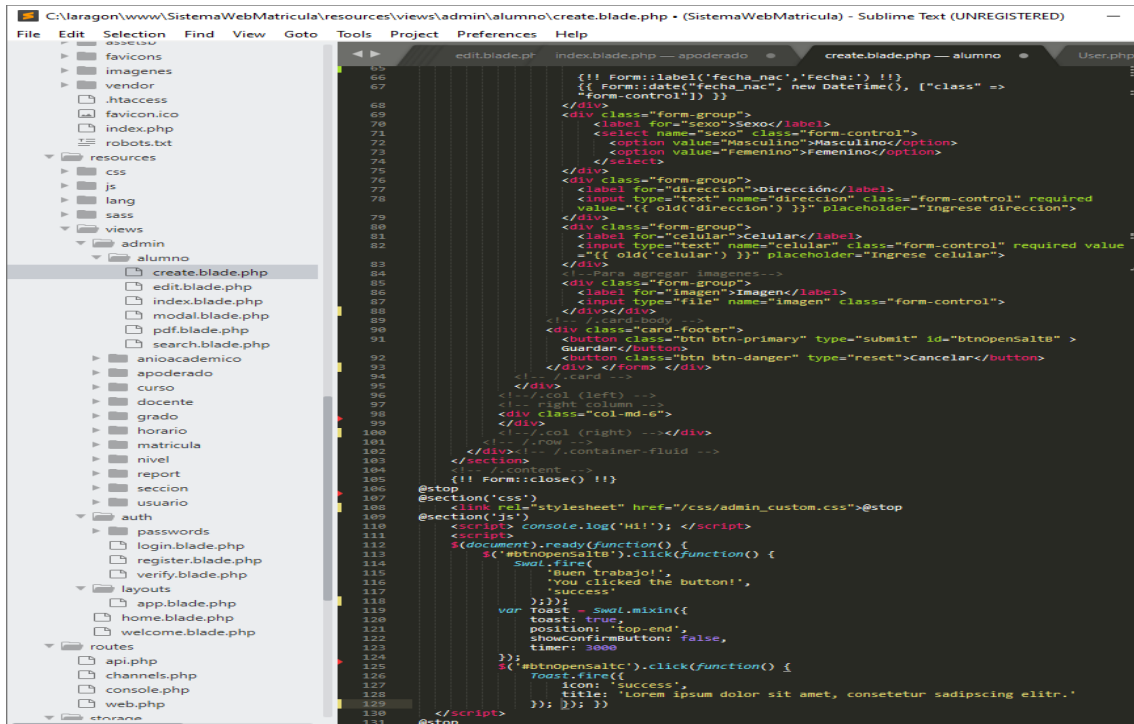
Figura 88: Vistas de Registrar Alumnos (create.blade.php).



```
1 @extends('adminlte::page')
2 @section('title', 'Alumno')
3 @section('plugins.Sweetalert2', true)
4 @section('content_header')
5 <!-- Nuevo Alumno/13 -->
6 @if (count($errors)>0)
7 <div class="alert alert-danger">
8 <ul>
9 @foreach($errors->all() as $errors)
10 <li>{{ $errors }}</li>
11 @endforeach
12 </ul>
13 @endif
14 @stop
15 @section('content')
16 <!-- Main content -->
17 <section class="content">
18 <div class="container-fluid">
19 <div class="row">
20 <div class="col-md-12">
21 <div class="card card-primary">
22 <div class="card-header">
23 <h3 class="card-title"><small> En este formulario se registrará a los
24 alumnos que pertenecen a dicho colegio.</small></h3>
25 </div>
26 <!-- Form start -->
27 {!! Form::open(array('url'=>'admin/
28 alumno', 'method'=>'POST', 'autocomplete'=>'off', 'files'=>true)) !!}
29 {!! Form::token() !!}
30 <form id="quickForm">
31 <div class="card-body">
32 <div class="form-group">
33 <label for="nombre">Nombre</label>
34 <input type="text" name="nombre" class="form-control" required value=
35 "{{ old('nombre') }}" placeholder="Ingrese nombre">
36 </div>
37 <div class="form-group">
38 <label for="apellido">Apellido</label>
39 <input type="text" name="apellido" class="form-control" required
40 value="{{ old('apellido') }}" placeholder="Ingrese apellido">
41 </div>
42 <div class="form-group">
43 <label for="apoderadoid">Aporadado</label>
44 <select name="apoderadoid" class="form-control" required
45 @foreach($apoderados as $alumn)
46 <option value="{{ $alumn->id }}">{{ $alumn->nombre }} {{
47 $alumn->apellido}}</option>
48 @endforeach
49 </select>
50 </div>
51 <div class="form-group">
52 <label for="email">Email address</label>
53 <input type="text" name="email" class="form-control" required value=
54 "{{ old('email') }}" placeholder="Enter email">
55 </div>
56 <div class="form-group">
57 <label for="DNI">DNI</label>
58 <input type="text" name="DNI" class="form-control" required value=
59 "{{ old('DNI') }}" placeholder="Ingrese DNI">
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
122 </div>
123 </div>
124 </div>
125 </div>
126 </div>
127 </div>
128 </div>
129 </div>
130 </div>
131 @stop
```

Nota. Código PHP de vista Registrar Alumnos (create.blade.php).

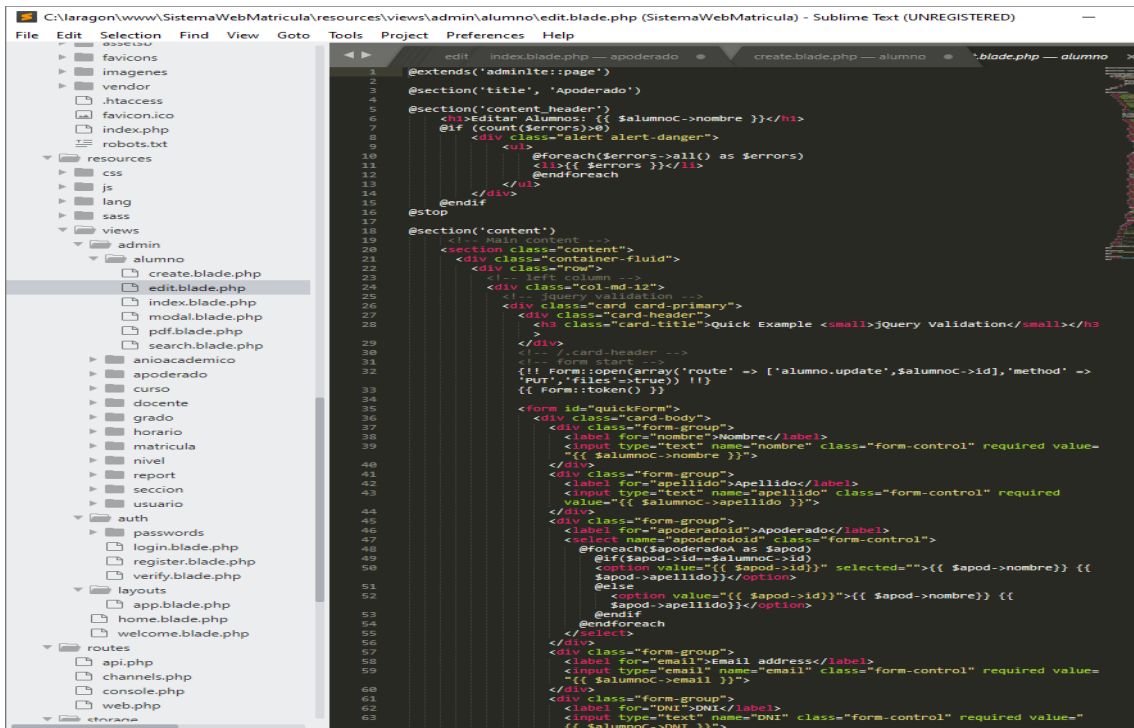
Figura 89: Vistas de Registrar Alumnos (create.blade.php)



```
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
122 </div>
123 </div>
124 </div>
125 </div>
126 </div>
127 </div>
128 </div>
129 </div>
130 </div>
131 @stop
```

Nota. Código PHP de vista Registrar Alumnos (create.blade.php).

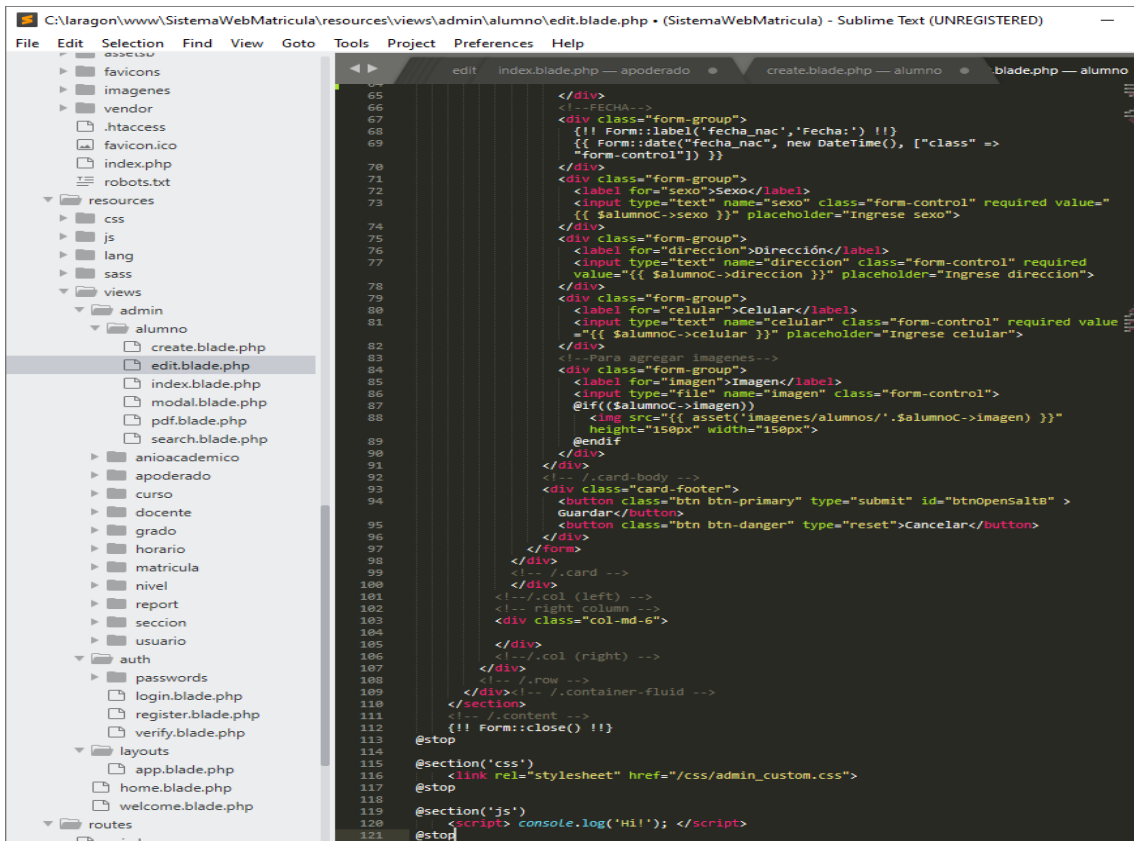
Figura 90: Vistas de Registrar Alumnos (edit.blade.php).



```
1 @extends('adminlte:page')
2
3 @section('title', 'Apoderado')
4
5 @section('content_header')
6 <h1>Editar Alumnos: {{ $alumno->nombre }}</h1>
7
8 @if (count($errors)>0)
9 <div class="alert alert-danger">
10 <ul>
11 @foreach($errors->all() as $errors)
12 <li>{{ $errors }}</li>
13 @endforeach
14 </ul>
15 </div>
16 @endif
17
18 @stop
19
20 @section('content')
21 <!-- Main content -->
22 <section class="content">
23 <div class="container-fluid">
24 <div class="row">
25 <!-- left column -->
26 <div class="col-md-4">
27 <div class="card card-primary">
28 <div class="card-header">
29 <h3 class="card-title">Quick Example <small>jQuery Validation</small></h3>
30 </div>
31 <!-- form start -->
32 {!! Form::open(array('route' => ['alumno.update',$alumno->id],'method' =>
33 'PUT','files'=>true)) !!}
34 {!! Form::token() !!}
35 <form id="quickForm">
36 <div class="card-body">
37 <div class="form-group">
38 <label for="nombre">Nombre</label>
39 <input type="text" name="nombre" class="form-control" required value=
40 "{{ $alumno->nombre }}">
41 </div>
42 <div class="form-group">
43 <label for="apellido">Apellido</label>
44 <input type="text" name="apellido" class="form-control" required
45 value="{{ $alumno->apellido }}">
46 </div>
47 <div class="form-group">
48 <label for="apoderadoId">Apoderado</label>
49 @foreach($apoderado as $apod)
50 <input type="radio" name="apoderado" value="{{ $apod->id }}" selected="">{{ $apod->nombre }} {{
51 $apod->apellido}}</input>
52 @else
53 <input type="radio" name="apoderado" value="">{{ $apod->nombre }} {{
54 $apod->apellido}}</input>
55 @endif
56 </div>
57 <div class="form-group">
58 <input type="email" name="email" class="form-control" required value=
59 "{{ $alumno->email }}">
60 </div>
61 <div class="form-group">
62 <label for="DNI">DNI</label>
63 <input type="text" name="DNI" class="form-control" required value=
64 "{{ $alumno->DNI }}">
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
```

Nota. Código PHP de vista Registrar Alumnos (edit.blade.php).

Figura 91: Vistas de Registrar Alumnos (edit.blade.php)



```
64 </div>
65 </div>
66 </div>
67 <div class="form-group">
68 {!! Form::label('fecha_nac','Fecha:') !!}
69 {!! Form::date("fecha_nac", new DateTime(), ["class" =>
70 "form-control"]) !!}
71 </div>
72 <div class="form-group">
73 <label for="sexo">Sexo</label>
74 <input type="text" name="sexo" class="form-control" required value=
75 "{{ $alumno->sexo }}" placeholder="Ingrese sexo">
76 </div>
77 <div class="form-group">
78 <label for="direccion">Dirección</label>
79 <input type="text" name="direccion" class="form-control" required
80 value="{{ $alumno->direccion }}" placeholder="Ingrese direccion">
81 </div>
82 <div class="form-group">
83 <label for="celular">Celular</label>
84 <input type="text" name="celular" class="form-control" required value=
85 "{{ $alumno->celular }}" placeholder="Ingrese celular">
86 </div>
87 <!-- Para agregar imagenes -->
88 <div class="form-group">
89 <label for="imagen">Imagen</label>
90 <input type="file" name="imagen" class="form-control">
91 @if($alumno->imagen)
92 
94 @endif
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>
120 </div>
121 </div>
```

Nota. Código PHP de vista Registrar Alumnos (edit.blade.php).

Figura 92: Vistas de Registrar Alumnos (index.blade.php).

```

1 @extends('adminite::page')
2
3 @section('title', 'Alumnos')
4
5 @section('content_header')
6 <h3>Estado de alumnos <!--<a href="apoderado/create"><button class="btn
7 btn-success">Nuevo</button></a--></h3>
8
9 @stop
10
11 @section('content')
12 <div class="card">
13 <div class="card-header">
14 <a href="alumno/create" class="btn btn-primary">Nuevo Alumno</a>
15 <div style="text-align:right">
16 <a href="{{ route('alumnos.pdf') }}" class="btn btn-secondary">Generar
17 PDF</a>
18 </div>
19 </div>
20 </div>
21
22 <div class="card-body">
23 <table id="example" style="width: 100%;" align="center">
24 <thead class="bg-primary text-white">
25 <tr>
26 <th>Nombres</th>
27 <th>Email</th>
28 <th>DNI</th>
29 <th>Fecha Nacimiento</th>
30 <th>Sexo</th>
31 <th>Dirección</th>
32 <th>Imagen</th>
33 <th>Apoderado</th>
34 <th>Opciones</th>
35 </tr>
36 </thead>
37 <tbody>
38 @foreach($alumnoA as $al)
39 <tr align="center">
40 <td>{{ $al->nombre }} {{ $al->apellido }}</td>
41 <td>{{ $al->email }}</td>
42 <td>{{ $al->DNI }}</td>
43 <td>{{ $al->fecha_nac }}</td>
44 <td>{{ $al->sexo }}</td>
45 <td>{{ $al->direccion }}</td>
46 <td>{{ $al->celular }}</td>
47 <td>
48 nombre }}" height="100px" width="100px" class="
50 img-thumbnail">
51 <a target="blank" href="/imagenes/alumnos/{{ $al->imagen }}"
52 > Abrir Foto</a>
53 </td>
54 <td>{{ $al->apoderado }}</td>
55 <td>
56 <a href="{{ route('alumno.edit', $al->id) }}"><button class="
57 btn btn-primary btn-sm">Editar</button></a>
58 <!-- Button trigger modal -->
59 <button type="button" class="btn btn-danger btn-sm"
60 data-bs-toggle="modal" data-bs-target="#modal-{{ $al->id }}">
61 Eliminar
62 </button>
63 </td>
64 </tr>
65 @include('admin.alumno.modal')
66 @endforeach
67 </tbody>
68 </table>
69 </div>
70 </div>

```

Nota. Código PHP de vista Registrar Alumnos (index.blade.php).

Figura 93: Vistas de Registrar Alumnos (index.blade.php).

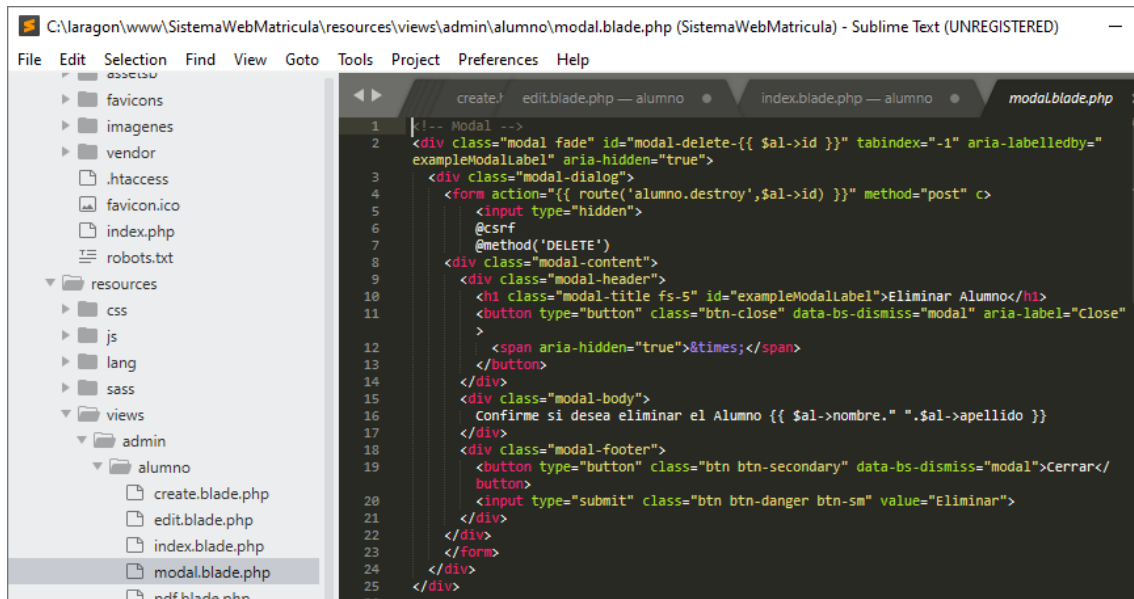
```

65 @stop
66
67 @section('css')
68 <link rel="stylesheet" href="/css/admin_custom.css">
69 <link rel="stylesheet" href="https://cdn.datatables.net/1.13.1/css/
70 dataTables.bootstrap5.min.css">
71 @stop
72
73 @section('js')
74 <script console.log('Hi!'); </script>
75 <script src="https://code.jquery.com/jquery-3.5.1.js"></script>
76 <script src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>
77 <script src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap5.min.js"></
78 script>
79 <script>
80 $(document).ready(function () {
81   $('#example').DataTable({
82     "language":{
83       "search": "Buscar",
84       "lengthMenu": "Mostrar_MENU registros por página",
85       "info": "Mostrando página_PAGE de_PAGES_",
86       "paginate":{
87         "previous": "Anterior",
88         "next": "siguiente",
89         "first": "Primero",
90         "last": "Último"
91       }
92     }
93   });
94 </script>
95
96 <script>
97 /*Swal.fire(
98   'Buen trabajo!',
99   'You clicked the button!',
100   'success'
101 );*/
102 </script>
103 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/
104 popper.min.js" integrity="sha384-08qvmz9ATKXieP9tCX5/
105 29Hf6XIDAYTuzJueBAsiFjCzsmw0BSSUNQlmyJp" crossorigin="anonymous"></script>
106 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
107 integrity="sha384-IDwe1+LCz02R09k972gdyv1+AESN10+x7T8kgC9I5HFtun0WnPCl2069vxnk"
108 crossorigin="anonymous"></script>
109 @stop

```

Nota. Código PHP de vista Registrar Alumnos (index.blade.php).

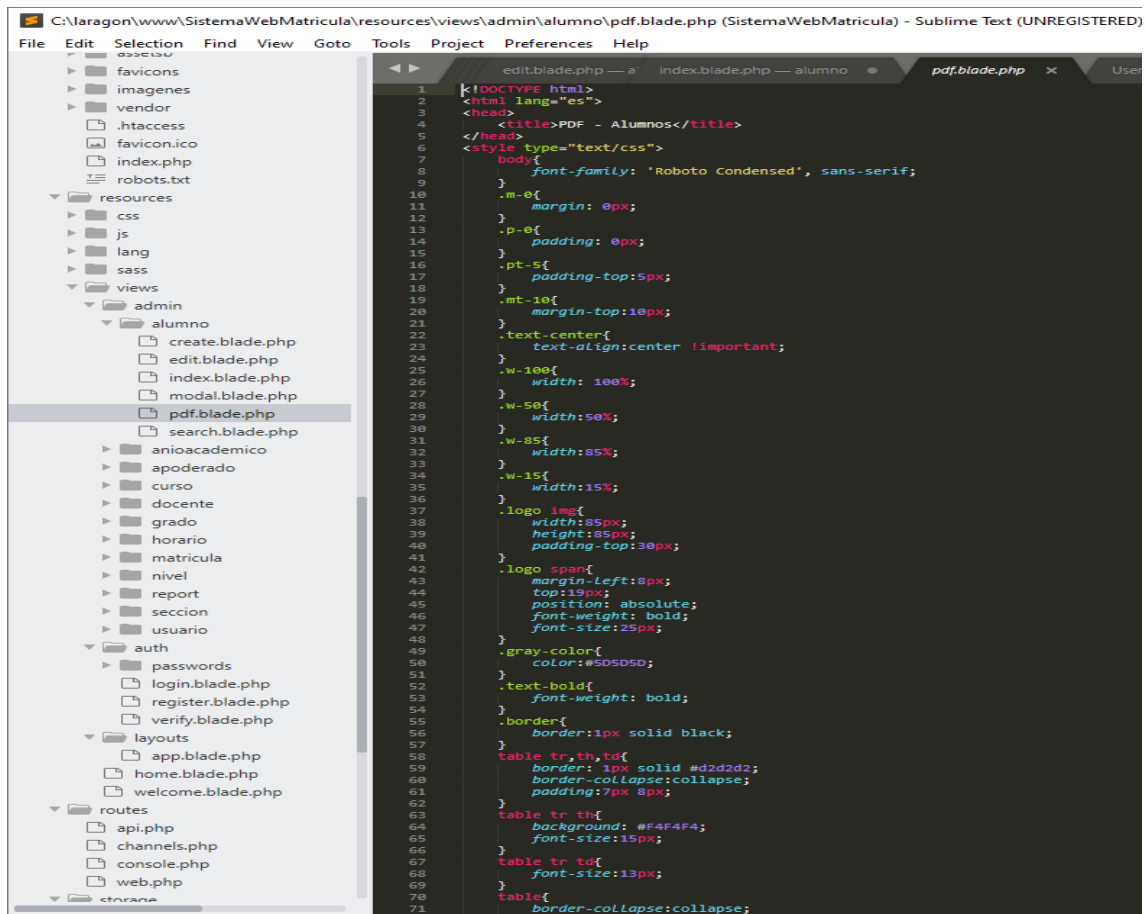
Figura 94: Vistas de Registrar Alumnos (modal.blade.php).



```
1 <!-- Modal -->
2 <div class="modal fade" id="modal-delete-{{ $al->id }}" tabindex="-1" aria-labelledby="
3 exampleModallabel" aria-hidden="true">
4 <div class="modal-dialog">
5 <form action="{{ route('alumno.destroy',$al->id) }}" method="post" c>
6 <input type="hidden">
7 @csrf
8 @method('DELETE')
9 <div class="modal-content">
10 <div class="modal-header">
11 <h1 class="modal-title fs-5" id="exampleModallabel">Eliminar Alumno</h1>
12 <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="close"
13 >
14 <span aria-hidden="true">&times;</span>
15 </button>
16 </div>
17 <div class="modal-body">
18 Confirme si desea eliminar el Alumno {{ $al->nombre." ".$al->apellido }}
19 </div>
20 <div class="modal-footer">
21 <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</
22 button>
23 <input type="submit" class="btn btn-danger btn-sm" value="Eliminar">
24 </div>
25 </div>
26 </div>
```

Nota. Código PHP de vista Registrar Alumnos (modal.blade.php).

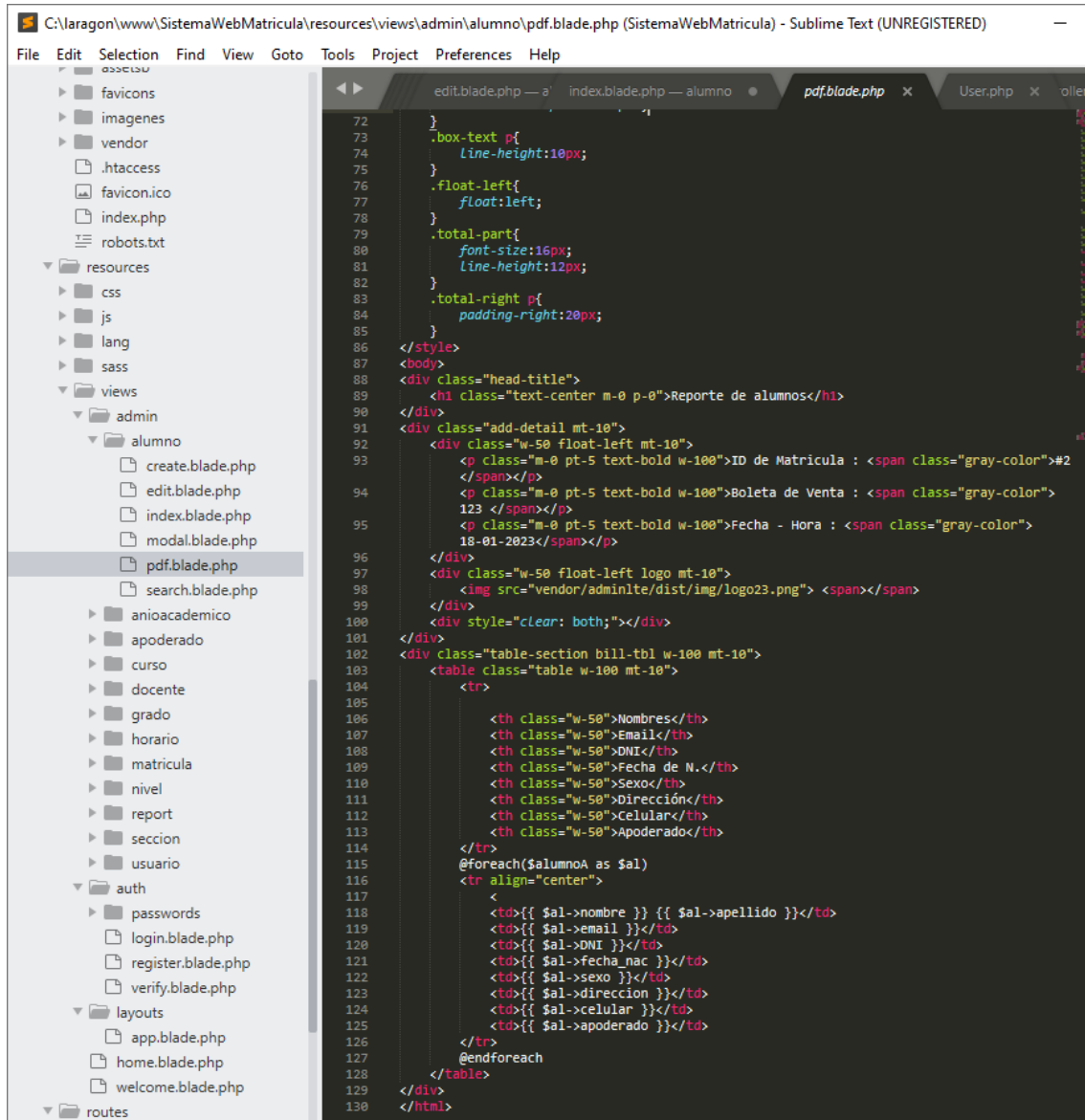
Figura 95: Vistas de Reporte de Alumnos (pdf.blade.php).



```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <title>PDF - Alumnos</title>
5 </head>
6 <style type="text/css">
7 body{
8 font-family: 'Roboto Condensed', sans-serif;
9 }
10 .m-0{
11 margin: 0px;
12 }
13 .p-0{
14 padding: 0px;
15 }
16 .pt-5{
17 padding-top:5px;
18 }
19 .mt-10{
20 margin-top:10px;
21 }
22 .text-center{
23 text-align:center !important;
24 }
25 .w-100{
26 width: 100%;
27 }
28 .w-50{
29 width:50%;
30 }
31 .w-85{
32 width:85%;
33 }
34 .w-15{
35 width:15%;
36 }
37 .logo img{
38 width:85px;
39 height:85px;
40 padding-top:30px;
41 }
42 .logo span{
43 margin-left:8px;
44 top:19px;
45 position: absolute;
46 font-weight: bold;
47 font-size:25px;
48 }
49 .gray-color{
50 color:#5D5D5D;
51 }
52 .text-bold{
53 font-weight: bold;
54 }
55 .border{
56 border:1px solid black;
57 }
58 table tr,th,td{
59 border: 1px solid #d2d2d2;
60 border-collapse:collapse;
61 padding:7px 8px;
62 }
63 table tr th{
64 background: #F4F4F4;
65 font-size:15px;
66 }
67 table tr td{
68 font-size:13px;
69 }
70 table{
71 border-collapse:collapse;
```

Nota. Código PHP de vista Reporte de Alumnos (pdf.blade.php).

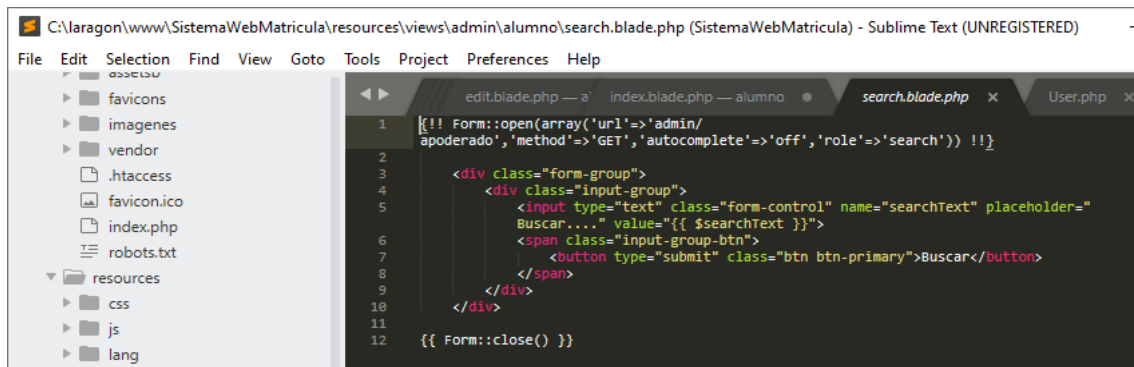
Figura 96: Vistas de Reporte de Alumnos (*pdf.blade.php*).



```
72     }
73     .box-text p{
74         Line-height:10px;
75     }
76     .float-left{
77         float:left;
78     }
79     .total-part{
80         font-size:16px;
81         Line-height:12px;
82     }
83     .total-right p{
84         padding-right:20px;
85     }
86 }
87 </style>
88 <body>
89 <div class="head-title">
90 <h1 class="text-center m-0 p-0">Reporte de alumnos</h1>
91 </div>
92 <div class="add-detail mt-10">
93 <div class="w-50 float-left mt-10">
94 <p class="m-0 pt-5 text-bold w-100">ID de Matricula : <span class="gray-color">#2
95 </span></p>
96 <p class="m-0 pt-5 text-bold w-100">Boleta de Venta : <span class="gray-color">
97 123 </span></p>
98 <p class="m-0 pt-5 text-bold w-100">Fecha - Hora : <span class="gray-color">
99 18-01-2023</span></p>
100 </div>
101 <div class="w-50 float-left logo mt-10">
102  <span></span>
103 </div>
104 <div style="clear: both;"></div>
105 </div>
106 <div class="table-section bill-tbl w-100 mt-10">
107 <table class="table w-100 mt-10">
108 <thead>
109 <tr>
110 <th class="w-50">Nombres</th>
111 <th class="w-50">Email</th>
112 <th class="w-50">DNI</th>
113 <th class="w-50">Fecha de N.</th>
114 <th class="w-50">Sexo</th>
115 <th class="w-50">Dirección</th>
116 <th class="w-50">Celular</th>
117 <th class="w-50">Apoderado</th>
118 </tr>
119 </thead>
120 <tbody>
121 @foreach($alumnoA as $al)
122 <tr align="center">
123 <td>{{ $al->nombre }} {{ $al->apellido }}</td>
124 <td>{{ $al->email }}</td>
125 <td>{{ $al->DNI }}</td>
126 <td>{{ $al->fecha nac }}</td>
127 <td>{{ $al->sexo }}</td>
128 <td>{{ $al->direccion }}</td>
129 <td>{{ $al->celular }}</td>
130 <td>{{ $al->apoderado }}</td>
131 </tr>
132 @endforeach
133 </tbody>
134 </table>
135 </div>
136 </html>
```

Nota. Código PHP de vista Reporte Alumnos (*pdf.blade.php*).

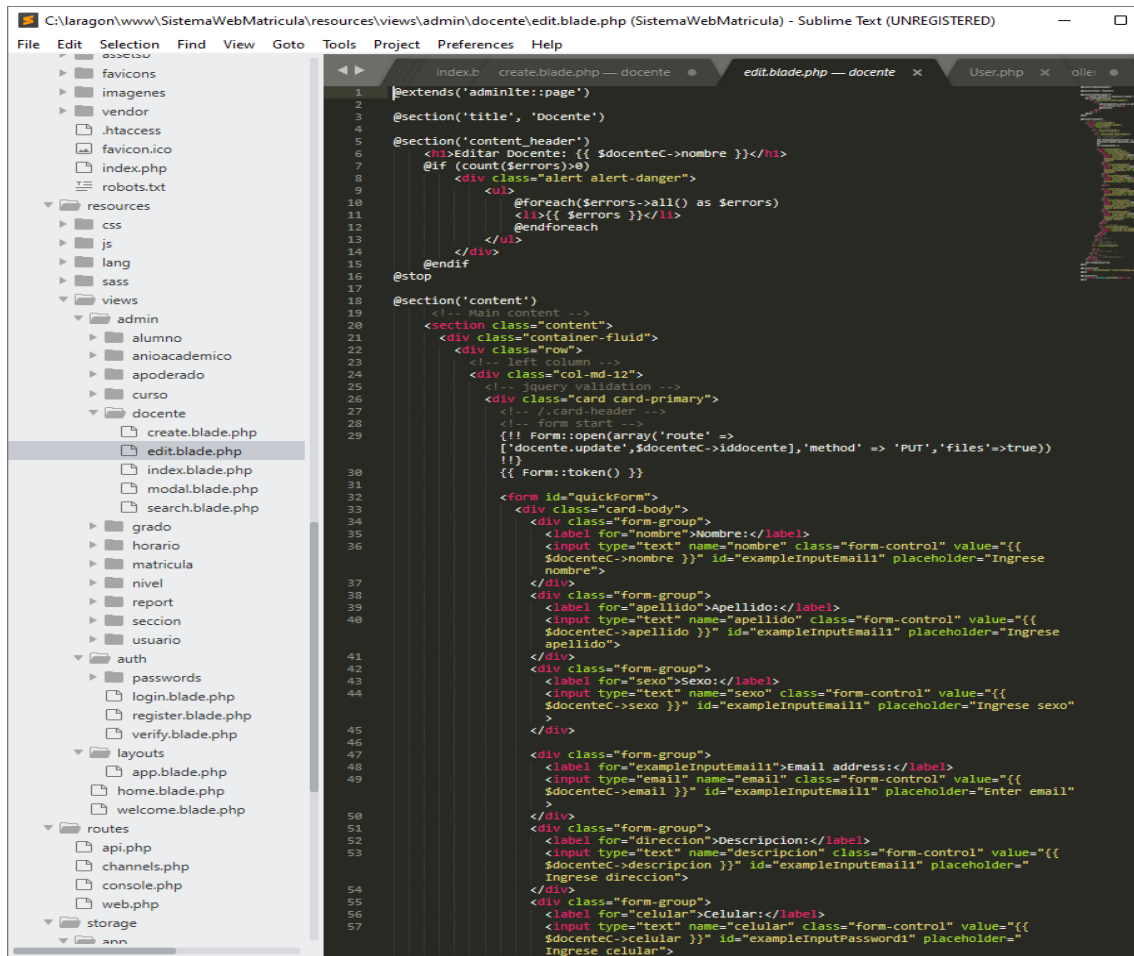
Figura 97: Vistas de Registrar Alumnos (*search.blade.php*).



```
1 <{!! Form::open(array('url'=>'admin/
2 apoderado','method'=>'GET','autocomplete'=>'off','role'=>'search')) !!>
3
4 <div class="form-group">
5 <div class="input-group">
6 <input type="text" class="form-control" name="searchText" placeholder="
7 Buscar..." value="{{ $searchText }}">
8 <span class="input-group-btn">
9 <button type="submit" class="btn btn-primary">Buscar</button>
10 </span>
11 </div>
12 </div>
13
14 {!! Form::close() !!>
```

Nota. Código PHP de vista Registrar de Alumnos (*search.blade.php*).

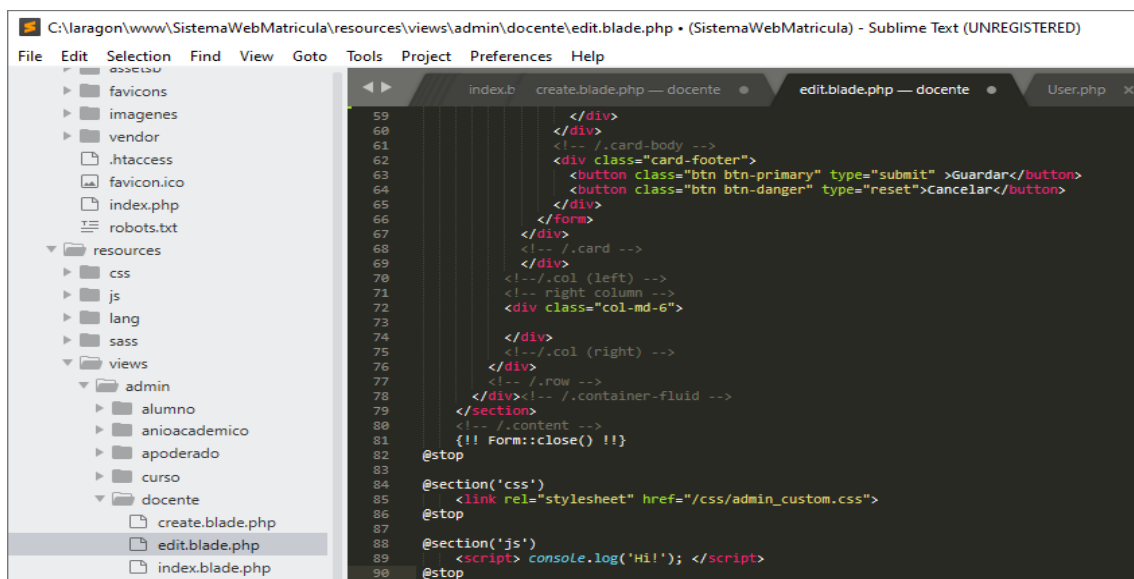
Figura 100: Vistas de Registrar Docentes (edit.blade.php).



```
1 @extends('adminlte::page')
2
3 @section('title', 'Docente')
4
5 @section('content_header')
6 <h1>Editar Docente: {{ $docentec->nombre }}</h1>
7 @if (count($errors)>0)
8 <div class="alert alert-danger">
9 <ul>
10 @foreach($errors->all() as $errors)
11 <li>{{ $errors }}</li>
12 @endforeach
13 </ul>
14 </div>
15 @endif
16 @stop
17
18 @section('content')
19 <!-- Main content -->
20 <section class="content">
21 <div class="container-fluid">
22 <div class="row">
23 <!-- left column -->
24 <div class="col-md-12">
25 <!-- jquery validation -->
26 <div class="card card-primary">
27 <!-- /.card-header -->
28 <!-- form start -->
29 {!! Form::open(array('route' =>
30 ['docente.update', $docentec->iddocente], 'method' => 'PUT', 'files'=>true))
31 !!)
32 {!! Form::token() !!}
33
34 <form id="quickForm">
35 <div class="card-body">
36 <div class="form-group">
37 <label for="nombre">Nombre:</label>
38 <input type="text" name="nombre" class="form-control" value="{{
39 $docentec->nombre }}" id="exampleInputEmail1" placeholder="Ingrese
40 nombre">
41 </div>
42 <div class="form-group">
43 <label for="apellido">Apellido:</label>
44 <input type="text" name="apellido" class="form-control" value="{{
45 $docentec->apellido }}" id="exampleInputEmail1" placeholder="Ingrese
46 apellido">
47 </div>
48 <div class="form-group">
49 <label for="sexo">Sexo:</label>
50 <input type="text" name="sexo" class="form-control" value="{{
51 $docentec->sexo }}" id="exampleInputEmail1" placeholder="Ingrese sexo"
52 >
53 </div>
54 <div class="form-group">
55 <label for="exampleInputEmail1">Email address:</label>
56 <input type="email" name="email" class="form-control" value="{{
57 $docentec->email }}" id="exampleInputEmail1" placeholder="Enter email"
58 >
59 </div>
60 <div class="form-group">
61 <label for="direccion">Descripcion:</label>
62 <input type="text" name="descripcion" class="form-control" value="{{
63 $docentec->descripcion }}" id="exampleInputEmail1" placeholder="Ingrese
64 direccion">
65 </div>
66 <div class="form-group">
67 <label for="celular">Celular:</label>
68 <input type="text" name="celular" class="form-control" value="{{
69 $docentec->celular }}" id="exampleInputPassword1" placeholder="Ingrese
70 celular">
71 </div>
72 </div>
73 </div>
74 <div class="card-footer">
75 <button class="btn btn-primary" type="submit">Guardar</button>
76 <button class="btn btn-danger" type="reset">Cancelar</button>
77 </div>
78 </form>
79 </div>
80 <!-- /.card -->
81 </div>
82 <!-- /.col (left) -->
83 <!-- right column -->
84 <div class="col-md-6">
85 </div>
86 </div>
87 <!-- /.col (right) -->
88 </div>
89 </div>
90 <!-- /.row -->
91 </div>
92 <!-- /.container-fluid -->
93 </section>
94 <!-- /.content -->
95 {!! Form::close() !!}
96 @stop
97
98 @section('css')
99 <link rel="stylesheet" href="/css/admin_custom.css">
100 @stop
101
102 @section('js')
103 <script> console.log('Hi!'); </script>
104 @stop
```

Nota. Código PHP de vista Registrar Docentes (edit.blade.php).

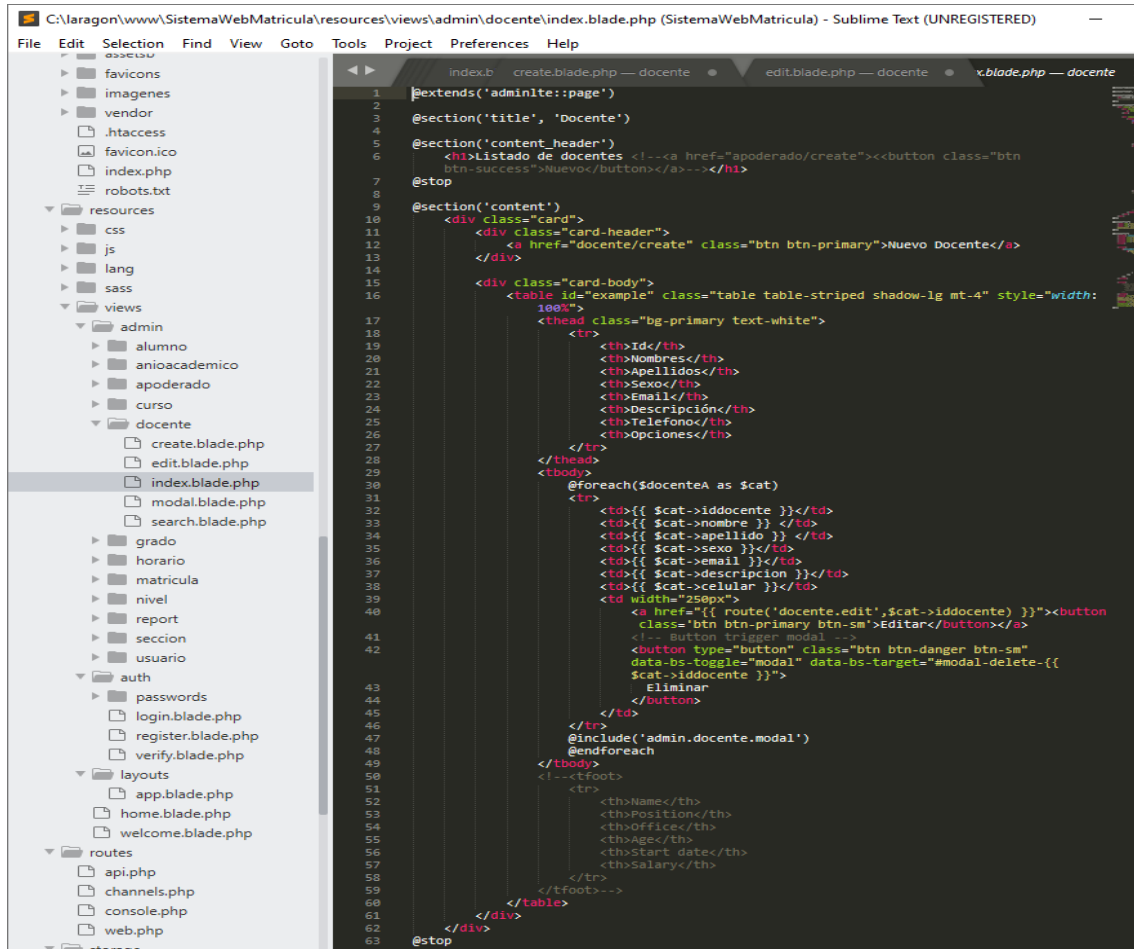
Figura 101: Vistas de Registrar Docentes (edit.blade.php).



```
59 </div>
60 </div>
61 <!-- /.card-body -->
62 <div class="card-footer">
63 <button class="btn btn-primary" type="submit">Guardar</button>
64 <button class="btn btn-danger" type="reset">Cancelar</button>
65 </div>
66 </form>
67 </div>
68 <!-- /.card -->
69 </div>
70 <!-- /.col (left) -->
71 <!-- right column -->
72 <div class="col-md-6">
73 </div>
74 </div>
75 <!-- /.col (right) -->
76 </div>
77 <!-- /.row -->
78 </div>
79 <!-- /.container-fluid -->
80 </section>
81 <!-- /.content -->
82 {!! Form::close() !!}
83 @stop
84
85 @section('css')
86 <link rel="stylesheet" href="/css/admin_custom.css">
87 @stop
88
89 @section('js')
90 <script> console.log('Hi!'); </script>
91 @stop
```

Nota. Código PHP de vista Registrar Docentes (edit.blade.php).

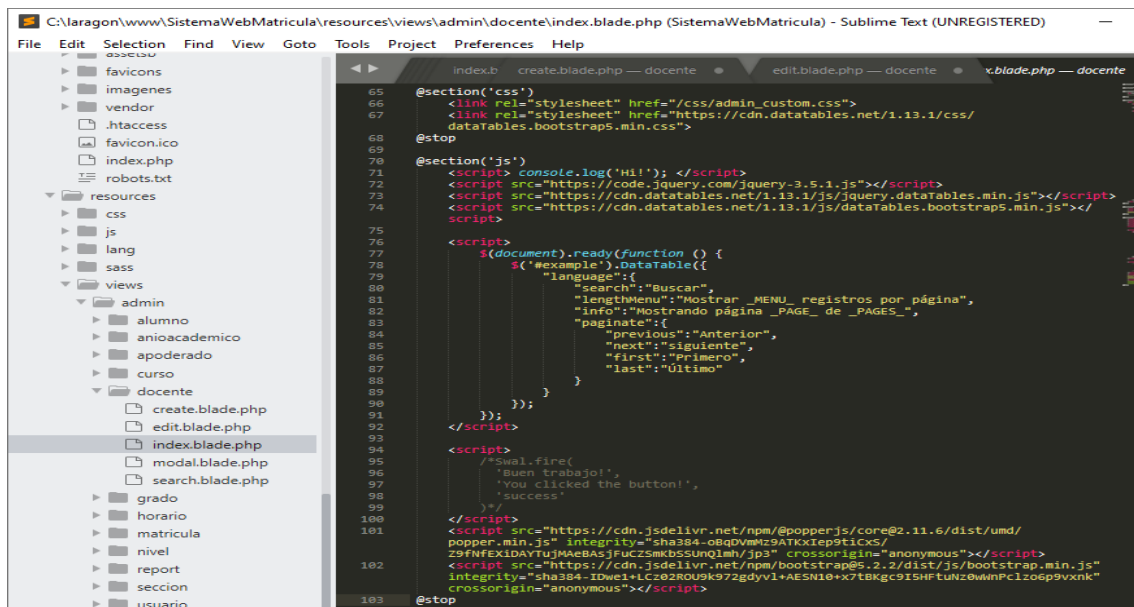
Figura 102: Vistas de Registrar Docentes (index.blade.php).



```
1 @extends('adminlte::page')
2
3 @section('title', 'Docente')
4
5 @section('content_header')
6 <h1>Listado de docentes </h1><a href="/apoderado/create"><button class="btn
7 btn-success">Nuevo</button></a></h1>
8
9 @stop
10
11 @section('content')
12 <div class="card">
13 <div class="card-header">
14 <a href="/docente/create" class="btn btn-primary">Nuevo Docente</a>
15 </div>
16 <div class="card-body">
17 <table id="example" class="table table-striped shadow-lg mt-4" style="width:
18 100%">
19 <thead class="bg-primary text-white">
20 <tr>
21 <th>Id</th>
22 <th>Nombres</th>
23 <th>Apellidos</th>
24 <th>Sexo</th>
25 <th>Email</th>
26 <th>Descripción</th>
27 <th>Telefono</th>
28 <th>Opciones</th>
29 </tr>
30 </thead>
31 <tbody>
32 @foreach($docenteA as $cat)
33 <tr>
34 <td>{{ $cat->iddocente }}</td>
35 <td>{{ $cat->nombre }}</td>
36 <td>{{ $cat->apellido }}</td>
37 <td>{{ $cat->sexo }}</td>
38 <td>{{ $cat->email }}</td>
39 <td>{{ $cat->descripcion }}</td>
40 <td>{{ $cat->celular }}</td>
41 <td width="250px">
42 <a href="{{ route('docente.edit',$cat->iddocente) }}"><button
43 class="btn btn-primary btn-sm">Editar</button></a>
44 <button type="button" class="btn btn-danger btn-sm"
45 data-bs-toggle="modal" data-bs-target="#modal-delete-{{
46 $cat->iddocente }}">
47 Eliminar
48 </button>
49 </td>
50 </tr>
51 @include('admin.docente.modal')
52 @endforeach
53 </tbody>
54 </table>
55 </div>
56 @stop
```

Nota. Código PHP de vista Registrar Docentes (index.blade.php).

Figura 103: Vistas de Registrar Docentes (index.blade.php).



```
65 @section('css')
66 <link rel="stylesheet" href="/css/admin_custom.css">
67 <link rel="stylesheet" href="https://cdn.datatables.net/1.13.1/css/
68 datatables.bootstrap.min.css">
69 @stop
70
71 @section('js')
72 <script> console.log('Hi!'); </script>
73 <script src="https://code.jquery.com/jquery-3.5.1.js"></script>
74 <script src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>
75 <script src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap.min.js"></
76 script>
77
78 <script>
79 $(document).ready(function () {
80     $('#example').DataTable({
81         "language":{
82             "search": "Buscar",
83             "lengthMenu": "Mostrar_MENU_registros por página",
84             "info": "Mostrando página_PAGE_de_PAGES_",
85             "paginate":{
86                 "previous": "Anterior",
87                 "next": "siguiente",
88                 "first": "Primero",
89                 "last": "Ultimo"
90             }
91         });
92     });
93 </script>
94
95 <script>
96 //Swal.fire(
97     'Buen trabajo!',
98     'You clicked the button!',
99     'success'
100 );
101 </script>
102 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/
103 popper.min.js" integrity="sha384-oBqVm4zA0pw79B7kA53308654332161q73O'></script>
104 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
105 integrity="sha384-IDwe1+LCz02ROU9K972gdyvl+AES10+x7tBKgc9ISH5FuZ2qWnPCl269p9vXnk"
106 crossorigin="anonymous"></script>
107 @stop
```

Nota. Código PHP de vista Registrar Docentes (index.blade.php).

Figura 104: Vistas de Registrar Docentes (modal.blade.php).

```

1 @extends('adminlte::page')
2
3 @section('title', 'Docente')
4
5 @section('content_header')
6 <h1>Listado de docentes <!--<a href="apoderado/create"><button class="btn
7 btn-success">Nuevo</button></a--></h1>
8
9 @stop
10
11 @section('content')
12 <div class="card">
13 <div class="card-header">
14 <a href="docente/create" class="btn btn-primary">Nuevo Docente</a>
15 </div>
16 <div class="card-body">
17 <table id="example" class="table table-striped shadow-lg mt-4" style="width:
18 100%">
19 <thead class="bg-primary text-white">
20 <tr>
21 <th>Id</th>
22 <th>Nombres</th>
23 <th>Apellidos</th>
24 <th>Sexo</th>
25 <th>Email</th>
26 <th>Descripción</th>
27 <th>Telefono</th>
28 <th>Opciones</th>
29 </tr>
30 </thead>
31 <tbody>
32 @foreach($docenteA as $scat)
33 <tr>
34 <td>{{ $scat->iddocente }}</td>
35 <td>{{ $scat->nombre }}</td>
36 <td>{{ $scat->apellidos }}</td>
37 <td>{{ $scat->sexo }}</td>
38 <td>{{ $scat->email }}</td>
39 <td>{{ $scat->descripcion }}</td>
40 <td>{{ $scat->celular }}</td>
41 <td width="250px">
42 <a href="{{ route('docente.edit',$scat->iddocente) }}"><button
43 class="btn btn-primary btn-sm">Editar</button></a>
44 <!-- Button trigger modal -->
45 <button type="button" class="btn btn-danger btn-sm"
46 data-bs-toggle="modal" data-bs-target="#modal-delete-{{
47 $scat->iddocente }}">
48 Eliminar
49 </button>
50 </td>
51 </tr>
52 @include('admin.docente.modal')
53 @endforeach
54 </tbody>
55 </table>
56 </div>
57 @stop

```

Nota. Código PHP de vista Registrar Docentes (modal.blade.php).

Figura 105: Vistas de Registrar Docentes (search.blade.php).

```

1 {!! Form::open(array('url'=>'admin/
2 apoderado','method'=>'GET','autocomplete'=>'off','role'=>'search')) !!}
3
4 <div class="form-group">
5 <div class="input-group">
6 <input type="text" class="form-control" name="searchText" placeholder="
7 Buscar..." value="{{ $searchText }}">
8 <span class="input-group-btn">
9 <button type="submit" class="btn btn-primary">Buscar</button>
10 </span>
11 </div>
12 {{ Form::close() }}

```

Nota. Código PHP de vista Registrar Docentes (search.blade.php).

Figura 106: Vistas de Registrar Cursos (create.blade.php).

```

1 @extends('adminlte::page')
2
3 @section('title', 'curso')
4
5 @section('plugins.Sweetalert2', true)
6
7 @section('content_header')
8 <h1>Nuevo Curso</h1>
9 @if (count($errors)>0)
10 <div class="alert alert-danger">
11 <ul>
12 @foreach($errors->all() as $errors)
13 <li>{{ $errors }}</li>
14 @endforeach
15 </ul>
16 </div>
17 @endif
18 @stop
19
20 @section('content')
21 <!-- Main content -->
22 <section class="content">
23 <div class="container-fluid">
24 <div class="row">
25 <!-- Left column -->
26 <div class="col-md-12">
27 <!-- Query validation -->
28 <div class="card card-primary">
29 <div class="card-header">
30 <h3 class="card-title"><small> En este formulario se registrará todos
31 los cursos que pertenecen a una determinada matricula</small></h3>
32 </div>
33 <!-- /.card-header -->
34 <!-- FORM START -->
35 {!! Form::open(array('url'=>'admin/
36 curso', 'method'=>'POST', 'autocomplete'=>'off', 'files'=>true)) !!}
37 {!! Form::token() !!}
38
39 <form id="quickForm">
40 <div class="form-group">
41 <label for="nombre">Nombre del curso:</label>
42 <input type="text" name="nombre" class="form-control" required value=
43 "{{ old('nombre') }}" placeholder="Ingrese el nombre del curso">
44 </div>
45 <div class="form-group">
46 <label for="nombre">Cantidad de alumnos de curso:</label>
47 <input type="text" name="cantidad" class="form-control" required
48 value="{{ old('cantidad') }}" placeholder="Ingrese cantidad">
49 </div>
50 <div class="form-group">
51 <label for="docente_id">Docente de curso:</label>
52 <select name="docente_id" class="form-control">
53 @foreach($docentes as $docent)
54 <option value="{{ $docent->iddocente }}">{{ $docent->nombre }} {{
55 $docent->apellido}}</option>
56 @endforeach
57 </select>
58 </div>
59 <div class="form-group">
60 <label for="descripcion">Descripción del curso:</label>
61 <input type="text" name="descripcion" class="form-control" required
62 value="{{ old('descripcion') }}" placeholder="Ingrese descripcion">
63 </div>
64 <!-- /.card-body -->
65 <div class="card-footer">
66 <button class="btn btn-primary" type="submit" id="btnOpenSalta" >
67 Guardar</button>
68 <button class="btn btn-danger" type="reset">Cancelar</button>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 <!-- /.col (right) -->
76 </div>
77 </div>
78 </div>
79 </div>
80 <!-- /.content -->
81 {!! Form::close() !!}
82 @stop
83
84 @section('css')
85 <link rel="stylesheet" href="/css/admin_custom.css">
86 @stop
87
88 @section('js')
89 <script> console.log('Hi!'); </script>
90 <script>
91 $(document).ready(function() {
92 $('#btnOpenSalta').click(function() {
93 Swal.fire(
94 'Buen trabajo!',
95 'Su curso a sido registrado con éxito!',
96 'success'
97 );
98 });
99
100 var Toast = Swal.mixin({
101 toast: true,
102 position: 'top-end',
103 showConfirmButton: false,
104 timer: 3000
105 });
106
107 $('#btnOpenSalta').click(function() {
108 Toast.fire({
109 icon: 'success',
110 title: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr.'
111 });
112 });
113 </script>
114 @stop
115

```

Nota. Código PHP de vista Registrar Cursos (create.blade.php).

Figura 107: Vistas de Registrar Cursos (create.blade.php).

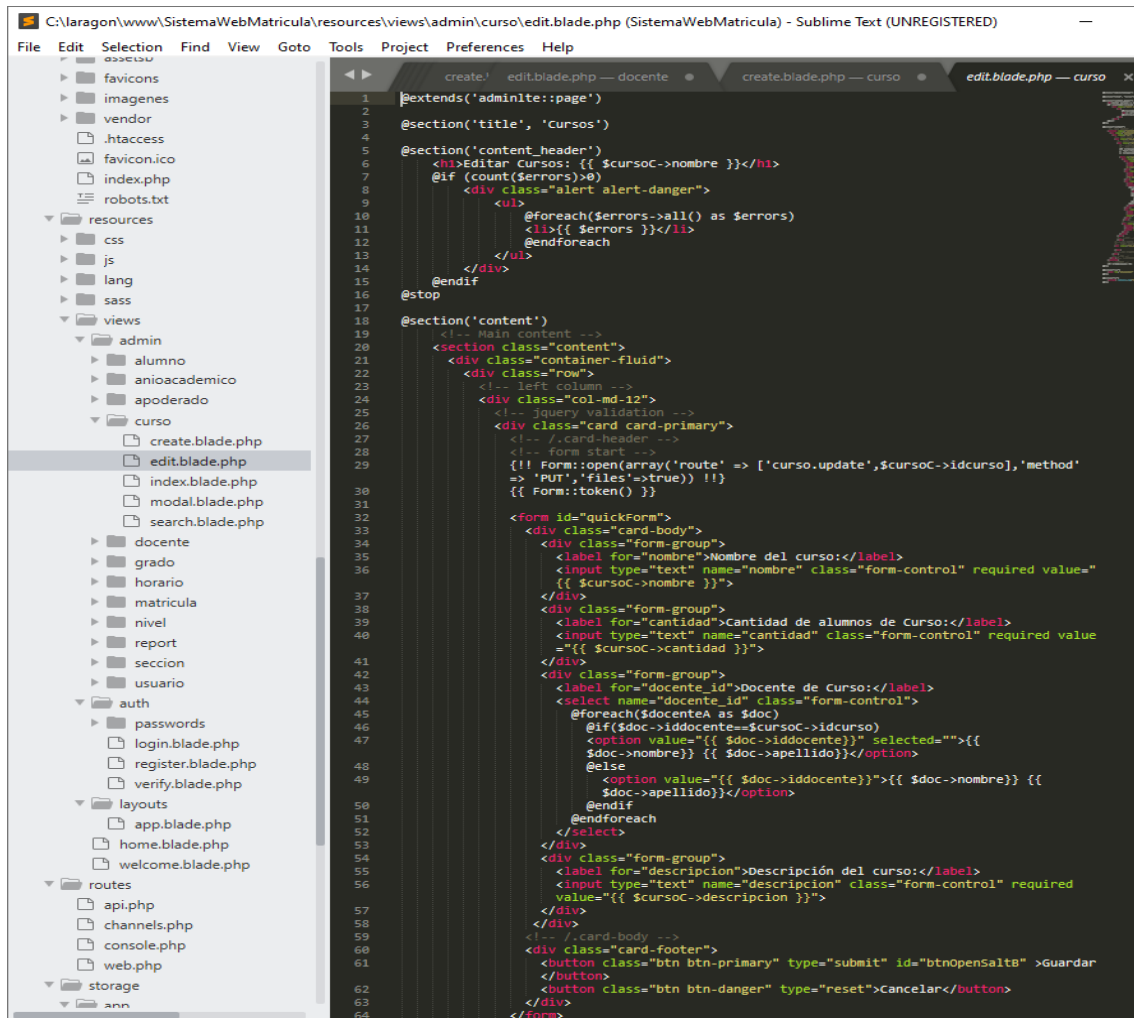
```

66 </form>
67 </div>
68 </div>
69 </div>
70 <!-- /.col (left) -->
71 <!-- right column -->
72 <div class="col-md-6">
73 </div>
74 </div>
75 <!-- /.col (right) -->
76 </div>
77 </div>
78 </div>
79 </div>
80 <!-- /.content -->
81 {!! Form::close() !!}
82 @stop
83
84 @section('css')
85 <link rel="stylesheet" href="/css/admin_custom.css">
86 @stop
87
88 @section('js')
89 <script> console.log('Hi!'); </script>
90 <script>
91 $(document).ready(function() {
92 $('#btnOpenSalta').click(function() {
93 Swal.fire(
94 'Buen trabajo!',
95 'Su curso a sido registrado con éxito!',
96 'success'
97 );
98 });
99
100 var Toast = Swal.mixin({
101 toast: true,
102 position: 'top-end',
103 showConfirmButton: false,
104 timer: 3000
105 });
106
107 $('#btnOpenSalta').click(function() {
108 Toast.fire({
109 icon: 'success',
110 title: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr.'
111 });
112 });
113 </script>
114 @stop
115

```

Nota. Código PHP de vista Registrar Cursos (create.blade.php).

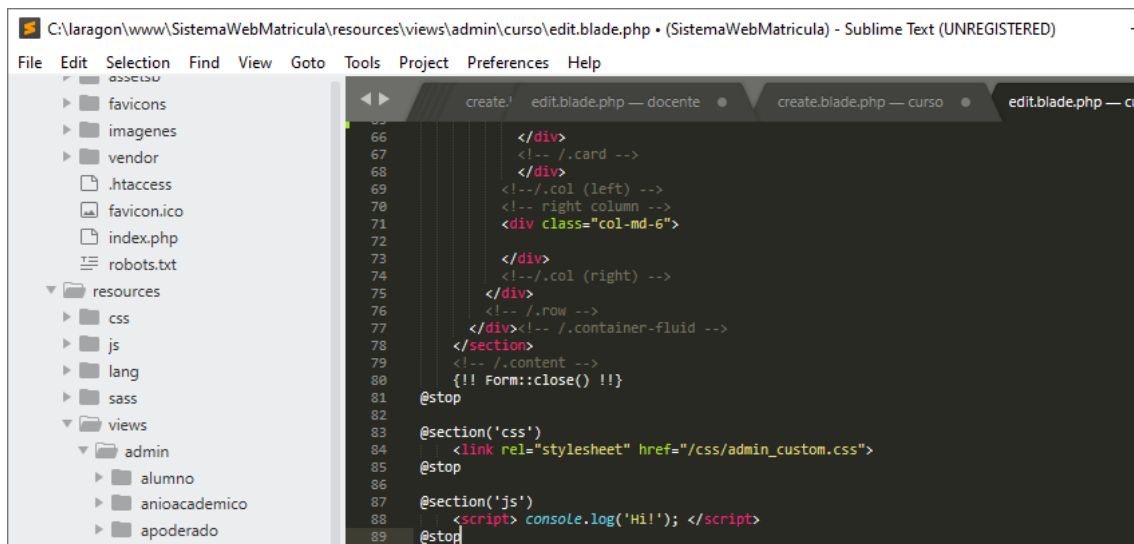
Figura 108: Vistas de Registrar Cursos (edit.blade.php).



```
1 @extends('adminlte::page')
2
3 @section('title', 'Cursos')
4
5 @section('content_header')
6 <h1>Editar Cursos: {{ $curso->nombre }}</h1>
7 @if (count($errors)>0)
8 <div class="alert alert-danger">
9 <ul>
10 @foreach($errors->all() as $errors)
11 <li>{{ $errors }}</li>
12 @endforeach
13 </ul>
14 </div>
15 @endif
16 @stop
17
18 @section('content')
19 <!-- main content -->
20 <section class="content">
21 <div class="container-fluid">
22 <div class="row">
23 <!-- left column -->
24 <div class="col-md-12">
25 <!-- jquery validation -->
26 <div class="card card-primary">
27 <!-- /.card-header -->
28 <!-- form start -->
29 {!! Form::open(array('route' => ['curso.update', $curso->idcurso], 'method'
30 => 'PUT', 'files' => true)) !!}
31 {!! Form::token() !!}
32 <form id="quickForm">
33 <div class="card-body">
34 <div class="form-group">
35 <label for="nombre">Nombre del curso:</label>
36 <input type="text" name="nombre" class="form-control" required value="
37 {{ $curso->nombre }}">
38 </div>
39 <div class="form-group">
40 <label for="cantidad">Cantidad de alumnos de curso:</label>
41 <input type="text" name="cantidad" class="form-control" required value
42 ="{{ $curso->cantidad }}">
43 </div>
44 <div class="form-group">
45 <label for="docente_id">Docente de Curso:</label>
46 <select name="docente_id" class="form-control">
47 @foreach($docenteA as $doc)
48 @if($doc->iddocente==$curso->idcurso)
49 <option value="{{ $doc->iddocente }}" selected="">{{
50 $doc->nombre}} {{ $doc->apellido}}</option>
51 @else
52 <option value="{{ $doc->iddocente }}">{{ $doc->nombre}} {{
53 $doc->apellido}}</option>
54 @endif
55 @endforeach
56 </select>
57 </div>
58 <div class="form-group">
59 <label for="descripcion">Descripción del curso:</label>
60 <input type="text" name="descripcion" class="form-control" required
61 value="{{ $curso->descripcion }}">
62 </div>
63 <!-- /.card-body -->
64 <div class="card-footer">
65 <button class="btn btn-primary" type="submit" id="btnopenSaltB">Guardar
66 </button>
67 <button class="btn btn-danger" type="reset">Cancelar</button>
68 </div>
69 </form>
70 </div>
71 <!-- /.col (left) -->
72 <div class="col-md-6">
73 </div>
74 <!-- /.col (right) -->
75 </div>
76 <!-- /.row -->
77 </div><!-- /.container-fluid -->
78 </section>
79 <!-- /.content -->
80 {!! Form::close() !!}
81 @stop
82
83 @section('css')
84 <link rel="stylesheet" href="/css/admin_custom.css">
85 @stop
86
87 @section('js')
88 <script> console.log('Hi!'); </script>
89 @stop
```

Nota. Código PHP de vista Registrar Cursos (edit.blade.php).

Figura 109: Vistas de Registrar Cursos (edit.blade.php).



```
66 </div>
67 <!-- /.card -->
68 </div>
69 <!-- /.col (left) -->
70 <!-- right column -->
71 <div class="col-md-6">
72 </div>
73 </div>
74 <!-- /.col (right) -->
75 </div>
76 <!-- /.row -->
77 </div><!-- /.container-fluid -->
78 </section>
79 <!-- /.content -->
80 {!! Form::close() !!}
81 @stop
82
83 @section('css')
84 <link rel="stylesheet" href="/css/admin_custom.css">
85 @stop
86
87 @section('js')
88 <script> console.log('Hi!'); </script>
89 @stop
```

Nota. Código PHP de vista Registrar Cursos (edit.blade.php).

Figura 110: Vistas de Registrar Cursos (index.blade.php).

```

1 @extends('adminite::page')
2
3 @section('title', 'Cursos')
4
5 @section('content_header')
6 <h1>Estado de cursos </h1><a href="apoderado/create"><button class="btn
7 btn-success">Nuevo</button></a></h1>
8
9 @stop
10
11 @section('content')
12 <div class="card">
13 <div class="card-header">
14 <a href="curso/create" class="btn btn-primary">Nuevo curso</a>
15 </div>
16 <div class="card-body">
17 <table id="example" class="table table-striped shadow-lg mt-4" style="width:
18 100%">
19 <thead class="bg-primary text-white">
20 <tr>
21 <th>Id</th>
22 <th>Nombre Curso</th>
23 <th>Cantidad</th>
24 <th>Descripcion</th>
25 <th>Docente</th>
26 <th>Opciones</th>
27 </tr>
28 </thead>
29 <tbody>
30 @foreach($cursoA as $scr)
31 <tr>
32 <td>{{ $scr->idcurso }}</td>
33 <td>{{ $scr->nombre }}</td>
34 <td>{{ $scr->xcantidad }}</td>
35 <td>{{ $scr->descripcion }}</td>
36 <td>{{ $scr->docente }}</td>
37 <td width="250px">
38 <a href="{{ route('curso.edit',$scr->idcurso) }}"><button class
39 ="btn btn-primary btn-sm">Editar</button></a>
40 <!-- Button trigger modal -->
41 <button type="button" class="btn btn-danger btn-sm"
42 data-bs-toggle="modal" data-bs-target="#modal-delete-{{
43 $scr->idcurso }}">
44 Eliminar
45 </button>
46 </td>
47 </tr>
48 </tbody>
49 @include('admin.curso.modal')
50 @endforeach
51 </tbody>
52 <!--<tfoot>
53 <tr>
54 <th>Name</th>
55 <th>Position</th>
56 <th>Office</th>
57 <th>Age</th>
58 <th>Start date</th>
59 <th>Salary</th>
60 </tr>
61 </tfoot-->
62 </table>
63 </div>
64 @stop
65
66 @section('css')
67 <link rel="stylesheet" href="/css/admin_custom.css">
68 <link rel="stylesheet" href="https://cdn.datatables.net/1.13.1/css/
69 dataTables.bootstrap5.min.css">
70 @stop
    
```

Nota. Código PHP de vista Registrar Cursos (index.blade.php).

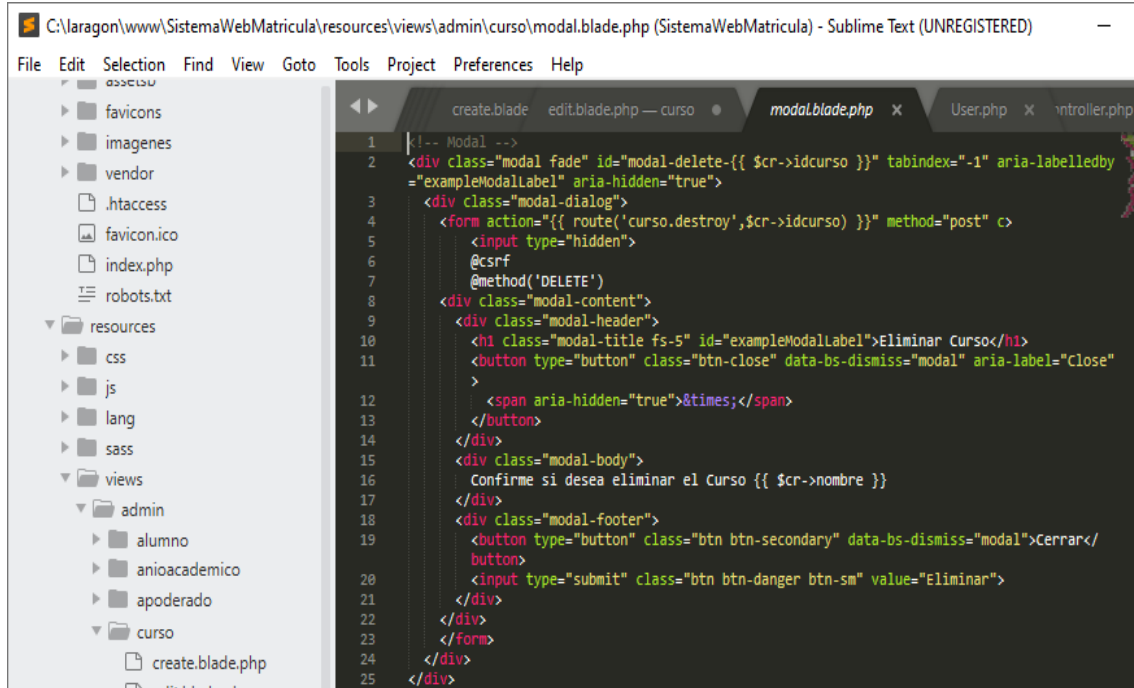
Figura 111: Vistas de Registrar Cursos (index.blade.php).

```

66 @section('js')
67 <script> console.log('Hi!'); </script>
68 <script src="https://code.jquery.com/jquery-3.5.1.js"></script>
69 <script src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>
70 <script src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap5.min.js"></
71 script>
72 <script>
73 $(document).ready(function () {
74     $('#example').DataTable({
75         "language": {
76             "search": "Buscar",
77             "lengthMenu": "Mostrar _MENU_ registros por página",
78             "info": "Mostrando página _PAGE_ de _PAGES_",
79             "paginate": {
80                 "previous": "Anterior",
81                 "next": "siguiente",
82                 "first": "Primero",
83                 "last": "Último"
84             }
85         }
86     });
87 });
88 </script>
89
90 <script>
91 /*Swal.fire(
92     '¡Bien trabajado!',
93     '¡You clicked the button!',
94     'success'
95 );*/
96 </script>
97 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
98 integrity="sha384-o8QDVmW29ATk2te4qLRJfUCZSMIvBvSx9p2fXRiSiz63K8Jebfhpw6dLlpg45f9p"
99 crossorigin="anonymous"></script>
100 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
101 integrity="sha384-IDw10CcZH90LGBwHZI+4tMjLkfa68FidQuCJ663R5Z0sbxft4DiFhtMbn51451A"
102 crossorigin="anonymous"></script>
    
```

Nota. Código PHP de vista Registrar Cursos (index.blade.php).

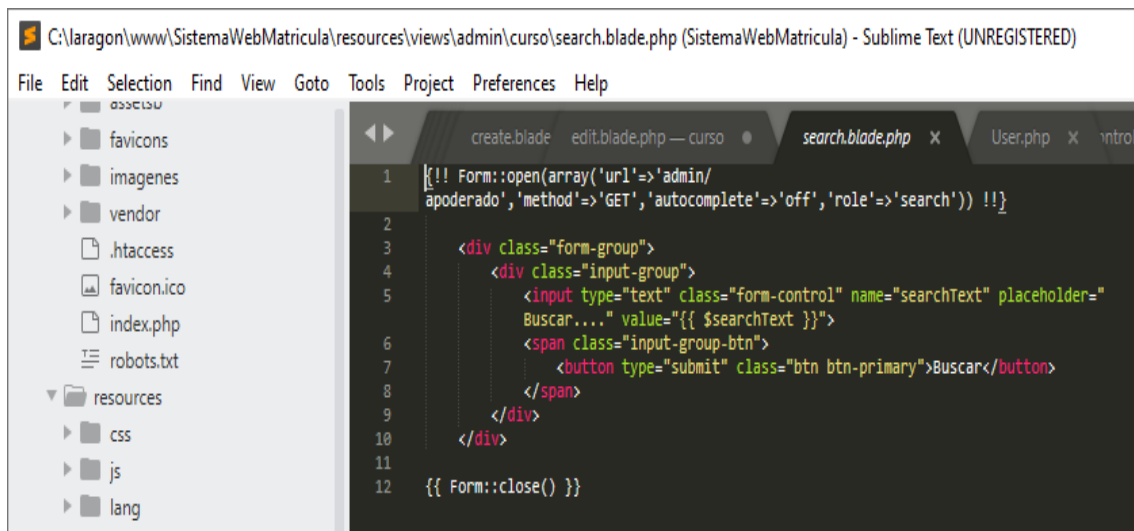
Figura 112: Vistas de Registrar Cursos (*modal.blade.php*).



```
1 |<!-- Modal -->
2 <div class="modal fade" id="modal-delete-{{ $cr->idcurso }}" tabindex="-1" aria-labelledby
  ="exampleModallabel" aria-hidden="true">
3 <div class="modal-dialog">
4 <form action="{{ route('curso.destroy',$cr->idcurso) }}" method="post" c
5 <input type="hidden">
6 @csrf
7 @method('DELETE')
8 <div class="modal-content">
9 <div class="modal-header">
10 <h1 class="modal-title fs-5" id="exampleModallabel">Eliminar Curso</h1>
11 <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"
12 >
13 </button>
14 </div>
15 <div class="modal-body">
16 Confirme si desea eliminar el Curso {{ $cr->nombre }}
17 </div>
18 <div class="modal-footer">
19 <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</
  button>
20 <input type="submit" class="btn btn-danger btn-sm" value="Eliminar">
21 </div>
22 </div>
23 </form>
24 </div>
25 </div>
```

Nota. Código PHP de vista Registrar Cursos (*modal.blade.php*).

Figura 113: Vistas de Registrar Cursos (*search.blade.php*).



```
1 {!! Form::open(array('url'=>'admin/
  apoderado','method'=>'GET','autocomplete'=>'off','role'=>'search')) !!}
2
3 <div class="form-group">
4 <div class="input-group">
5 <input type="text" class="form-control" name="searchText" placeholder="
  Buscar..." value="{{ $searchText }}">
6 <span class="input-group-btn">
7 <button type="submit" class="btn btn-primary">Buscar</button>
8 </span>
9 </div>
10 </div>
11
12 {!! Form::close() !!}
```

Nota. Código PHP de vista Registrar Cursos (*search.blade.php*).

Figura 114: Vistas de Gestionar Horario (create.blade.php).

```

1 @extends('adminlte::page')
2
3 @section('title', 'Horario')
4
5 @section('plugins.sweetalert2', true)
6
7 @section('content_header')
8 <h1>Nuevo Horario</h1>
9 @if (count($errors) > 0)
10 <div class="alert alert-danger">
11 <ul>
12 @foreach($errors->all() as $errors)
13 <li>{{ $errors }}</li>
14 @endforeach
15 </ul>
16 </div>
17 @endif
18 @stop
19
20 @section('content')
21 <!-- Main content -->
22 <section class="content">
23 <div class="container-fluid">
24 <div class="row">
25 <!-- left column -->
26 <div class="col-md-12">
27 <!-- jquery validation -->
28 <div class="card card-primary">
29 <div class="card-header">
30 <div class="card-title"><small>En este formulario se registrará todos los
31 horarios que pertenecen a un determinado curso</small></div>
32 <!-- /.card-header -->
33 <!-- form start -->
34 {!! Form::open(array('url'=>'admin/
35 horario', 'method'=>'POST', 'autocomplete'=>'off', 'files'=>true)) !!}
36 {!! Form::token() !!}
37
38 <form id="quickForm">
39 <div class="card-body">
40 <div class="form-group">
41 <label for="dia">Día de la semana:</label>
42 <input type="text" name="dia" class="form-control" id="
43 exampleInputEmail" placeholder="Ingrese el día">
44 </div>
45 <div class="form-group">
46 <label for="hora_inicio">Hora de Inicio:</label>
47 <input type="time" name="hora_inicio" />
48 </div>
49 <div class="form-group">
50 <label for="hora_fin">Hora de Fin:</label>
51 <input type="time" name="hora_fin" />
52 </div>
53 </div> /.card-body -->
54 <div class="card-footer">
55 <button class="btn btn-primary" type="submit" id="btnOpenSaltB" >
56 Guardar</button>
57 <button class="btn btn-danger" type="reset">Cancelar</button>
58 </div>
59 </form>
60 </div>
61 <!-- /.col (left) -->
62 <!-- right column -->
63 <div class="col-md-6">
64
65 </div>
66 <!-- /.col (right) -->
67 </div>

```

Nota. Código PHP de vista Gestionar Horario (create.blade.php).

Figura 115: Vistas de Gestionar Horario (create.blade.php).

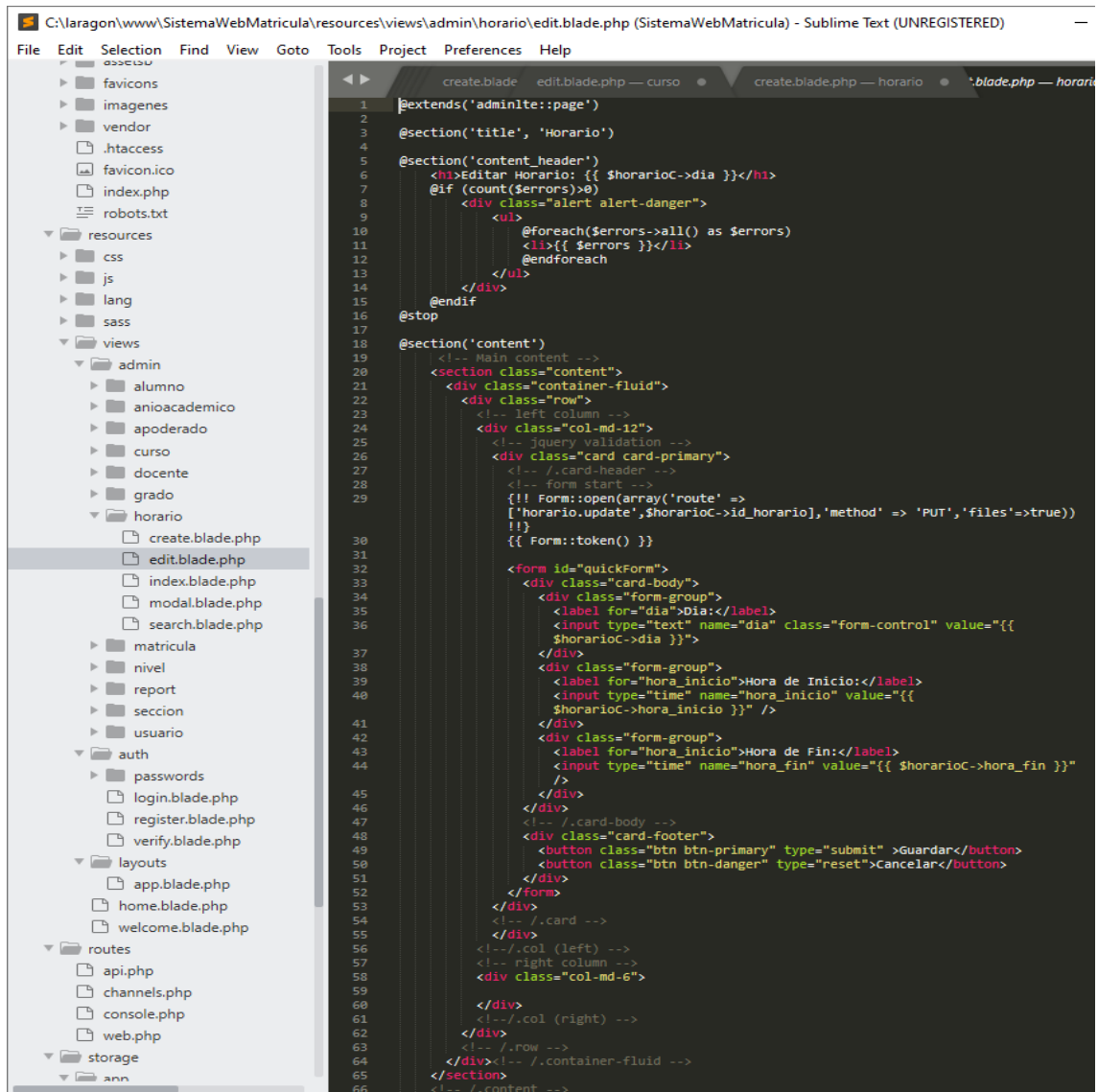
```

69 </div> /.row -->
70 </div> /.container-fluid -->
71 </section>
72 </div> /.content -->
73 {!! Form::close() !!}
74 @stop
75
76 @section('css')
77 <link rel="stylesheet" href="/css/admin_custom.css">
78 @stop
79
80 @section('js')
81 <script> console.log('HI!'); </script>
82 <script>
83 $(document).ready(function() {
84 $('#btnOpenSaltB').click(function() {
85 Swal.fire({
86 'title': 'Buen trabajo!',
87 'text': 'El horario ha sido registrado con éxito!',
88 'success': true,
89 });
90 });
91
92 var Toast = Swal.mixin({
93 toast: true,
94 position: 'top-end',
95 showConfirmButton: false,
96 timer: 3000
97 });
98
99 $('#btnOpenSaltC').click(function() {
100 Toast.fire({
101 icon: 'success',
102 title: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr.'
103 });
104 });
105
106 </script>
107 @stop

```

Nota. Código PHP de vista Gestionar Horario (create.blade.php).

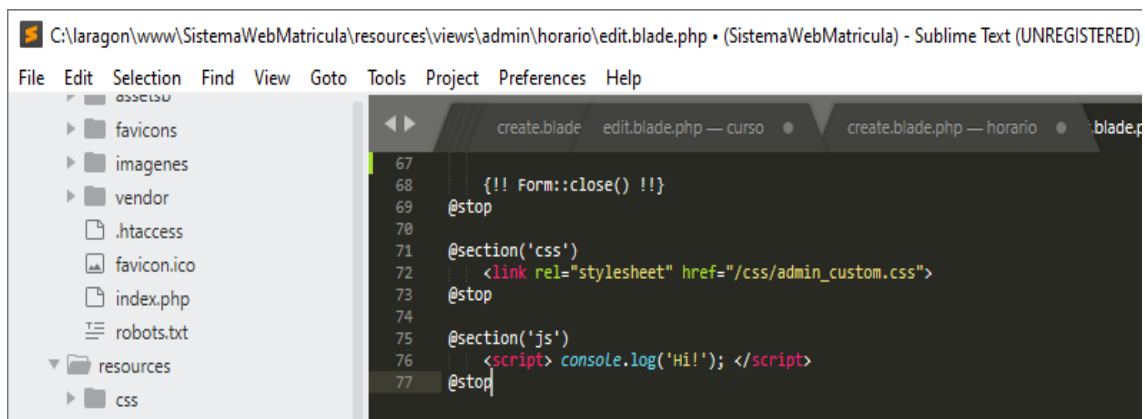
Figura 116: Vistas de Gestionar Horario (edit.blade.php).



```
1 @extends('adminlte::page')
2
3 @section('title', 'Horario')
4
5 @section('content header')
6 <h1>Editar Horario: {{ $horarioC->dia }}</h1>
7 @if (count($errors)>0)
8 <div class="alert alert-danger">
9 <ul>
10 @foreach($errors->all() as $errors)
11 <li>{{ $errors }}</li>
12 @endforeach
13 </ul>
14 </div>
15 @endif
16 @stop
17
18 @section('content')
19 <!-- Main content -->
20 <section class="content">
21 <div class="container-fluid">
22 <div class="row">
23 <!-- left column -->
24 <div class="col-md-12">
25 <!-- query validation -->
26 <div class="card card-primary">
27 <!-- /.card-header -->
28 <!-- form start -->
29 {!! Form::open(array('route' =>
30 ['horario.update',$horarioC->id_horario],'method' => 'PUT','files'=>true))
31 !!}
32 {!! Form::token() !!}
33
34 <form id="quickForm">
35 <div class="card-body">
36 <div class="form-group">
37 <label for="dia">Dia:</label>
38 <input type="text" name="dia" class="form-control" value="{{
39 $horarioC->dia }}" />
40 </div>
41 <div class="form-group">
42 <label for="hora_inicio">Hora de Inicio:</label>
43 <input type="time" name="hora_inicio" value="{{
44 $horarioC->hora_inicio }}" />
45 </div>
46 <div class="form-group">
47 <label for="hora_fin">Hora de Fin:</label>
48 <input type="time" name="hora_fin" value="{{ $horarioC->hora_fin }}"
49 />
50 </div>
51 </div>
52 <div class="card-footer">
53 <button class="btn btn-primary" type="submit" >Guardar</button>
54 <button class="btn btn-danger" type="reset" >Cancelar</button>
55 </div>
56 </form>
57 </div>
58 </div>
59 <!-- /.card -->
60 </div>
61 <!-- /col (left) -->
62 <!-- right column -->
63 <div class="col-md-6">
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 @stop
71
72 @section('css')
73 <link rel="stylesheet" href="/css/admin_custom.css">
74 @stop
75
76 @section('js')
77 <script> console.log('Hi!'); </script>
78 @stop
```

Nota. Código PHP de vista Gestionar Horario (edit.blade.php).

Figura 117: Vistas de Gestionar Horario (edit.blade.php).



```
67
68 {!! Form::close() !!}
69 @stop
70
71 @section('css')
72 <link rel="stylesheet" href="/css/admin_custom.css">
73 @stop
74
75 @section('js')
76 <script> console.log('Hi!'); </script>
77 @stop
```

Nota. Código PHP de vista Gestionar Horario (edit.blade.php).

Figura 118: Vistas de Gestionar Horario (index.blade.php).

```

1 @extends('adminlte::page')
2
3 @section('title', 'Horario')
4
5 @section('content_header')
6 <h1>Listado de horarios <!--<a href="apoderado/create"><button class="btn
7 btn-success">Nuevo</button></a--></h1>
8
9 @stop
10
11 @section('content')
12 <div class="card">
13 <div class="card-header">
14 <a href="horario/create" class="btn btn-primary">Nuevo Horario</a>
15 </div>
16 <div class="card-body">
17 <table id="example" class="table table-striped shadow-lg mt-4" style="width:
18 100%">
19 <thead class="bg-primary text-white">
20 <tr>
21 <th>ID</th>
22 <th>Dia</th>
23 <th>Hora de inicio</th>
24 <th>Hora de fin</th>
25 <th>Opciones</th>
26 </tr>
27 </thead>
28 <tbody>
29 <tr>
30 <td>{{ $horario->id_horario }}</td>
31 <td>{{ $horario->dia }}</td>
32 <td>{{ $horario->hora_inicio }}</td>
33 <td>{{ $horario->hora_fin }}</td>
34 <td width="250px">
35 <a href="{{ route('horario.edit',$horario->id_horario) }}"><button
36 class="btn btn-primary btn-sm">Editar</button></a>
37 <!-- <button trigger="modal" -->
38 <button type="button" class="btn btn-danger btn-sm"
39 data-bs-toggle="modal" data-bs-target="#modal-delete-{{
40 $horario->id_horario }}">
41 Eliminar
42 </button>
43 </td>
44 </tr>
45 </tbody>
46 @include('admin.horario.modal')
47 @endforeach
48 </tbody>
49 <!--<tfoot>
50 <tr>
51 <th>Name</th>
52 <th>Position</th>
53 <th>Office</th>
54 <th>Age</th>
55 <th>Start date</th>
56 <th>Salary</th>
57 </tr>
58 </tfoot-->
59 </table>
60 </div>
61 @stop
62
63 @section('css')
64 <link rel="stylesheet" href="/css/admin_custom.css">
65 <link rel="stylesheet" href="https://cdn.datatables.net/1.13.1/css/
66 dataTables.bootstrap5.min.css">
67 @stop
68
69 @section('js')
70 <script> console.log('Hi!'); </script>
71

```

Nota. Código PHP de vista Gestionar Horario (index.blade.php).

Figura 119: Vistas de Gestionar Horario (index.blade.php).

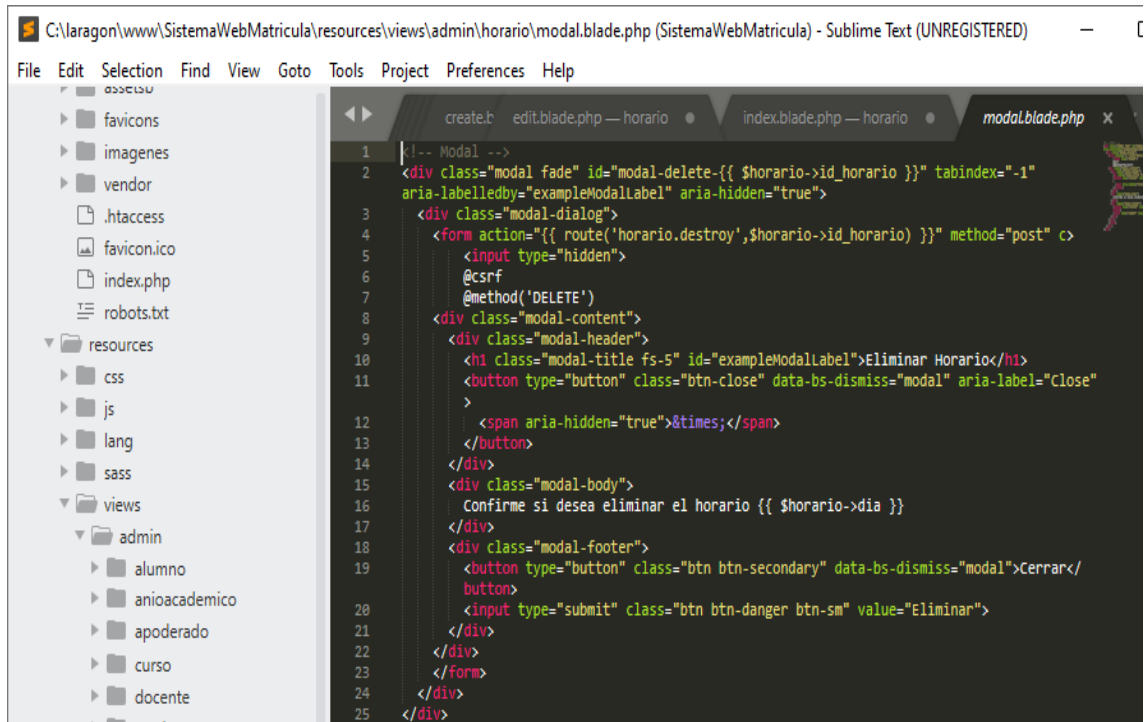
```

67 <script src="https://code.jquery.com/jquery-3.5.1.js"></script>
68 <script src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>
69 <script src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap5.min.js"></
70 script>
71
72 <script>
73 $(document).ready(function () {
74     $('#example').DataTable({
75         "language": {
76             "search": "Buscar",
77             "lengthMenu": "Mostrar MENU_Registros por página",
78             "info": "Mostrando página _PAGE_ de _PAGES_",
79             "paginate": {
80                 "previous": "Anterior",
81                 "next": "siguiente",
82                 "first": "Primero",
83                 "last": "Último"
84             }
85         }
86     });
87 </script>
88
89 <script>
90 /*Swal.fire(
91     'Buen trabajo!',
92     'You clicked the button!',
93     'success'
94 );*/
95 </script>
96 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js
97 integrity="sha384-oBqDVmW9ATx2DPJdIqMAYvdd/tp9tIXS929fHFEXIDAYTUJMAEbasJFUCzSMKbDSSURQlMh/jpa"
98 crossorigin="anonymous"></script>
99 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
100 integrity="sha384-IDw1+Lz2R0U9k972gdyv1+AESN10+x7tBKgc9I5HFtUNz0WwPclz0p9vxnk"
101 crossorigin="anonymous"></script>
102 @stop

```

Nota. Código PHP de vista Gestionar Horario (index.blade.php).


Figura 120: Vistas de Gestionar Horario (modal.blade.php).



```
1 <!-- Modal -->
2 <div class="modal fade" id="modal-delete-{{ $horario->id_horario }}" tabindex="-1"
  aria-labelledby="exampleModallabel" aria-hidden="true">
3   <div class="modal-dialog">
4     <form action="{{ route('horario.destroy',$horario->id_horario) }}" method="post" c>
5       <input type="hidden">
6       @csrf
7       @method('DELETE')
8     <div class="modal-content">
9       <div class="modal-header">
10        <h1 class="modal-title fs-5" id="exampleModallabel">Eliminar Horario</h1>
11        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="close"
12        >
13          <span aria-hidden="true">&times;</span></button>
14      </div>
15      <div class="modal-body">
16        Confirme si desea eliminar el horario {{ $horario->dia }}
17      </div>
18      <div class="modal-footer">
19        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</
20        button>
21        <input type="submit" class="btn btn-danger btn-sm" value="Eliminar">
22      </div>
23    </form>
24  </div>
25 </div>
```

Nota. Código PHP de vista Gestionar Horario (modal.blade.php).

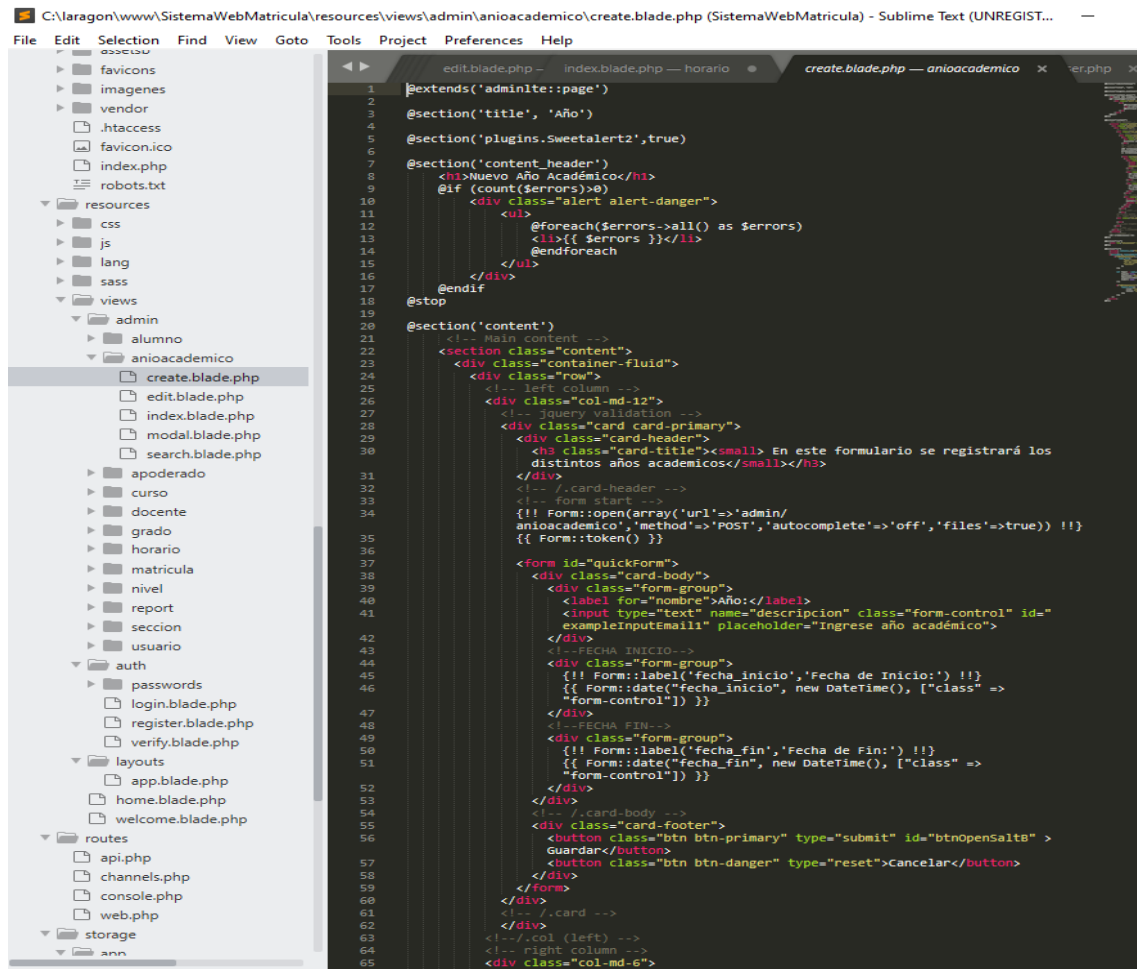
Figura 121: Vistas de Gestionar Horario (search.blade.php).



```
1 {!! Form::open(array('url'=>'admin/
  apoderado','method'=>'GET','autocomplete'=>'off','role'=>'search')) !!}
2
3   <div class="form-group">
4     <div class="input-group">
5       <input type="text" class="form-control" name="searchText" placeholder="
6       Buscar..." value="{{ $searchText }}">
7       <span class="input-group-btn">
8         <button type="submit" class="btn btn-primary">Buscar</button>
9       </span>
10    </div>
11  </div>
12  {!! Form::close() !!}
```

Nota. Código PHP de vista Gestionar Horario (search.blade.php).

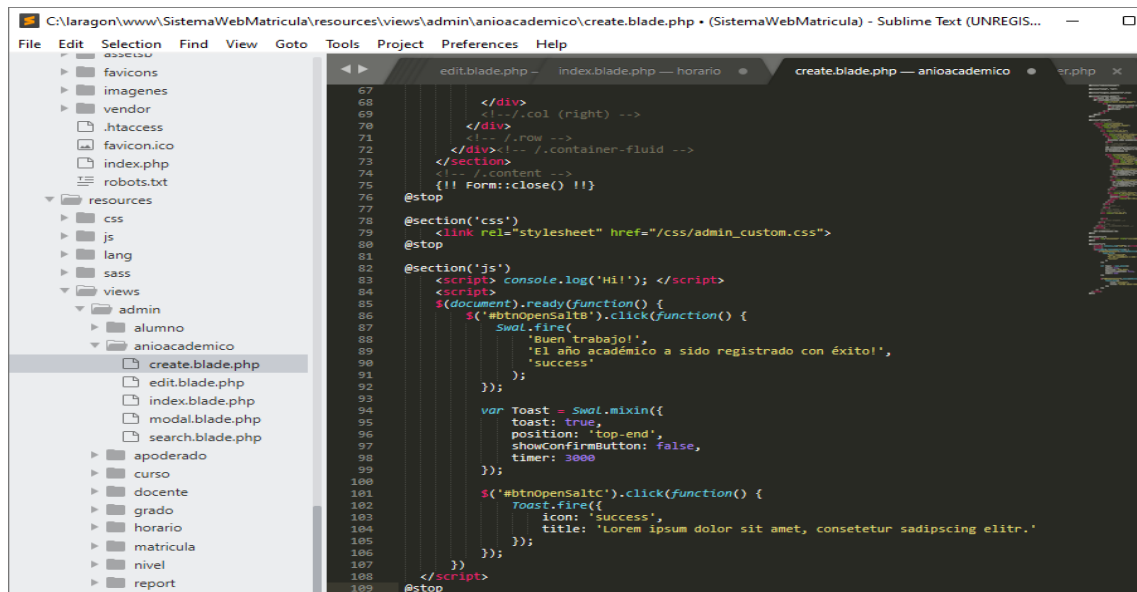
Figura 122: Vistas de Gestionar Año Académico (create.blade.php).



```
1 @extends('adminlte::page')
2
3 @section('title', 'Año')
4
5 @section('plugins.Sweetalert2', true)
6
7 @section('content_header')
8 <h1>Nuevo Año Académico</h1>
9 @if (count($errors)>0)
10 <div class="alert alert-danger">
11 <ul>
12 @foreach($errors->all() as $errors)
13 <li>{{ $errors }}</li>
14 @endforeach
15 </ul>
16 </div>
17 @endif
18 @stop
19
20 @section('content')
21 <!-- Main content -->
22 <section class="content">
23 <div class="container-fluid">
24 <div class="row">
25 <!-- left column -->
26 <div class="col-md-12">
27 <!-- jquery validation -->
28 <div class="card card-primary">
29 <div class="card-header">
30 <h3 class="card-title"><small> En este formulario se registrará los
31 distintos años academicos</small></h3>
32 </div>
33 <!-- /.card-header -->
34 <!-- form start -->
35 {!! Form::open(array('url'=>'admin/
36 anioacademico', 'method'=>'POST', 'autocomplete'=>'off', 'files'=>true)) !!}
37 {!! Form::token() !!}
38 <form id="quickForm">
39 <div class="form-group">
40 <label for="nombre">Año:</label>
41 <input type="text" name="descripcion" class="form-control" id="
42 exampleinputEmail1" placeholder="Ingrese año académico">
43 </div>
44 <!-- FECHA INICIO -->
45 <div class="form-group">
46 {!! Form::label('fecha_inicio', 'Fecha de Inicio:') !!}
47 {!! Form::date('fecha_inicio', new DateTime(), ["class" =>
48 "form-control"]) !!}
49 </div>
50 <!-- FECHA FIN -->
51 <div class="form-group">
52 {!! Form::label('fecha_fin', 'Fecha de Fin:') !!}
53 {!! Form::date('fecha_fin', new DateTime(), ["class" =>
54 "form-control"]) !!}
55 </div>
56 </div>
57 <!-- /.card-body -->
58 <div class="card-footer">
59 <button class="btn btn-primary" type="submit" id="btnOpenSaltB">
60 Guardar</button>
61 <button class="btn btn-danger" type="reset">Cancelar</button>
62 </div>
63 </div>
64 </div>
65 <!-- right column -->
66 <div class="col-md-6">
```

Nota. Código PHP de vista Gestionar Año Académico (create.blade.php).

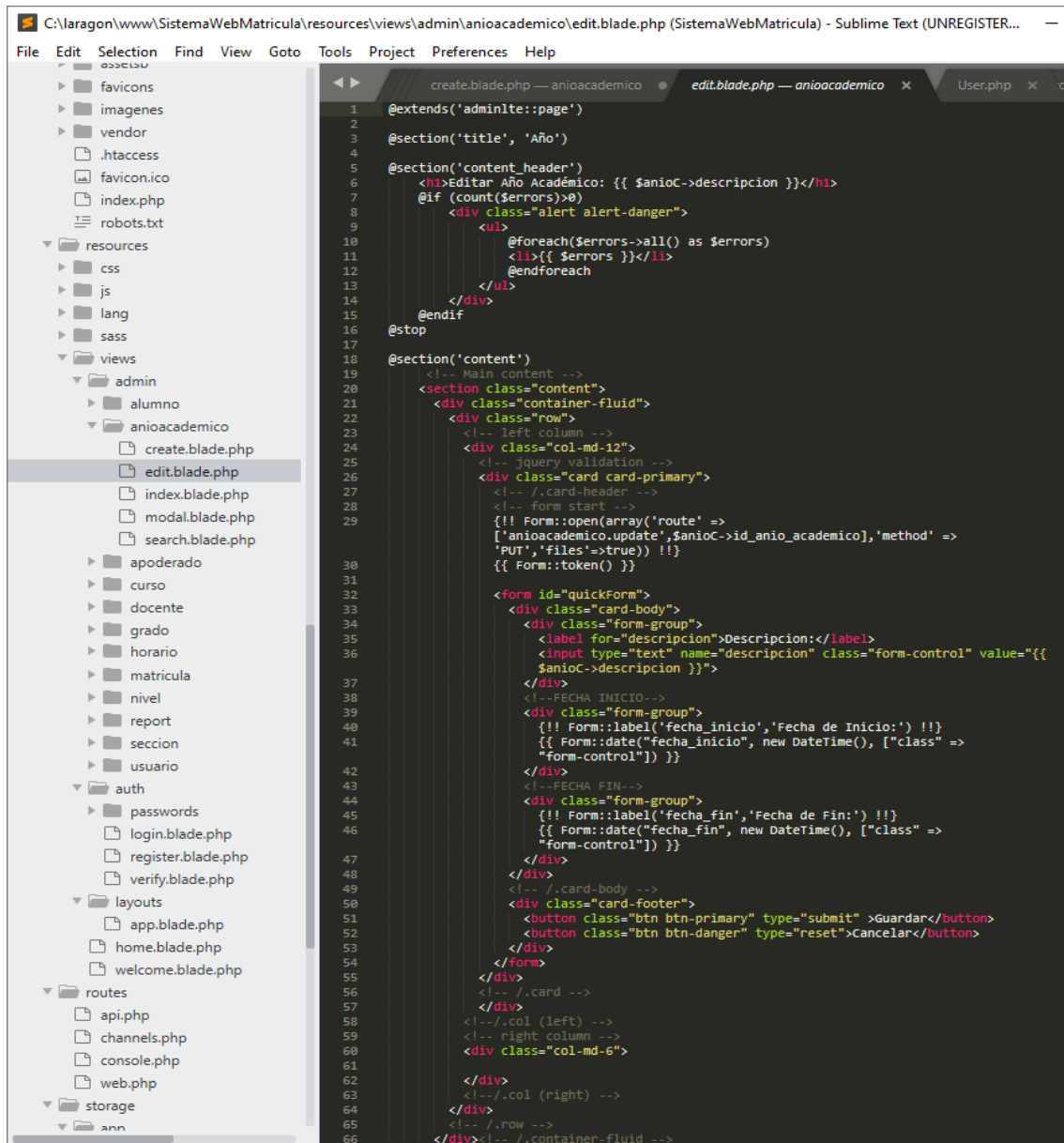
Figura 123: Vistas de Gestionar Año Académico (create.blade.php).



```
67 </div>
68 <!-- /.col (right) -->
69 </div>
70 <!-- /.row -->
71 </div><!-- /.container-fluid -->
72 </section>
73 <!-- /.content -->
74 {!! Form::close() !!}
75 @stop
76
77 @section('css')
78 <link rel="stylesheet" href="/css/admin_custom.css">
79 @stop
80
81 @section('js')
82 <script> console.log('Hi!'); </script>
83 <script>
84 $(document).ready(function() {
85 $(' #btnOpenSaltB').click(function() {
86 Swal.fire(
87 'Buen trabajo!',
88 'El año académico a sido registrado con éxito!',
89 'success'
90 );
91 });
92
93 var Toast = Swal.mixin({
94 toast: true,
95 position: 'top-end',
96 showConfirmButton: false,
97 timer: 3000
98 });
99
100 $(' #btnOpenSaltC').click(function() {
101 Toast.fire({
102 icon: 'success',
103 title: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr.'
104 });
105 });
106
107 </script>
108 @stop
```

Nota. Código PHP de vista Gestionar Año Académico (create.blade.php).

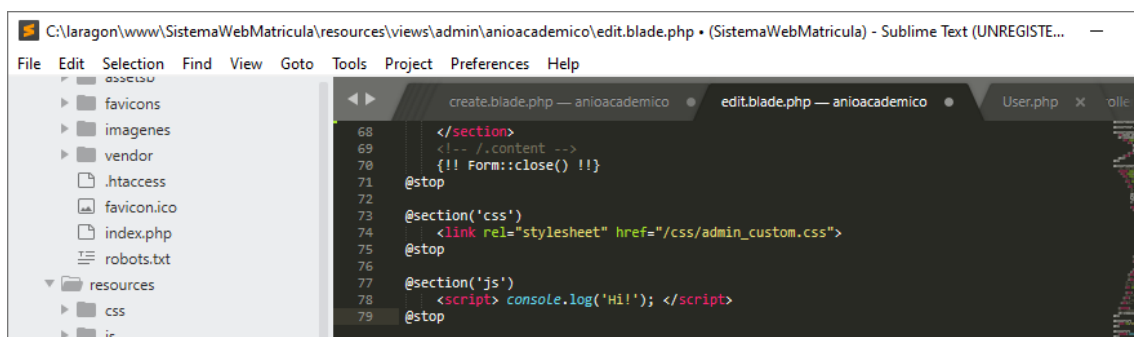
Figura 124: Vistas de Gestionar Año Académico (edit.blade.php).



```
1 @extends('adminlte::page')
2
3 @section('title', 'Año')
4
5 @section('content_header')
6 <h1>Editar Año Académico: {{ $anioC->descripcion }}</h1>
7 @if (count($errors)>0)
8 <div class="alert alert-danger">
9 <ul>
10 @foreach($errors->all() as $errors)
11 <li>{{ $errors }}</li>
12 @endforeach
13 </ul>
14 </div>
15 @endif
16 @stop
17
18 @section('content')
19 <!-- Main content -->
20 <section class="content">
21 <div class="container-fluid">
22 <div class="row">
23 <!-- left column -->
24 <div class="col-md-12">
25 <!-- jquery validation -->
26 <div class="card card-primary">
27 <!-- /.card-header -->
28 <!-- form start -->
29 {!! Form::open(array('route' =>
30 ['anioacademico.update',$anioC->id_anio_academico],'method' =>
31 'PUT','files'=>true) !!)
32 {!! Form::token() !!}
33
34 <form id="quickForm">
35 <div class="card-body">
36 <div class="form-group">
37 <label for="descripcion">Descripcion:</label>
38 <input type="text" name="descripcion" class="form-control" value="{{
39 $anioC->descripcion }}">
40 </div>
41 <!-- FECHA INICIO-->
42 <div class="form-group">
43 {!! Form::label('fecha_inicio','Fecha de Inicio:') !!}
44 {!! Form::date("fecha_inicio", new DateTime(), ["class" =>
45 "form-control"]) !!}
46 </div>
47 <!-- FECHA FIN-->
48 <div class="form-group">
49 {!! Form::label('fecha_fin','Fecha de Fin:') !!}
50 {!! Form::date("fecha_fin", new DateTime(), ["class" =>
51 "form-control"]) !!}
52 </div>
53 </div>
54 <!-- /.card-body -->
55 <div class="card-footer">
56 <button class="btn btn-primary" type="submit" >Guardar</button>
57 <button class="btn btn-danger" type="reset" >Cancelar</button>
58 </div>
59 </form>
60 </div>
61 </div>
62 <!-- /.card -->
63 </div>
64 <!-- right column -->
65 <div class="col-md-6">
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
```

Nota. Código PHP de vista Gestionar Año Académico (edit.blade.php).

Figura 125: Vistas de Gestionar Año Académico (edit.blade.php).



```
68 </section>
69 <!-- /.content -->
70 {!! Form::close() !!}
71 @stop
72
73 @section('css')
74 <link rel="stylesheet" href="/css/admin_custom.css">
75 @stop
76
77 @section('js')
78 <script> console.log('Hi!'); </script>
79 @stop
```

Nota. Código PHP de vista Gestionar Año Académico (edit.blade.php).

Figura 126: Vistas de Gestionar Año Académico (index.blade.php).

```

1 @extends('adminlte::page')
2
3 @section('title', 'Año')
4
5 @section('content_header')
6 <h1>Listado de años académicos <!--<a href="apoderado/create"><button class="btn
7 btn-success">Nuevo</button></a--></h1>
8
9 @stop
10
11 @section('content')
12 <div class="card">
13 <div class="card-header">
14 <a href="anioacademico/create" class="btn btn-primary">Nuevo Año Académico</a>
15 </div>
16 <div class="card-body">
17 <table id="example" class="table table-striped shadow-lg mt-4" style="width:
18 100%">
19 <thead class="bg-primary text-white">
20 <tr>
21 <th>Id</th>
22 <th>Descripción</th>
23 <th>Fecha Inicio</th>
24 <th>Fecha Fin</th>
25 <th>Opciones</th>
26 </tr>
27 </thead>
28 <tbody>
29 @foreach($anio as $anio)
30 <tr>
31 <td>{{ $anio->id_anio_academico }}</td>
32 <td>{{ $anio->descripcion }}</td>
33 <td>{{ $anio->fecha_inicio }}</td>
34 <td>{{ $anio->fecha_fin }}</td>
35 <td width="250px">
36 <a href="{{
37 route('anioacademico.edit',$anio->id_anio_academico) }}">
38 <button class="btn btn-primary btn-sm">Editar</button></a>
39 <!-- Button trigger modal -->
40 <button type="button" class="btn btn-danger btn-sm"
41 data-bs-toggle="modal" data-bs-target="#modal-delete-{{
42 $anio->id_anio_academico }}">
43 Eliminar
44 </button>
45 </td>
46 </tr>
47 @include('admin.anioacademico.modal')
48 @endforeach
49 </tbody>
50 </tfoot>
51 <!--tfoot-->
52 <tr>
53 <th>Name</th>
54 <th>Position</th>
55 <th>Office</th>
56 <th>Age</th>
57 <th>Start date</th>
58 <th>Salary</th>
59 </tr>
60 </tfoot-->
61 </table>
62 </div>
63 @stop
64
65 @section('css')
66 <link rel="stylesheet" href="/css/admin_custom.css">
67 <link rel="stylesheet" href="https://cdn.datatables.net/1.13.1/css/
68 dataTables.bootstrap5.min.css">
69
70 @stop

```

Nota. Código PHP de vista Gestionar Año Académico (index.blade.php).

Figura 127: Vistas de Gestionar Año Académico (index.blade.php).

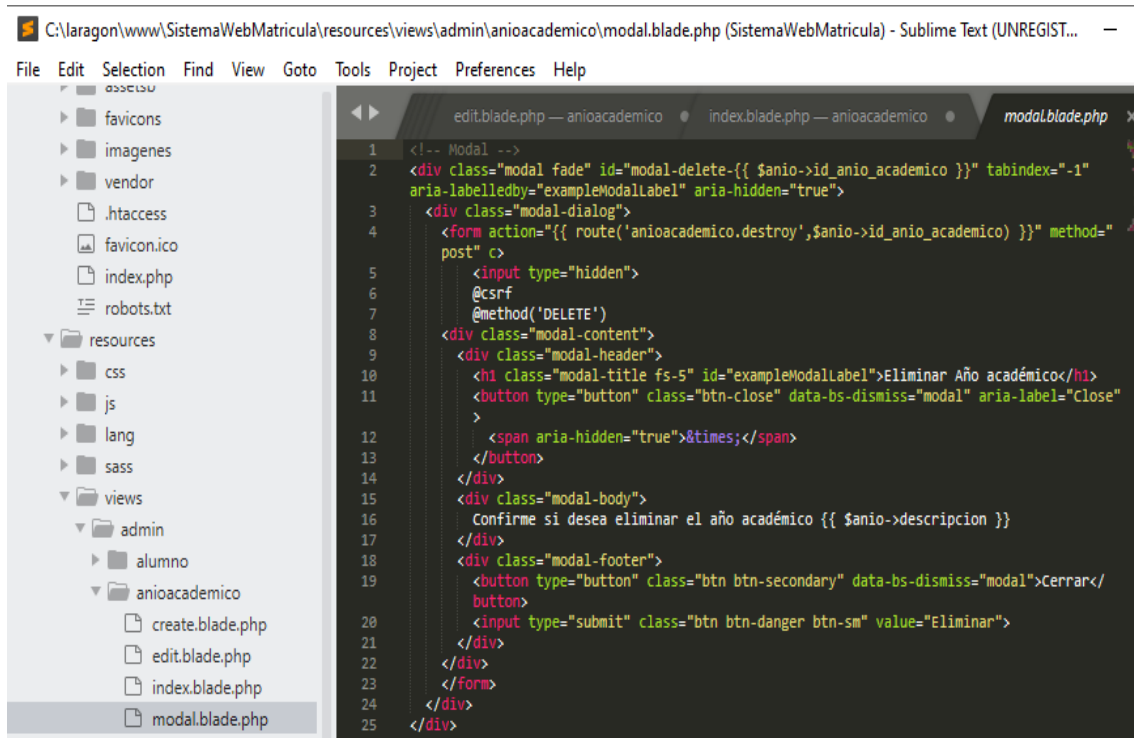
```

65 @section('js')
66 <script> console.log('Hi!'); </script>
67 <script src="https://code.jquery.com/jquery-3.5.1.js"></script>
68 <script src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>
69 <script src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap5.min.js"></
70 script>
71
72 <script>
73 $(document).ready(function () {
74     $('#example').DataTable({
75         "language": {
76             "search": "Buscar",
77             "lengthMenu": "Mostrar _MENU_ registros por página",
78             "info": "Mostrando página _PAGE_ de _PAGES_",
79             "paginate": {
80                 "previous": "Anterior",
81                 "next": "Siguiente",
82                 "first": "Primero",
83                 "last": "Último"
84             }
85         }
86     });
87 </script>
88
89 <script>
90 //swal.fire(
91     'Buen trabajo!',
92     'You clicked the button!',
93     'success'
94 );
95 </script>
96 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js
97 integrity="sha384-oBqDVmMz9ATKzIep9tiCxS/Z9fNfEXiDAYtUJmAEAsJFUCZsmkBSUnQlhm/jp3"
98 crossorigin="anonymous"></script>
99 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
100 integrity="sha384-IDwe1+Ic0e0R09k97dyv1+AESN10+x7tBKgc9IShtFtunZ0wPCL20p9vknk"
101 crossorigin="anonymous"></script>
102 @stop

```

Nota. Código PHP de vista Gestionar Año Académico (index.blade.php).

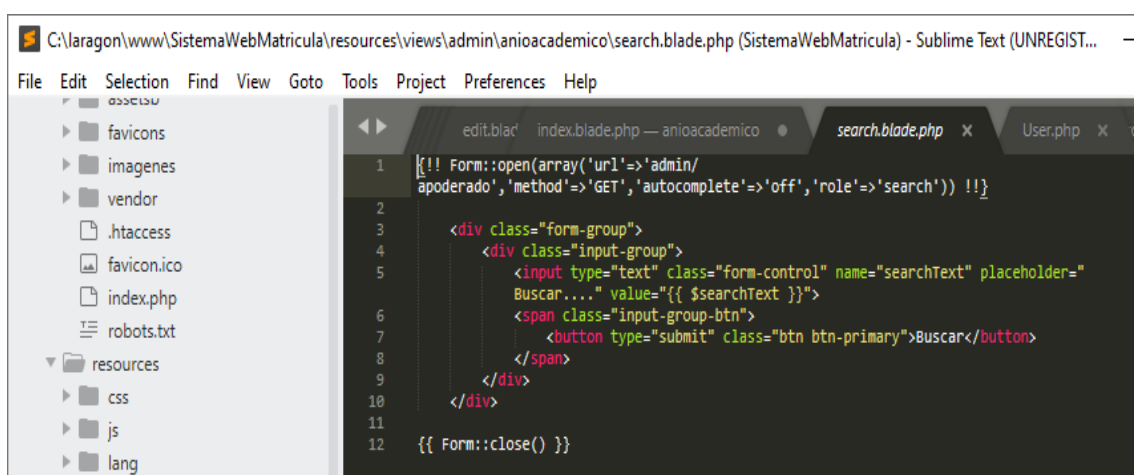
Figura 128: Vistas de Gestionar Año Académico (modal.blade.php).



```
1 <!-- Modal -->
2 <div class="modal fade" id="modal-delete-{{ $anio->id_anio_academico }}" tabindex="-1"
3   aria-labelledby="exampleModallabel" aria-hidden="true">
4   <div class="modal-dialog">
5     <form action="{{ route('anioacademico.destroy',$anio->id_anio_academico) }}" method="
6     post" >
7       <input type="hidden">
8       @csrf
9       @method('DELETE')
10      <div class="modal-content">
11        <div class="modal-header">
12          <h1 class="modal-title fs-5" id="exampleModallabel">Eliminar Año académico</h1>
13          <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"
14          >
15            <span aria-hidden="true">&times;</span>
16          </button>
17        </div>
18        <div class="modal-body">
19          Confirme si desea eliminar el año académico {{ $anio->descripcion }}
20        </div>
21        <div class="modal-footer">
22          <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</
23          button>
24          <input type="submit" class="btn btn-danger btn-sm" value="Eliminar">
25        </div>
26      </div>
27    </form>
28  </div>
29 </div>
```

Nota. Código PHP de vista Gestionar Año Académico (modal.blade.php).

Figura 129: Vistas de Gestionar Año Académico (search.blade.php).



```
1 {!! Form::open(array('url'=>'admin/
2   apoderado','method'=>'GET','autocomplete'=>'off','role'=>'search')) !!}
3
4   <div class="form-group">
5     <div class="input-group">
6       <input type="text" class="form-control" name="searchText" placeholder="
7       Buscar..." value="{{ $searchText }}">
8       <span class="input-group-btn">
9         <button type="submit" class="btn btn-primary">Buscar</button>
10      </span>
11    </div>
12  </div>
13
14  {!! Form::close() !!}
```

Nota. Código PHP de vista Gestionar Año Académico (search.blade.php).

Figura 130: Vistas de Gestionar Secciones Grados (create.blade.php).

```

1  @extends('adminlte::page')
2
3  @section('title', 'grado')
4
5  @section('plugins.Sweetalert2',true)
6
7  @section('content_header')
8      <h2>Nuevo Grado</h2>
9      @if (count($errors)>0)
10         <div class="alert alert-danger">
11             <ul>
12                 @foreach($errors->all() as $errors)
13                     <li>{{ $errors }}</li>
14                 @endforeach
15             </ul>
16         </div>
17     @endif
18 @stop
19
20 @section('content')
21     <!-- Main content -->
22     <section class="content">
23         <div class="container-fluid">
24             <div class="row">
25                 <!-- left column -->
26                 <div class="col-md-12">
27                     <!-- jquery validation -->
28                     <div class="card card-primary">
29                         <div class="card-header">
30                             <h3 class="card-title"> <small> En este formulario se registrará los
31                                 distintos grados que tiene el colegio de acuerdo a un nivel académico</small>
32                             </div>
33                             <!-- form start -->
34                             <!-- Form: open(array('url'=>'admin/
35                                 grado','method'=>'POST','autocomplete'=>'off','files'=>true)) !!
36                             {!! Form::token() !!}
37                             <form id="quickForm">
38                                 <div class="card-body">
39                                     <div class="form-group">
40                                         <label for="grado">Grado:</label>
41                                         <input type="text" name="grado" class="form-control" id="
42                                             exampleInpEmail1" placeholder="Ingrese grado">
43                                     </div>
44                                     <!-- /.card-body -->
45                                     <div class="card-footer">
46                                         <button class="btn btn-primary" type="submit" id="btnOpenSaltB">Guardar
47                                         </button>
48                                         <button class="btn btn-danger" type="reset">Cancelar</button>
49                                     </div>
50                                 </div>
51                             <!-- /.card -->
52                             </div>
53                             <!-- /col (left) -->
54                             <!-- right column -->
55                             <div class="col-md-6">
56
57                             </div>
58                             <!-- /.col (right) -->
59                             </div>
60                             <!-- /.row -->
61                         </div><!-- /.container-fluid -->
62                     </section>
63                     <!-- /.content -->
64                     {!! Form::close() !!}
65                 @stop

```

Nota. Código PHP de vista Gestionar Secciones Grados (create.blade.php).

Figura 131: Vistas de Gestionar Secciones Grados (create.blade.php).

```

66
67 @section('css')
68     <link rel="stylesheet" href="/css/admin_custom.css">
69 @stop
70
71 @section('js')
72     <script> console.log('Hi!'); </script>
73     <script>
74     $(document).ready(function() {
75         $('#btnOpenSaltB').click(function() {
76             Swal.fire(
77                 'Buen trabajo!',
78                 'El grado ha sido registrado con éxito!',
79                 'success'
80             );
81         });
82
83         var Toast = Swal.mixin({
84             toast: true,
85             position: 'top-end',
86             showConfirmButton: false,
87             timer: 3000
88         });
89
90         $('#btnOpenSaltC').click(function() {
91             Toast.fire({
92                 icon: 'success',
93                 title: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr.'
94             });
95         });
96     });
97     </script>
98 @stop

```

Nota. Código PHP de vista Gestionar Secciones Grados (create.blade.php).

Figura 132: Vistas de Gestionar Secciones Grados (edit.blade.php).

```

1 @extends('adminite::page')
2
3 @section('title', 'Grado')
4
5 @section('content_header')
6 <h1>Editar Grado: {{ $gradoC->grado }}</h1>
7 @if (count($errors)>0)
8 <div class="alert alert-danger">
9 <ul>
10 @foreach($errors->all() as $errors)
11 <li>{{ $errors }}</li>
12 @endforeach
13 </ul>
14 </div>
15 @endif
16 @stop
17
18 @section('content')
19 <!-- Main content -->
20 <section class="content">
21 <div class="container-fluid">
22 <div class="row">
23 <!-- left column -->
24 <div class="col-md-12">
25 <!-- query validation -->
26 <div class="card card-primary">
27 <!-- /.card-header -->
28 <!-- form start -->
29 {!! Form::open(array('route' => ['grado.update', $gradoC->id_grado], 'method'
=> 'PUT', 'files' => true)) !!}
30 {!! Form::token() !!}
31 <form id="quickForm">
32 <div class="card-body">
33 <div class="form-group">
34 <label for="grado">Grado:</label>
35 <input type="text" name="grado" class="form-control" value="{{
$gradoC->grado }}">
36 </div>
37 </div>
38 </div>
39 <!-- /.card-body -->
40 <div class="card-footer">
41 <button class="btn btn-primary" type="submit">Guardar</button>
42 <button class="btn btn-danger" type="reset">Cancelar</button>
43 </div>
44 </form>
45 </div>
46 </div>
47 </div>
48 <!-- /col (left) -->
49 <!-- right column -->
50 <div class="col-md-6">
51 </div>
52 </div>
53 </div>
54 </div>
55 <!-- /.row -->
56 </div>
57 </section>
58 <!-- /.content -->
59 @stop
60
61 @section('css')
62 <link rel="stylesheet" href="/css/admin_custom.css">
63 @stop
64
65 @section('js')
66 <script> console.log('Hi!'); </script>
67 @stop
68

```

Nota. Código PHP de vista Gestionar Secciones Grados (edit.blade.php).

Figura 133: Vistas de Gestionar Secciones Grados (index.blade.php).

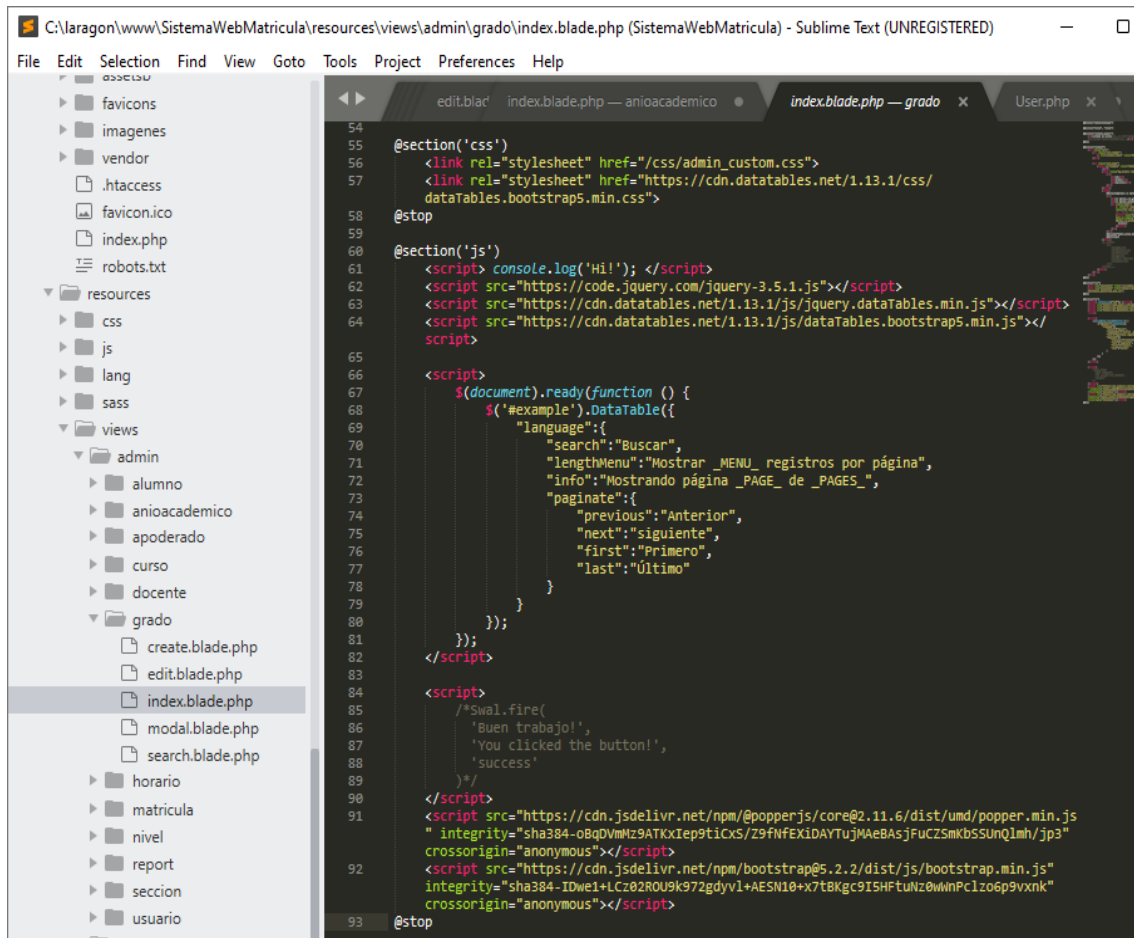
```

1 @extends('adminite::page')
2
3 @section('title', 'Grado')
4
5 @section('content_header')
6 <h1>Listado de grados <a href="apoderado/create"><button class="btn
btn-success">Nuevo</button></a></h1>
7 @stop
8
9 @section('content')
10 <div class="card">
11 <div class="card-header">
12 <a href="grado/create" class="btn btn-primary">Nuevo Grado</a>
13 </div>
14 <div class="card-body">
15 <table id="example" class="table table-striped shadow-lg mt-4" style="width:
100%;>
16 <thead>
17 <thead class="bg-primary text-white">
18 <tr>
19 <th>Id</th>
20 <th>Grado</th>
21 <th>Opciones</th>
22 </tr>
23 </thead>
24 <tbody>
25 @foreach($gradoA as $grado)
26 <tr>
27 <td>{{ $grado->id_grado }}</td>
28 <td>{{ $grado->grado }}</td>
29 <td width="200px">
30 <a href="{{ route('grado.edit', $grado->id_grado) }}"><button
class="btn btn-primary btn-sm">Editar</button></a>
31 <!-- Button trigger modal -->
32 <button type="button" class="btn btn-danger btn-sm"
data-bs-toggle="modal" data-bs-target="#modal-delete-{{
$grado->id_grado }}">
33 Eliminar
34 </button>
35 </td>
36 </tr>
37 @endforeach
38 @include('admin.grado.modal')
39 @endforeach
40 </tbody>
41 <!--tfoot-->
42 <tr>
43 <th>Name</th>
44 <th>Position</th>
45 <th>Office</th>
46 <th>Age</th>
47 <th>Start date</th>
48 <th>Salary</th>
49 </tr>
50 </tfoot>
51 </tbody>
52 </table>
53 </div>
54 @stop
55

```

Nota. Código PHP de vista Gestionar Secciones Grados (index.blade.php).

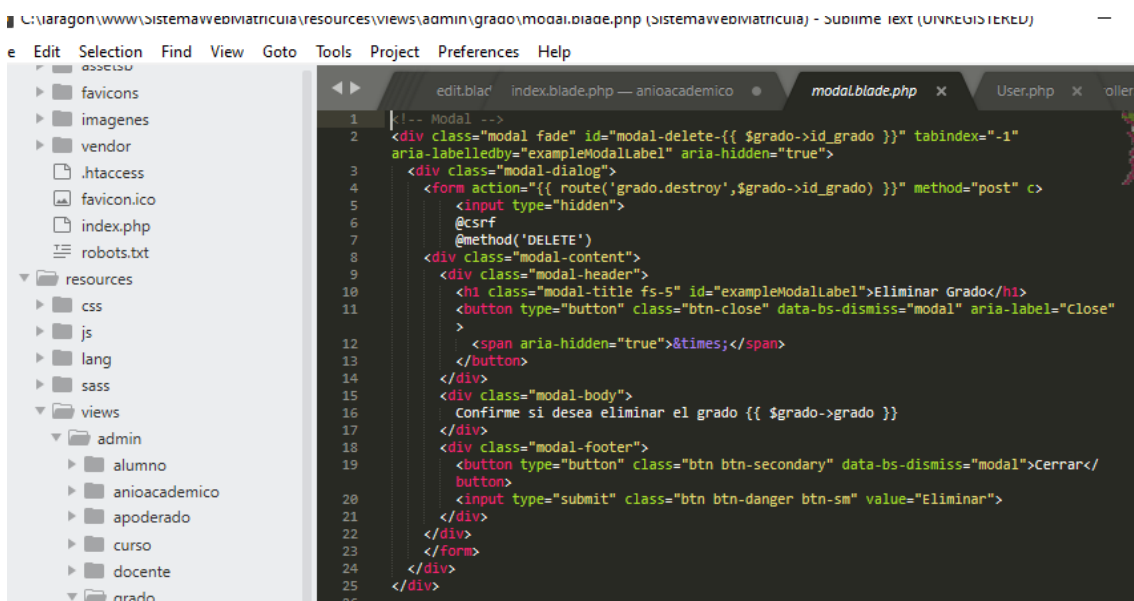
Figura 134: Vistas de Gestionar Secciones Grados (index.blade.php).



```
54 @section('css')
55 <link rel="stylesheet" href="/css/admin_custom.css">
56 <link rel="stylesheet" href="https://cdn.datatables.net/1.13.1/css/
57   dataTables.bootstrap5.min.css">
58 @stop
59
60 @section('js')
61 <script> console.log('Hi!'); </script>
62 <script src="https://code.jquery.com/jquery-3.5.1.js"></script>
63 <script src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>
64 <script src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap5.min.js"></
65   script>
66
67 <script>
68   $(document).ready(function () {
69     $('#example').DataTable({
70       "language": {
71         "search": "Buscar",
72         "lengthMenu": "Mostrar _MENU_ registros por página",
73         "info": "Mostrando página _PAGE_ de _PAGES_",
74         "paginate": {
75           "previous": "Anterior",
76           "next": "siguiente",
77           "first": "Primero",
78           "last": "Último"
79         }
80       }
81     });
82   });
83 </script>
84
85 <script>
86   /*Swal.fire(
87     'Buen trabajo!',
88     'You clicked the button!',
89     'success'
90   );*/
91 </script>
92 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js
93   integrity="sha384-oBqDVmWz9ATKXIp99tCs/Z9fNfEXiDAYTujMAeBAsjFUCZ5mkBSSunQlmh/jp3"
94   crossorigin="anonymous"></script>
95 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
96   integrity="sha384-IDwe1+LCz02R09k972gyv1+AESN10+x7tBKgc9ISHTFtUNz0WmPcLz06p9vxnk"
97   crossorigin="anonymous"></script>
98
99 @stop
```

Nota. Código PHP de vista Gestionar Secciones Grados (index.blade.php).

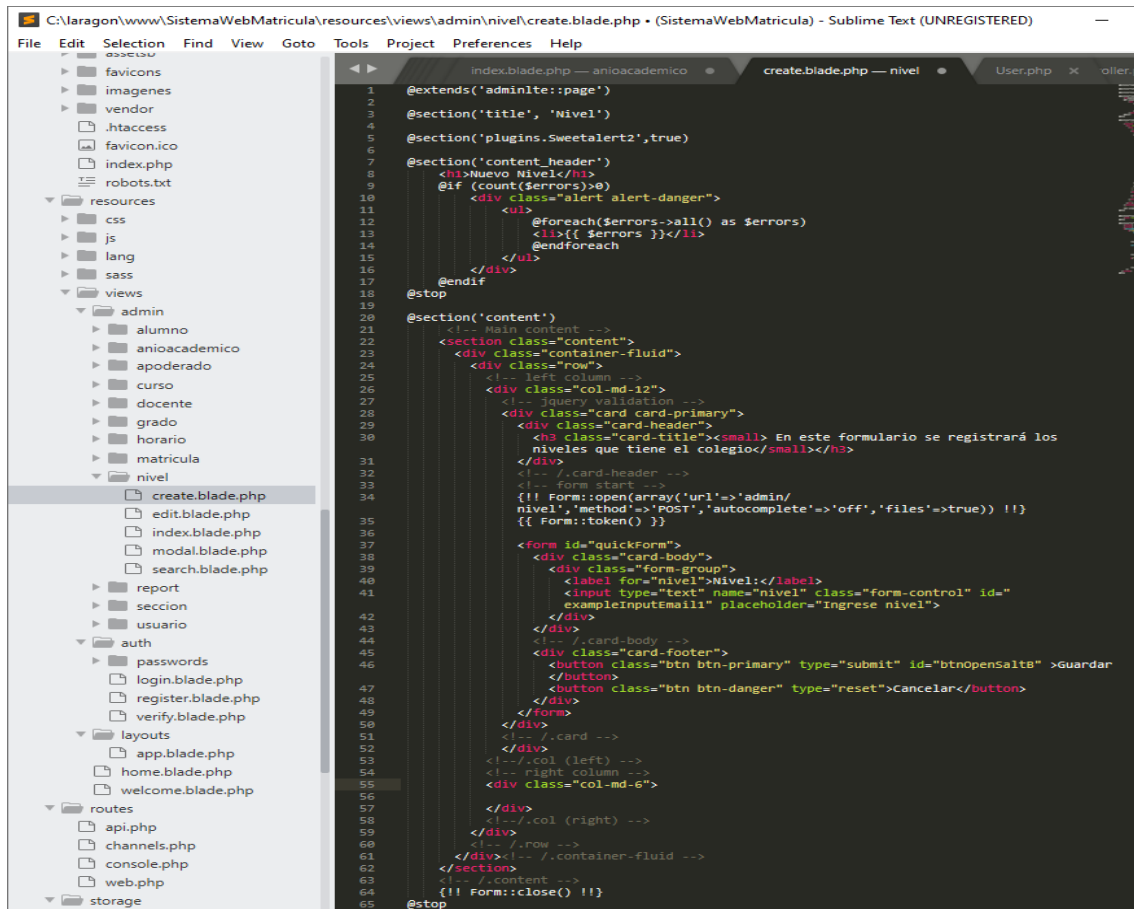
Figura 135: Vistas de Gestionar Secciones Grados (modal.blade.php).



```
1 <!-- Modal -->
2 <div class="modal fade" id="modal-delete-{{ $grado->id_grado }}" tabindex="-1"
3   aria-labelledby="exampleModalLabel" aria-hidden="true">
4   <div class="modal-dialog">
5     <form action="{{ route('grado.destroy', $grado->id_grado) }}" method="post" c>
6       <input type="hidden">
7       @csrf
8       @method('DELETE')
9     <div class="modal-content">
10      <div class="modal-header">
11        <h1 class="modal-title fs-5" id="exampleModalLabel">Eliminar Grado</h1>
12        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close">
13      </div>
14      <div class="modal-body">
15        <span aria-hidden="true">&times;</span>
16      </div>
17      <div class="modal-body">
18        Confirme si desea eliminar el grado {{ $grado->grado }}
19      </div>
20      <div class="modal-footer">
21        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</
22        button>
23        <input type="submit" class="btn btn-danger btn-sm" value="Eliminar">
24      </div>
25    </div>
26  </div>
```

Nota. Código PHP de vista Gestionar Secciones Grados (modal.blade.php).

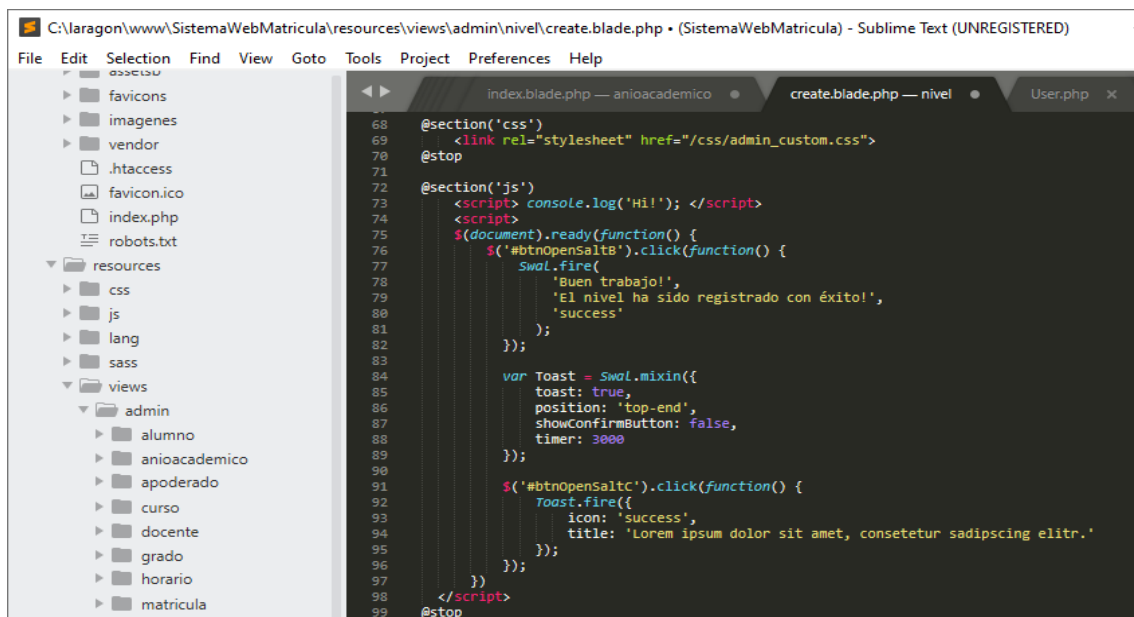
Figura 136: Vistas de Gestionar Secciones Nivel (create.blade.php).



```
1 @extends('adminite::page')
2
3 @section('title', 'Nivel')
4
5 @section('plugins.Sweetalert2', true)
6
7 @section('content_header')
8 <!--Nuevo Nivel-->
9 @if (count($errors)>0)
10 <div class="alert alert-danger">
11 <ul>
12 @foreach($errors->all() as $errors)
13 <li>{{ $errors }}</li>
14 @endforeach
15 </ul>
16 </div>
17 @endif
18 @stop
19
20 @section('content')
21 <!-- Main content -->
22 <section class="content">
23 <div class="container-fluid">
24 <div class="row">
25 <!-- left column -->
26 <div class="col-md-12">
27 <!-- jquery validation -->
28 <div class="card card-primary">
29 <div class="card-header">
30 <h3 class="card-title"><small> En este formulario se registrará los
31 niveles que tiene el colegio</small></h3>
32 </div>
33 <!-- Form start -->
34 {!! Form::open(array('url'=>'admin/
35 nivel', 'method'=>'POST', 'autocomplete'=>'off', 'files'=>true)) !!}
36 {!! Form::token() !!}
37
38 <div id="quickForm">
39 <div class="form-group">
40 <label for="nivel">Nivel:</label>
41 <input type="text" name="nivel" class="form-control" id="
42 exampleInputEmail" placeholder="Ingrese nivel">
43 </div>
44 </div>
45 <div class="card-body">
46 <div class="card-footer">
47 <button class="btn btn-primary" type="submit" id="btnOpenSalts">Guardar
48 </button>
49 <button class="btn btn-danger" type="reset">Cancelar</button>
50 </div>
51 </div>
52 </div>
53 <!-- /col (left) -->
54 <!-- right column -->
55 <div class="col-md-6">
56 </div>
57 </div>
58 <!-- /col (right) -->
59 </div>
60 <!-- /row -->
61 </div><!-- /.container-fluid -->
62 </section>
63 <!-- /content -->
64 {!! Form::close() !!}
65 @stop
```

Nota. Código PHP de vista Gestionar Secciones Nivel (create.blade.php).

Figura 137: Vistas de Gestionar Secciones Nivel (create.blade.php).



```
68 @section('css')
69 <link rel="stylesheet" href="/css/admin_custom.css">
70 @stop
71
72 @section('js')
73 <script> console.log('Hi!'); </script>
74 <script>
75 $(document).ready(function() {
76 $('#btnOpenSaltb').click(function() {
77 Swal.fire({
78 'Buen trabajo!',
79 'El nivel ha sido registrado con éxito!',
80 'success'
81 });
82 });
83
84 var Toast = Swal.mixin({
85 toast: true,
86 position: 'top-end',
87 showConfirmButton: false,
88 timer: 3000
89 });
90
91 $('#btnOpenSaltc').click(function() {
92 Toast.fire({
93 icon: 'success',
94 title: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr.'
95 });
96 });
97 </script>
98 @stop
```

Nota. Código PHP de vista Gestionar Secciones Nivel(create.blade.php).

Figura 138: Vistas de Gestionar Secciones Nivel (edit.blade.php).

```

1 @extends('adminlte::page')
2
3 @section('title', 'Nivel')
4
5 @section('content_header')
6 <h1>editar Nivel: {{ $nivel->nivel }}</h1>
7 @if (count($errors)>0)
8 <div class="alert alert-danger">
9 <ul>
10 @foreach($errors->all() as $errors)
11 <li>{{ $errors }}</li>
12 @endforeach
13 </ul>
14 @endif
15 @stop
16
17 @section('content')
18 <!-- Main content -->
19 <section class="content">
20 <div class="container-fluid">
21 <div class="row">
22 <!-- left column -->
23 <div class="col-md-12">
24 <!-- jquery validation -->
25 <div class="card card-primary">
26 <!-- /.card-header -->
27 <!-- form start -->
28 {!! Form::open(array('route' => ['nivel.update',$nivel->id_nivel],'method'
29 => 'PUT','files'=>true) !!)
30 {!! Form::token() !!}
31
32 <form id="quickForm">
33 <div class="card-body">
34 <div class="form-group">
35 <label for="grado">Nivel:</label>
36 <input type="text" name="nivel" class="form-control" value="{{
37 $nivel->nivel }}">
38 </div>
39 <!-- /.card-body -->
40 <div class="card-footer">
41 <button class="btn btn-primary" type="submit">Guardar</button>
42 <button class="btn btn-danger" type="reset">Cancelar</button>
43 </div>
44 </form>
45 <!-- /.card -->
46 </div>
47 <!-- /.col (left) -->
48 <!-- right column -->
49 <div class="col-md-6">
50 </div>
51 </div>
52 <!-- /.col (right) -->
53 <!-- /.row -->
54 </div><!-- /.container-fluid -->
55 </section><!-- /.content -->
56 {!! Form::close() !!}
57 @stop
58
59 @section('css')
60 <link rel="stylesheet" href="/css/admin_custom.css">
61 @stop
62
63 @section('js')
64 <script> console.log('Hi!'); </script>
65 @stop
66
67
68

```

Nota. Código PHP de vista Gestionar Secciones Nivel (edit.blade.php).

Figura 139: Vistas de Gestionar Secciones Nivel (index.blade.php).

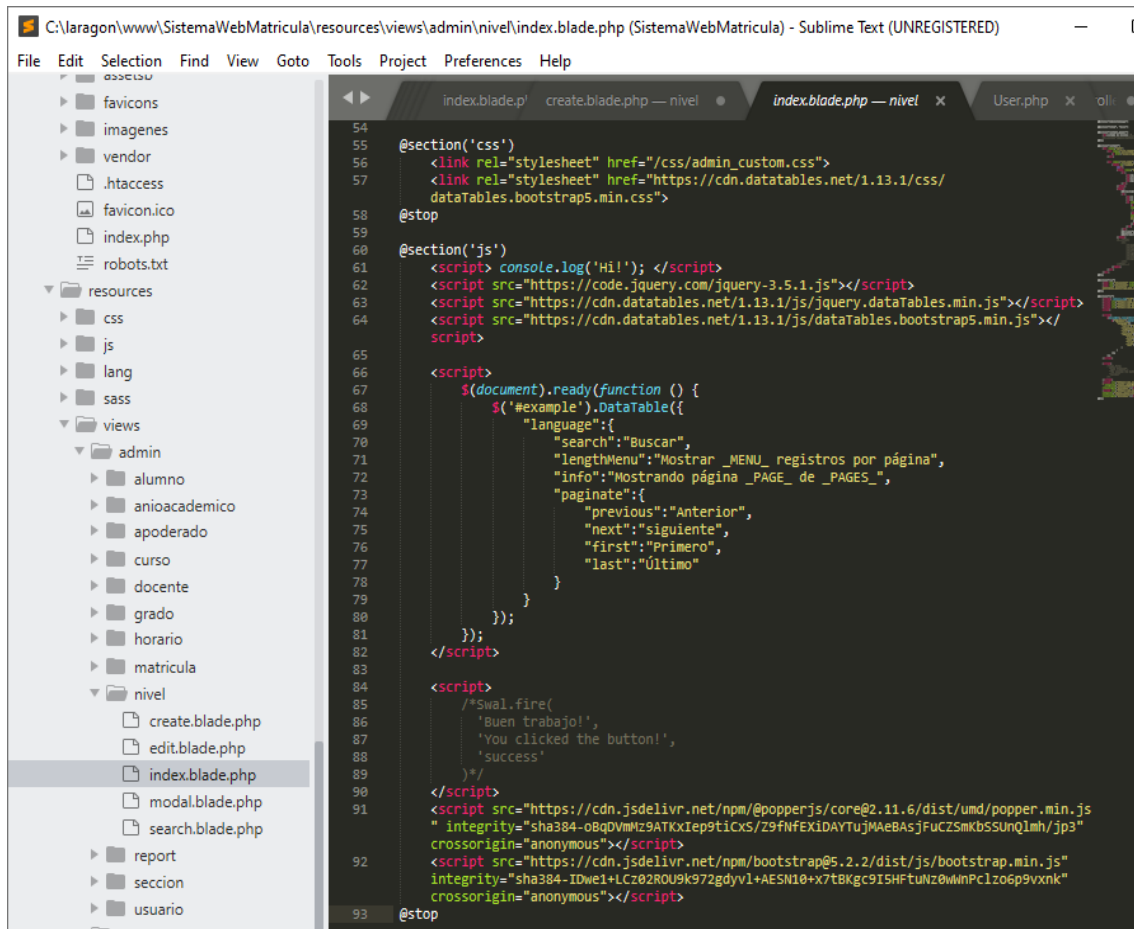
```

1 @extends('adminlte::page')
2
3 @section('title', 'Nivel')
4
5 @section('content_header')
6 <h1>listado de Niveles <a href="/apoderado/create"><button class="btn
7 btn-primary">Nuevo Nivel</a></h1>
8 @stop
9
10 @section('content')
11 <div class="card">
12 <div class="card-header">
13 <h2><a href="/nivel/create" class="btn btn-primary">Nuevo Nivel</a>
14 </div>
15 <div class="card-body">
16 <table id="example" class="table table-striped shadow-lg mt-4" style="width:
17 100%;>
18 <thead class="bg-primary text-white">
19 <tr>
20 <th>id</th>
21 <th>nivel</th>
22 <th>Opciones</th>
23 </tr>
24 </thead>
25 <tbody>
26 @foreach($niveles as $nivel)
27 <tr>
28 <td>{{ $nivel->id_nivel }}</td>
29 <td>{{ $nivel->nivel }}</td>
30 <td>
31 <a href="{{ route('nivel.edit',$nivel->id_nivel) }}"><button
32 class="btn btn-primary btn-sm">editar</button></a>
33 <button type="button" class="btn btn-danger btn-sm"
34 data-bs-toggle="modal" data-bs-target="#modal-delete-{{
35 $nivel->id_nivel }}">
36 </button>
37 </td>
38 </tr>
39 </tbody>
40 </table>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 @stop

```

Nota. Código PHP de vista Gestionar Secciones Nivel (index.blade.php).

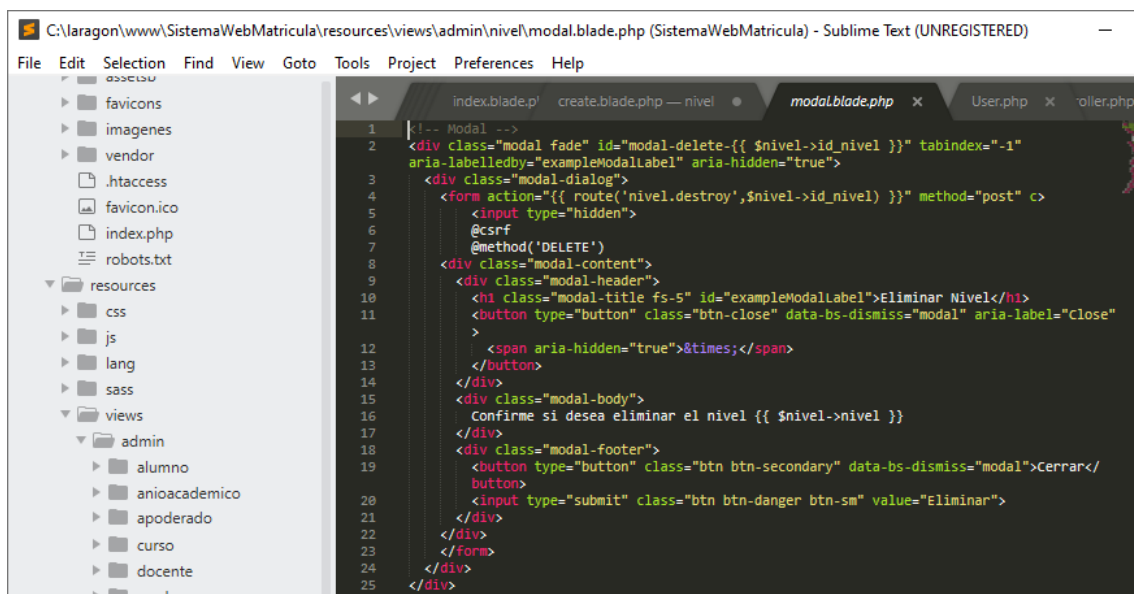
Figura 140: Vistas de Gestionar Secciones Nivel (*index.blade.php*).



```
54
55 @section('css')
56 <link rel="stylesheet" href="/css/admin_custom.css">
57 <link rel="stylesheet" href="https://cdn.datatables.net/1.13.1/css/
58   dataTables.bootstrap5.min.css">
59 @stop
60
61 @section('js')
62 <script> console.log('Hi!'); </script>
63 <script src="https://code.jquery.com/jquery-3.5.1.js"></script>
64 <script src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>
65 <script src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap5.min.js"></
66   script>
67
68 <script>
69   $(document).ready(function () {
70     $('#example').DataTable({
71       "language": {
72         "search": "Buscar",
73         "lengthMenu": "Mostrar MENU registros por página",
74         "info": "Mostrando página _PAGE_ de _PAGES_",
75         "paginate": {
76           "previous": "Anterior",
77           "next": "siguiente",
78           "first": "Primero",
79           "last": "Último"
80         }
81       });
82     });
83   </script>
84
85   <script>
86     /*Swal.fire(
87       'Buen trabajo!',
88       'You clicked the button!',
89       'success'
90     );*/
91   </script>
92   <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
93     integrity="sha384-08QDMZ9ATKxIep9tiCxs/Z9fnEXiDAYTujMAeBAsjFucZSmkbSSunQ1mh/jp3"
94     crossorigin="anonymous"></script>
95   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
96     integrity="sha384-IDwe1+LCz02ROU9k972gdYv1+AESN10+xt7tBKGc915HFTunz0winPcLz06p9vxnk"
97     crossorigin="anonymous"></script>
98
99 @stop
```

Nota. Código PHP de vista Gestionar Secciones Nivel (*index.blade.php*).

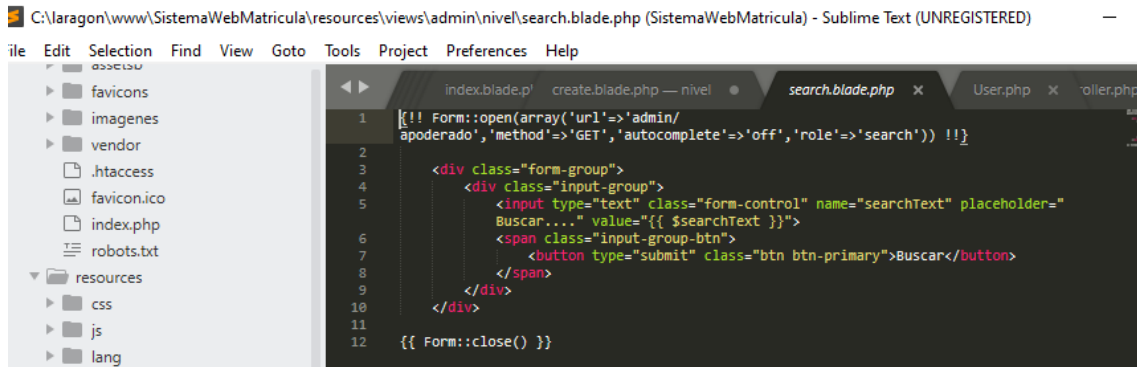
Figura 141: Vistas de Gestionar Secciones Nivel (*modal.blade.php*).



```
1 |<!-- Modal -->
2 <div class="modal fade" id="modal-delete-{{ $nivel->id_nivel }}" tabindex="-1"
3   aria-labelledby="exampleModallabel" aria-hidden="true">
4   <div class="modal-dialog">
5     <form action="{{ route('nivel.destroy', $nivel->id_nivel) }}" method="post" c>
6       <input type="hidden">
7       @csrf
8       @method('DELETE')
9     <div class="modal-content">
10      <div class="modal-header">
11        <h1 class="modal-title fs-5" id="exampleModallabel">Eliminar Nivel</h1>
12        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close">
13          <span aria-hidden="true">&times;</span>
14        </button>
15      </div>
16      <div class="modal-body">
17        Confirme si desea eliminar el nivel {{ $nivel->nivel }}
18      </div>
19      <div class="modal-footer">
20        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</
21        button>
22        <input type="submit" class="btn btn-danger btn-sm" value="Eliminar">
23      </div>
24    </form>
25  </div>
```

Nota. Código PHP de vista Gestionar Secciones Nivel (*modal.blade.php*).

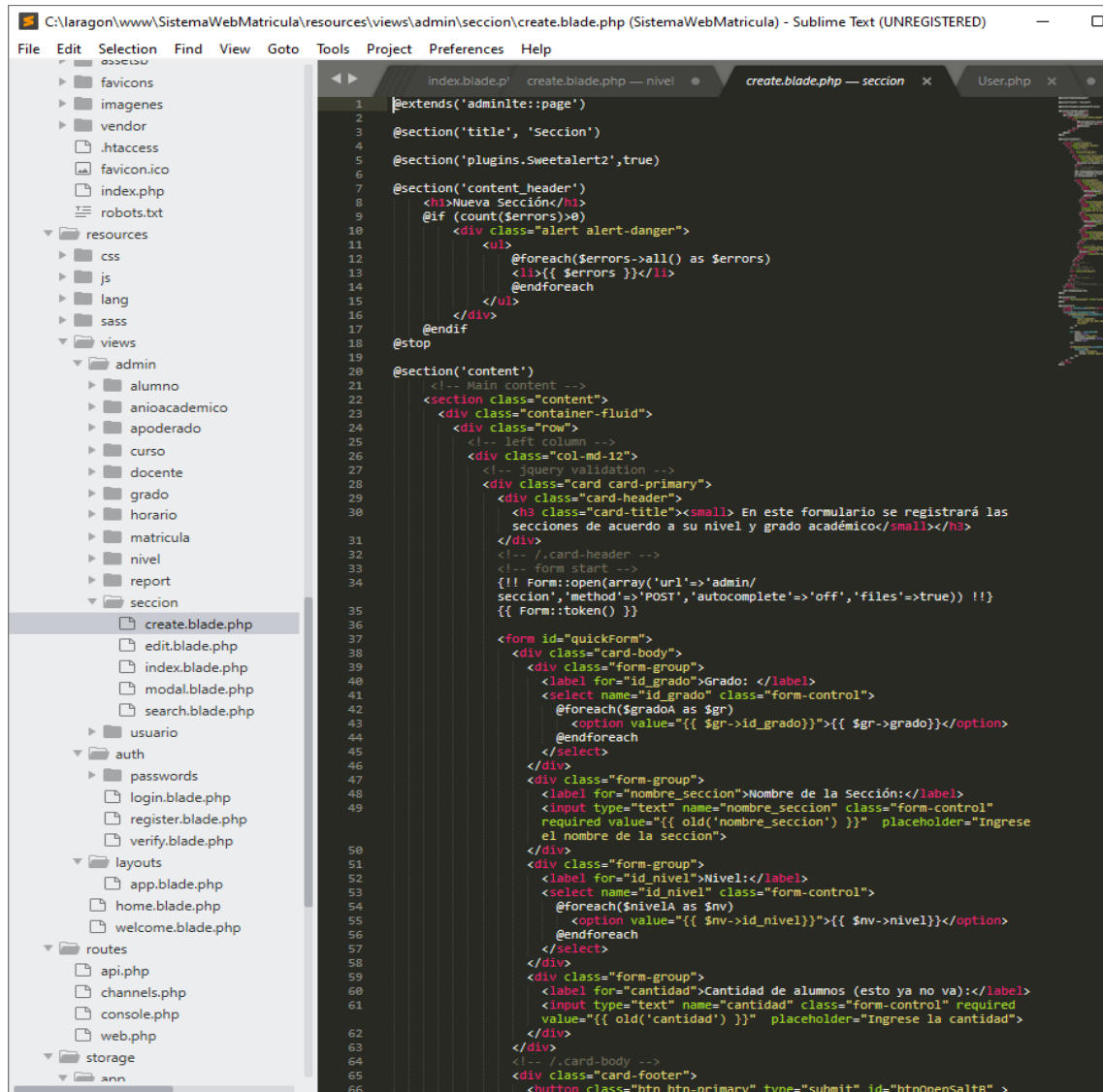
Figura 142: Vistas de Gestionar Secciones Nivel (search.blade.php).



```
1  {!! Form::open(array('url'=>'admin/
apoderado', 'method'=>'GET', 'autocomplete'=>'off', 'role'=>'search')) !!}
2
3  <div class="form-group">
4  <div class="input-group">
5  <input type="text" class="form-control" name="searchText" placeholder="
Buscar..." value="{{ $searchText }}"
6  <span class="input-group-btn">
7  <button type="submit" class="btn btn-primary">Buscar</button>
8  </span>
9  </div>
10 </div>
11
12 {!! Form::close() !!}
```

Nota. Código PHP de vista Gestionar Secciones Nivel (search.blade.php).

Figura 143: Vistas de Gestionar Secciones Sección (create.blade.php).



```
1  @extends('adminlte:page')
2
3  @section('title', 'Sección')
4
5  @section('plugins.Sweetalert2', true)
6
7  @section('content_header')
8  <h1>Nueva Sección</h1>
9  @if (count($errors)>0)
10 <div class="alert alert-danger">
11 <ul>
12 @foreach($errors->all() as $errors)
13 <li>{{ $errors }}</li>
14 @endforeach
15 </ul>
16 </div>
17 @endif
18 @stop
19
20 @section('content')
21 <!-- Main content -->
22 <section class="content">
23 <div class="container-fluid">
24 <div class="row">
25 <!-- left column -->
26 <div class="col-md-12">
27 <!-- jQuery validation -->
28 <div class="card card-primary">
29 <div class="card-header">
30 <h3 class="card-title"><small>En este formulario se registrará las
secciones de acuerdo a su nivel y grado académico</small></h3>
31 </div>
32 <!-- /.card-header -->
33 <!-- form start -->
34 {!! Form::open(array('url'=>'admin/
seccion', 'method'=>'POST', 'autocomplete'=>'off', 'files'=>true)) !!}
35 {!! Form::token() !!}
36
37 <form id="quickForm">
38 <div class="card-body">
39 <div class="form-group">
40 <label for="id_grado">Grado: </label>
41 <select name="id_grado" class="form-control">
42 @foreach($gradoA as $gr)
43 <option value="{{ $gr->id_grado }}">{{ $gr->grado }}</option>
44 @endforeach
45 </select>
46 </div>
47 <div class="form-group">
48 <label for="nombre_seccion">Nombre de la Sección:</label>
49 <input type="text" name="nombre_seccion" class="form-control"
required value="{{ old('nombre_seccion') }}" placeholder="Ingrese
el nombre de la seccion">
50 </div>
51 <div class="form-group">
52 <label for="id_nivel">Nivel:</label>
53 <select name="id_nivel" class="form-control">
54 @foreach($nivelA as $nv)
55 <option value="{{ $nv->id_nivel }}">{{ $nv->nivel }}</option>
56 @endforeach
57 </select>
58 </div>
59 <div class="form-group">
60 <label for="cantidad">Cantidad de alumnos (esto ya no va):</label>
61 <input type="text" name="cantidad" class="form-control" required
value="{{ old('cantidad') }}" placeholder="Ingrese la cantidad">
62 </div>
63 <!-- /.card-body -->
64 <div class="card-footer">
65 <button class="btn btn-primary" type="submit" id="btnopensalt8" >
```

Nota. Código PHP de vista Gestionar Secciones Sección (create.blade.php).

Figura 144: Vistas de Gestionar Secciones Sección (create.blade.php).

```

68         Guardar</button>
69         <button class="btn btn-danger" type="reset">cancelar</button>
70     </div>
71 </form>
72 </div>
73 <!-- /.card -->
74 </div>
75 <!-- /.col (left) -->
76 <!-- right column -->
77 <div class="col-md-6">
78
79 </div>
80 <!-- /.col (right) -->
81 </div>
82 <!-- /.row -->
83 </div><!-- /.container-fluid -->
84 </section>
85 <!-- /.content -->
86 {!! Form::close() !!}
87 @stop
88
89 @section('css')
90     <link rel="stylesheet" href="/css/admin_custom.css">
91 @stop
92
93 @section('js')
94     <script> console.log('Hi!'); </script>
95     <script>
96     $(document).ready(function() {
97         $('#btnOpenSaltb').click(function() {
98             Swal.fire({
99                 'Buena trabajo!',
100                'La Sección ha sido registrada con éxito!',
101                'success'
102            });
103        });
104
105        var Toast = Swal.mixin({
106            toast: true,
107            position: 'top-end',
108            showconfirmButton: false,
109            timer: 3000
110        });
111
112        $('#btnOpenSaltc').click(function() {
113            Toast.fire({
114                icon: 'success',
115                title: 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr.'
116            });
117        });
118    });
119 </script>
120 @stop
    
```

Nota. Código PHP de vista Gestionar Secciones Sección (create.blade.php).

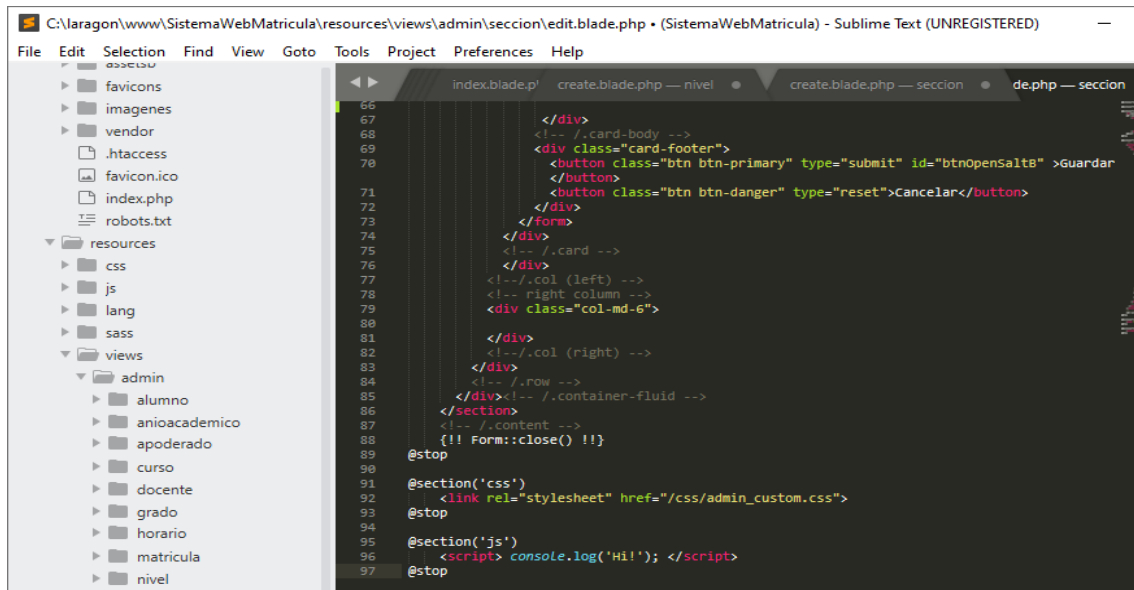
Figura 145: Vistas de Gestionar Secciones Sección (edit.blade.php).

```

1  @extends('adminlte::page')
2
3  @section('title', 'Seccion')
4
5  @section('content_header')
6      <h2>Editar Seccion: {{ $seccion->nombre_seccion }}</h2>
7  @if (count($errors) > 0)
8      <div class="alert alert-danger">
9          <ul>
10             @foreach($errors->all() as $errors)
11                 <li>{{ $errors }}</li>
12             @endforeach
13         </ul>
14     </div>
15 @endif
16 @stop
17
18 @section('content')
19     <!-- Main content -->
20     <section class="content">
21         <div class="container-fluid">
22             <div class="row">
23                 <!-- left column -->
24                 <div class="col-md-12">
25                     <!-- jquery validation -->
26                     <div class="card card-primary">
27                         <!-- /.card-header -->
28                         <!-- form start -->
29                         {!! Form::open(array('route' => 'PUT', 'files' => true)) !!}
30                         {!! Form::token() !!}
31
32                         <form id="quickForm">
33                             <div class="card-body">
34                                 <div class="form-group">
35                                     <label for="id_grado">Grado:</label>
36                                     <select name="id_grado" class="form-control">
37                                         @foreach($gradoA as $gr)
38                                             @if($gr->id_grado==$seccion->id_seccion)
39                                                 <option value="{{ $gr->id_grado }}" selected="">{{ $gr->grado}}</option>
40                                             @else
41                                                 <option value="{{ $gr->id_grado }}">{{ $gr->grado}}</option>
42                                             @endif
43                                         @endforeach
44                                     </select>
45                                 </div>
46                                 <div class="form-group">
47                                     <label for="nombre_seccion">Nombre de la Sección:</label>
48                                     <input type="text" name="nombre_seccion" class="form-control" required
49                                         value="{{ $seccion->nombre_seccion }}">
50                                 </div>
51                                 <div class="form-group">
52                                     <label for="id_nivel">Nivel:</label>
53                                     <select name="id_nivel" class="form-control">
54                                         @foreach($nivelA as $nv)
55                                             @if($nv->id_nivel==$seccion->id_seccion)
56                                                 <option value="{{ $nv->id_nivel }}" selected="">{{ $nv->nivel}}</option>
57                                             @else
58                                                 <option value="{{ $nv->id_nivel }}">{{ $nv->nivel}}</option>
59                                             @endif
60                                         @endforeach
61                                     </select>
62                                 </div>
63                                 <div class="form-group">
64                                     <label for="cantidad">Cantidad de alumnos (esto ya no va):</label>
65                                     <input type="text" name="cantidad" class="form-control" required value
66                                        ="{{ $seccion->cantidad }}">
67                                 </div>
        
```

Nota. Código PHP de vista Gestionar Secciones Sección (edit.blade.php).

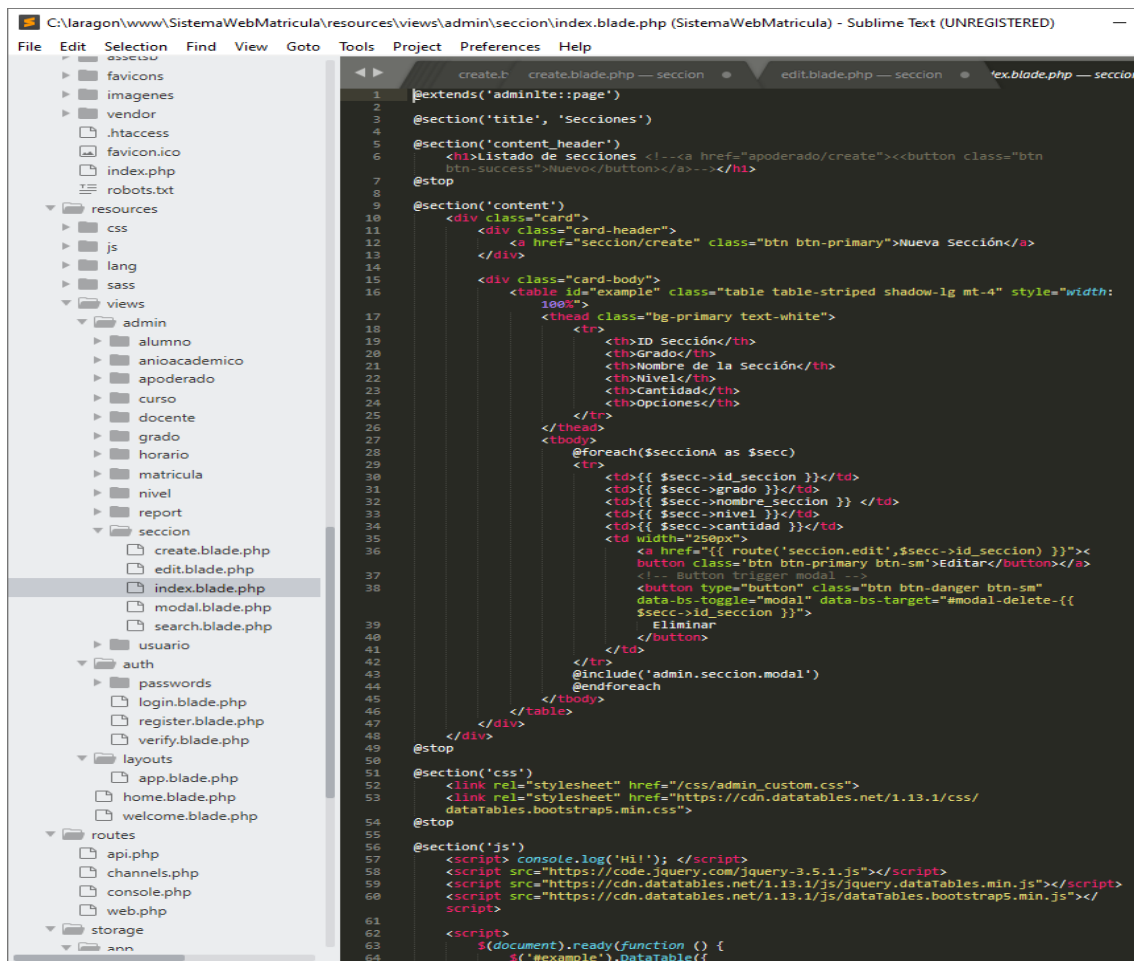
Figura 146: Vistas de Gestionar Secciones Sección (edit.blade.php).



```
66 </div>
67 <!-- /.card-body -->
68 <div class="card-footer">
69 <button class="btn btn-primary" type="submit" id="btnOpensaltB">Guardar
70 </button>
71 <button class="btn btn-danger" type="reset">Cancelar</button>
72 </div>
73 </div>
74 </div>
75 <!-- /.card -->
76 </div>
77 <!-- /.col (left) -->
78 <!-- right column -->
79 <div class="col-md-6">
80
81 </div>
82 <!-- /.col (right) -->
83 </div>
84 <!-- /.row -->
85 </div><!-- /.container-fluid -->
86 </section>
87 <!-- /.content -->
88 {!! Form::close() !!}
89 @stop
90
91 @section('css')
92 <link rel="stylesheet" href="/css/admin_custom.css">
93 @stop
94
95 @section('js')
96 <script> console.log('Hi!'); </script>
97 @stop
```

Nota. Código PHP de vista Gestionar Secciones Sección (edit.blade.php).

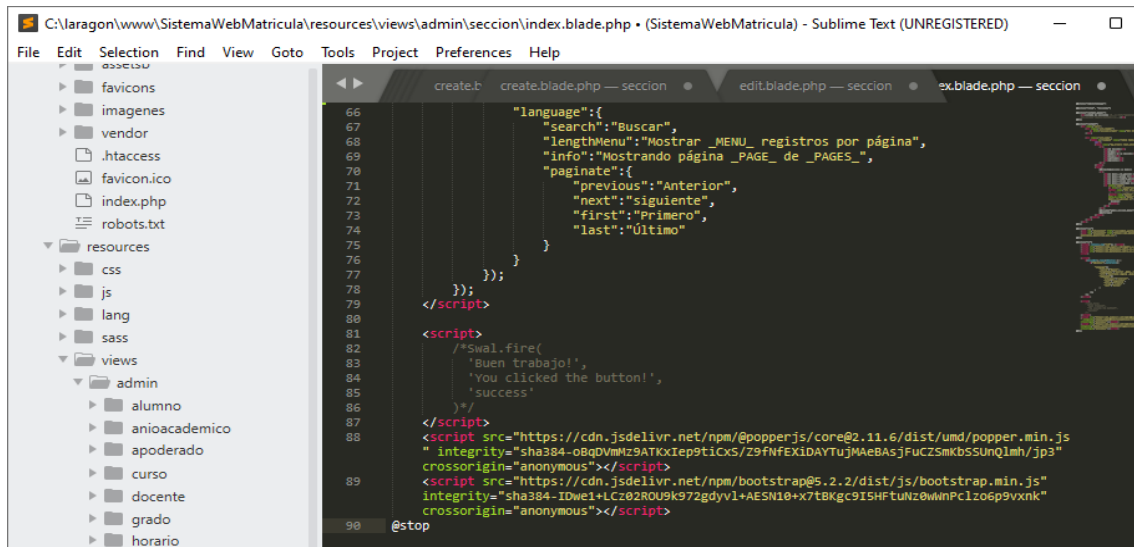
Figura 147: Vistas de Gestionar Secciones Sección (index.blade.php).



```
1 @extends('adminlte:page')
2
3 @section('title', 'Secciones')
4
5 @section('content_header')
6 <h1>Listado de secciones <!--<a href="apoderado/create"><button class="btn
7 btn-success">Nuevo</button></a--></h1>
8 @stop
9
10 @section('content')
11 <div class="card">
12 <div class="card-header">
13 <a href="seccion/create" class="btn btn-primary">Nueva Sección</a>
14 </div>
15 <div class="card-body">
16 <table id="example" class="table table-striped shadow-lg mt-4" style="width:
17 100%">
18 <thead class="bg-primary text-white">
19 <tr>
20 <th>ID Sección</th>
21 <th>Grado</th>
22 <th>Nombre de la Sección</th>
23 <th>Nivel</th>
24 <th>Cantidad</th>
25 <th>Opciones</th>
26 </tr>
27 </thead>
28 <tbody>
29 @foreach($seccionA as $secc)
30 <tr>
31 <td>{{ $secc->id_seccion }}</td>
32 <td>{{ $secc->grado }}</td>
33 <td>{{ $secc->nombre_seccion }}</td>
34 <td>{{ $secc->nivel }}</td>
35 <td>{{ $secc->cantidad }}</td>
36 <td width="250px">
37 <a href="{{ route('seccion.edit',$secc->id_seccion) }}"><button
38 class="btn btn-primary btn-sm">Editar</button></a>
39 <!-- button trigger modal -->
40 <button type="button" class="btn btn-danger btn-sm"
41 data-bs-toggle="modal" data-bs-target="#modal-delete-{{
42 $secc->id_seccion }}">
43 Eliminar
44 </button>
45 </td>
46 </tr>
47 @endforeach
48 @include('admin.seccion.modal')
49 @endforeach
50 </tbody>
51 </table>
52 @stop
53
54 @section('css')
55 <link rel="stylesheet" href="/css/admin_custom.css">
56 <link rel="stylesheet" href="https://cdn.datatables.net/1.13.1/css/
57 dataTables.bootstrap.min.css">
58 @stop
59
60 @section('js')
61 <script> console.log('Hi!'); </script>
62 <script src="https://code.jquery.com/jquery-3.5.1.js"></script>
63 <script src="https://cdn.datatables.net/1.13.1/js/jquery.dataTables.min.js"></script>
64 <script src="https://cdn.datatables.net/1.13.1/js/dataTables.bootstrap.min.js"></script>
65
66 <script>
67 $(document).ready(function () {
68 <!-- #example -->
69 $('#example').DataTable({
```

Nota. Código PHP de vista Gestionar Secciones Sección (index.blade.php).

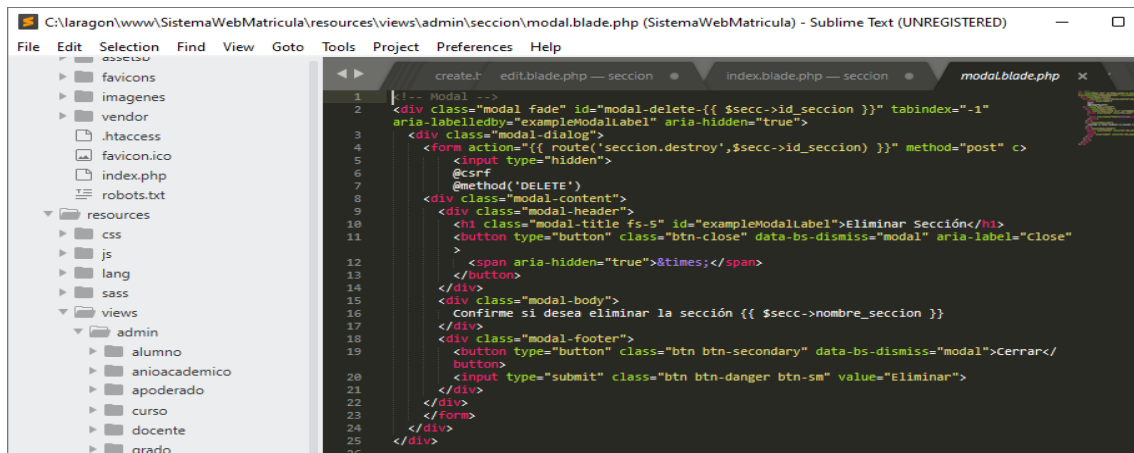
Figura 148: Vistas de Gestionar Secciones Sección (*index.blade.php*).



```
66         "language":{
67             "search":"Buscar",
68             "lengthMenu":"Mostrar _MENU_ registros por página",
69             "info":"Mostrando página _PAGE_ de _PAGES_",
70             "paginate":{
71                 "previous":"Anterior",
72                 "next":"siguiente",
73                 "first":"Primero",
74                 "last":"Último"
75             }
76         }
77     });
78 });
79 </script>
80
81 <script>
82     /*Swal.fire(
83         'Buen trabajo!',
84         'You clicked the button!',
85         'success'
86     );*/
87 </script>
88 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js
89     " integrity="sha384-oBqDVmz9ATKixep9tiCxS/Z9fNfExiDAYtjMAeBAsjFucZsmKbSSunQlhm/jp3"
90     crossorigin="anonymous"></script>
91 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.min.js"
92     integrity="sha384-IDwe1+LCz0R0U9K972gdyv1+AESN10+X7T8Kgc9ISHFtUz0W9PcLzo6p9vxnk"
93     crossorigin="anonymous"></script>
94 @stop
```

Nota. Código PHP de vista Gestionar Secciones Sección (*index.blade.php*).

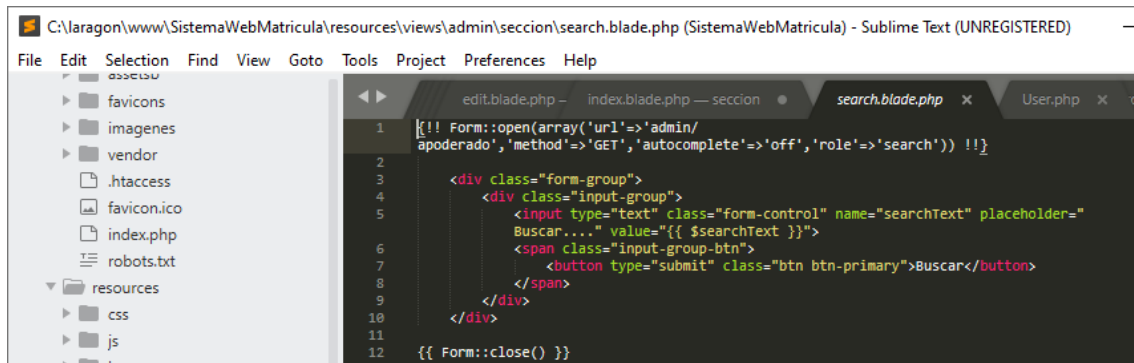
Figura 149: Vistas de Gestionar Secciones Sección (*modal.blade.php*).



```
1 <!-- Modal -->
2 <div class="modal fade" id="modal-delete-{{ $secc->id_seccion }}" tabindex="-1"
3     aria-labelledby="exampleModallabel" aria-hidden="true">
4     <div class="modal-dialog">
5         <form action="{{ route('seccion.destroy',$secc->id_seccion) }}" method="post" >
6             <input type="hidden">
7             @csrf
8             @method('DELETE')
9             <div class="modal-content">
10                <div class="modal-header">
11                    <h1 class="modal-title fs-5" id="exampleModallabel">Eliminar Sección</h1>
12                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close">
13                        <span aria-hidden="true">&times;</span></button>
14                </div>
15                <div class="modal-body">
16                    Confirme si desea eliminar la sección {{ $secc->nombre_seccion }}
17                </div>
18                <div class="modal-footer">
19                    <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cerrar</
20                    button>
21                    <input type="submit" class="btn btn-danger btn-sm" value="Eliminar">
22                </div>
23            </form>
24        </div>
25    </div>
26 </div>
```

Nota. Código PHP de vista Gestionar Secciones Sección (*modal.blade.php*).

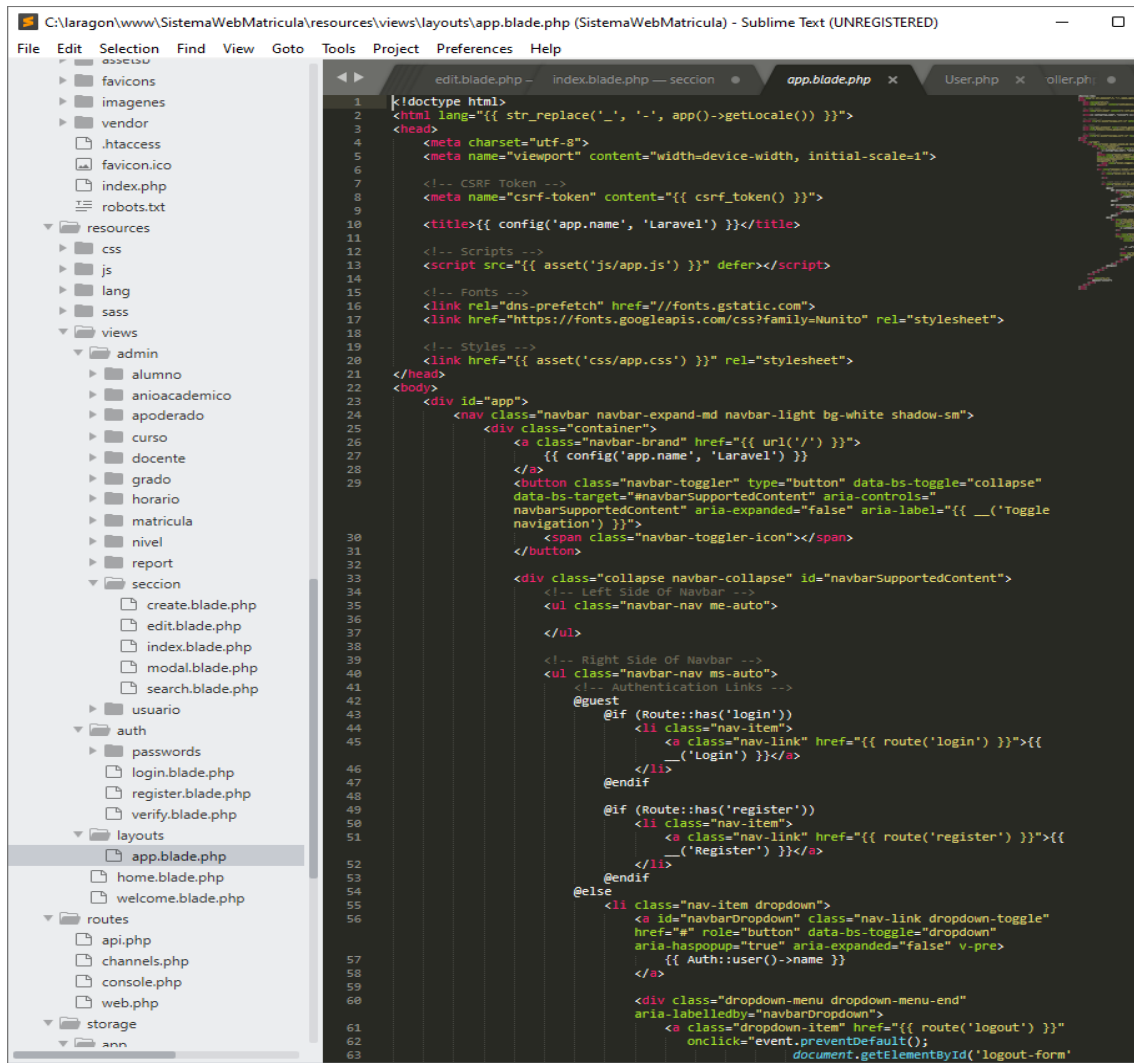
Figura 150: Vistas de Gestionar Secciones Sección (*search.blade.php*).



```
1 {!! Form::open(array('url'=>'admin/
2     apoderado','method'=>'GET','autocomplete'=>'off','role'=>'search')) !!}
3
4     <div class="form-group">
5         <div class="input-group">
6             <input type="text" class="form-control" name="searchText" placeholder="
7                 Buscar..." value="{{ $searchText }}">
8             <span class="input-group-btn">
9                 <button type="submit" class="btn btn-primary">Buscar</button>
10            </span>
11        </div>
12    </div>
13    {!! Form::close() !!}
```

Nota. Código PHP de vista Gestionar Secciones Sección (*search.blade.php*).

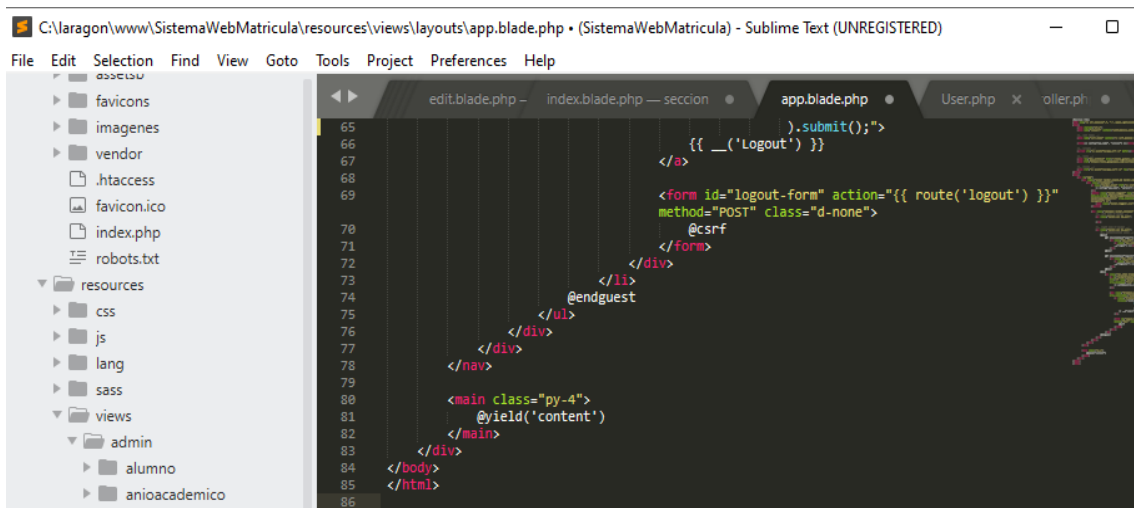
Figura 151: Vistas de Layouts (Diseños) (app.blade.php).



```
1 <!doctype html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6
7 <!-- CSRF Token -->
8 <meta name="csrf-token" content="{{ csrf_token() }}">
9
10 <title>{{ config('app.name', 'Laravel') }}</title>
11
12 <!-- Scripts -->
13 <script src="{{ asset('js/app.js') }}" defer</script>
14
15 <!-- Fonts -->
16 <link rel="dns-prefetch" href="//fonts.gstatic.com">
17 <link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">
18
19 <!-- Styles -->
20 <link href="{{ asset('css/app.css') }}" rel="stylesheet">
21 </head>
22 <body>
23 <div id="app">
24 <nav class="navbar navbar-expand-md navbar-light bg-white shadow-sm">
25 <div class="container">
26 <a class="navbar-brand" href="{{ url('/') }}">
27 {{ config('app.name', 'Laravel') }}
28 </a>
29 <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
30 data-bs-target="#navbarSupportedContent" aria-controls="
31 navbarSupportedContent" aria-expanded="false" aria-label="{{ __('Toggle
32 navigation') }}">
33 <span class="navbar-toggler-icon"></span>
34 </button>
35
36 <div class="collapse navbar-collapse" id="navbarSupportedContent">
37 <!-- Left Side Of Navbar -->
38 <ul class="navbar-nav me-auto">
39
40 <!-- Right Side Of Navbar -->
41 <ul class="navbar-nav ms-auto">
42 <!-- Authentication Links -->
43 @guest
44 @if (Route::has('login'))
45 <li class="nav-item">
46 <a class="nav-link" href="{{ route('login') }}">{{
47 __('Login') }}</a>
48 @endif
49 @if (Route::has('register'))
50 <li class="nav-item">
51 <a class="nav-link" href="{{ route('register') }}">{{
52 __('Register') }}</a>
53 @endif
54 @else
55 <li class="nav-item dropdown">
56 <a id="navbarDropdown" class="nav-link dropdown-toggle"
57 href="#" role="button" data-bs-toggle="dropdown"
58 aria-haspopup="true" aria-expanded="false" v-pre>
59 {{ Auth::user()->name }}
60 </a>
61 <div class="dropdown-menu dropdown-menu-end"
62 aria-labelledby="navbarDropdown">
63 <a class="dropdown-item" href="{{ route('logout') }}"
64 onclick="event.preventDefault();
65 document.getElementById('logout-form')
66 .submit();">
67 {{ __('Logout') }}
68 </a>
69 <form id="logout-form" action="{{ route('logout') }}"
70 method="POST" class="d-none">
71 @csrf
72 </form>
73 </div>
74 @endguest
75 </ul>
76 </div>
77 </nav>
78
79 <main class="py-4">
80 @yield('content')
81 </main>
82 </div>
83 </body>
84 </html>
```

Nota. Código PHP de Vistas de Layouts (Diseños) (app.blade.php).

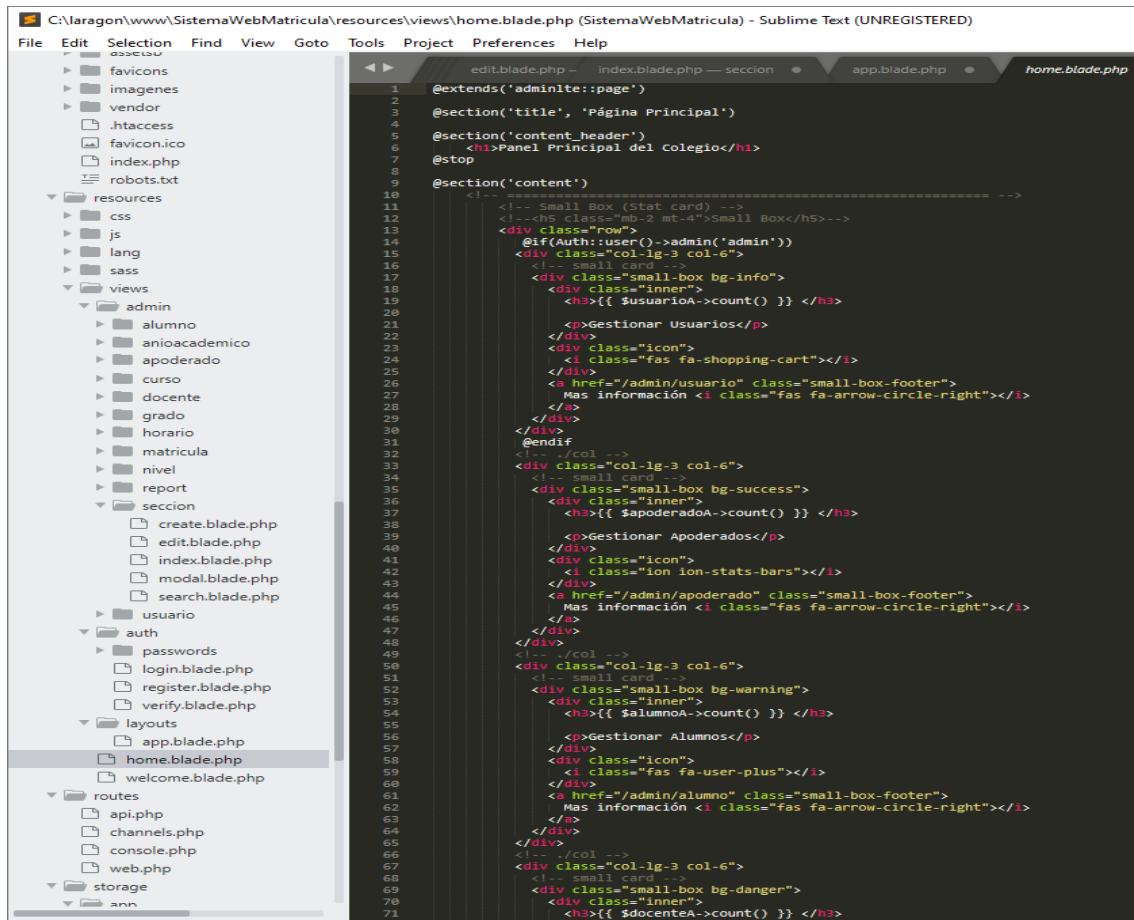
Figura 152: Vistas de Layouts (Diseños)(app.blade.php).



```
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 <form id="logout-form" action="{{ route('logout') }}"
71 method="POST" class="d-none">
72 @csrf
73 </form>
74 </div>
75 @endguest
76 </ul>
77 </div>
78 </nav>
79
80 <main class="py-4">
81 @yield('content')
82 </main>
83 </div>
84 </body>
85 </html>
86
```

Nota. Código PHP de Vistas de Layouts (Diseños) (app.blade.php).

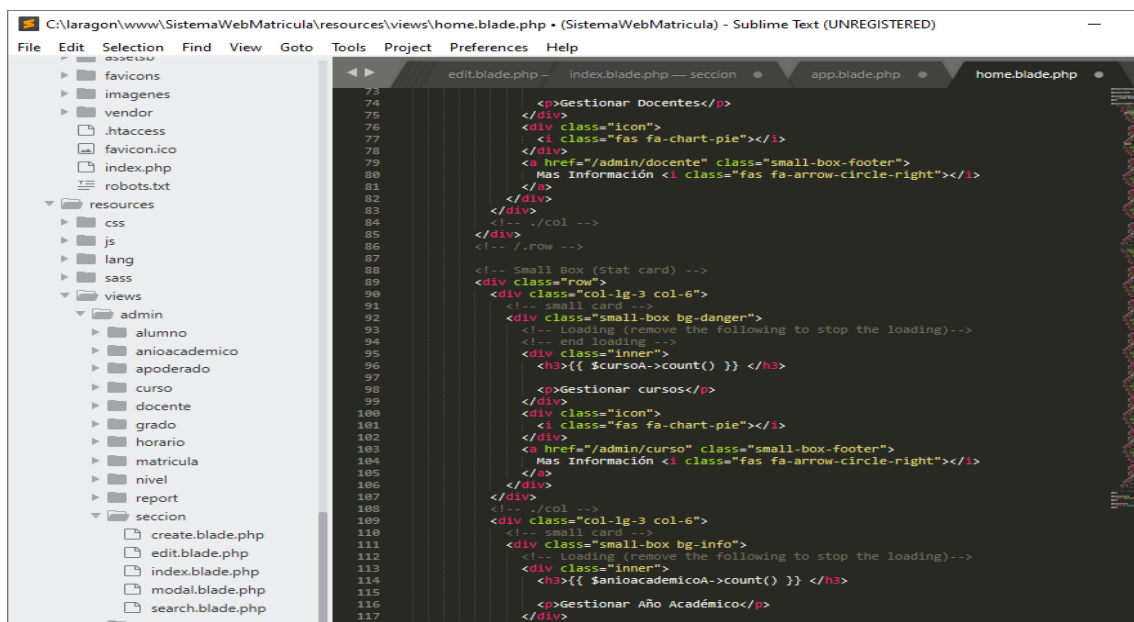
Figura 153: Vistas de Layouts (Diseños) (home.blade.php).



```
1 @extends('adminlte:page')
2
3 @section('title', 'Página Principal')
4
5 @section('content_header')
6 <h1>Panel Principal del Colegio</h1>
7 @stop
8
9 @section('content')
10 <!-- Small Box (Stat card) -->
11 <!--<h5 class="mb-2 mt-4">Small Box</h5-->
12 <div class="row">
13
14 @if(Auth::user()->admin('admin'))
15 <div class="col-lg-3 col-6">
16 <!-- small card -->
17 <div class="small-box bg-info">
18 <div class="inner">
19 <h3>{{ $usuarioA->count() }} </h3>
20
21 <p>Gestionar Usuarios</p>
22 </div>
23 <div class="icon">
24 <i class="fas fa-shopping-cart"></i>
25 </div>
26 <a href="/admin/usuario" class="small-box-footer">
27 Mas información <i class="fas fa-arrow-circle-right"></i>
28 </a>
29 </div>
30 </div>
31 @endif
32 <!-- ./col -->
33 <div class="col-lg-3 col-6">
34 <!-- small card -->
35 <div class="small-box bg-success">
36 <div class="inner">
37 <h3>{{ $apoderadoA->count() }} </h3>
38
39 <p>Gestionar Apoderados</p>
40 </div>
41 <div class="icon">
42 <i class="ion ion-stats-bars"></i>
43 </div>
44 <a href="/admin/apoderado" class="small-box-footer">
45 Mas información <i class="fas fa-arrow-circle-right"></i>
46 </a>
47 </div>
48 </div>
49 <!-- ./col -->
50 <div class="col-lg-3 col-6">
51 <!-- small card -->
52 <div class="small-box bg-warning">
53 <div class="inner">
54 <h3>{{ $alumnoA->count() }} </h3>
55
56 <p>Gestionar Alumnos</p>
57 </div>
58 <div class="icon">
59 <i class="fas fa-user-plus"></i>
60 </div>
61 <a href="/admin/alumno" class="small-box-footer">
62 Mas información <i class="fas fa-arrow-circle-right"></i>
63 </a>
64 </div>
65 </div>
66 <!-- ./col -->
67 <div class="col-lg-3 col-6">
68 <!-- small card -->
69 <div class="small-box bg-danger">
70 <div class="inner">
71 <h3>{{ $docenteA->count() }} </h3>
```

Nota. Código PHP de Vistas de Layouts (Diseños) (home.blade.php).

Figura 154: Vistas de Layouts (Diseños) (home.blade.php).



```
73 <p>Gestionar Docentes</p>
74 </div>
75 <div class="icon">
76 <i class="fas fa-chart-pie"></i>
77 </div>
78 <a href="/admin/docente" class="small-box-footer">
79 Mas Información <i class="fas fa-arrow-circle-right"></i>
80 </a>
81 </div>
82 </div>
83 </div>
84 <!-- ./col -->
85 <!-- ./row -->
86
87 <!-- Small Box (Stat card) -->
88 <div class="row">
89
90 <div class="col-lg-3 col-6">
91 <!-- small card -->
92 <div class="small-box bg-danger">
93 <!-- Loading (remove the following to stop the loading)-->
94 <!-- end loading -->
95 <div class="inner">
96 <h3>{{ $cursoA->count() }} </h3>
97
98 <p>Gestionar cursos</p>
99 </div>
100 <div class="icon">
101 <i class="fas fa-chart-pie"></i>
102 </div>
103 <a href="/admin/curso" class="small-box-footer">
104 Mas Información <i class="fas fa-arrow-circle-right"></i>
105 </a>
106 </div>
107 </div>
108 </div>
109 <!-- ./col -->
110 <div class="col-lg-3 col-6">
111 <!-- small card -->
112 <div class="small-box bg-info">
113 <!-- Loading (remove the following to stop the loading)-->
114 <div class="inner">
115 <h3>{{ $anioacademicoA->count() }} </h3>
116
117 <p>Gestionar Año Académico</p>
118 </div>
```

Nota. Código PHP de Vistas de Layouts (Diseños) (home.blade.php).

Figura 155: Vistas de Layouts (Diseños) (home.blade.php).

```

119 <div class="icon">
120 <i class="fas fa-shopping-cart"></i>
121 </div>
122 <a href="/admin/anioacademico" class="small-box-footer">
123 Mas Información <i class="fas fa-arrow-circle-right"></i>
124 </a>
125 </div>
126 </div>
127 <!-- /col -->
128 <div class="col-lg-3 col-6">
129 <!-- small card -->
130 <div class="small-box bg-info">
131 <!-- Loading (remove the following to stop the loading)-->
132 <div class="inner">
133 <h3>{{ $gradoA->count() }} </h3>
134 <p><a href="/admin/grado" class="small-box-footer">
135 Mas Información <i class="fas fa-arrow-circle-right"></i>
136 </a>
137 </div>
138 </div>
139 </div>
140 <!-- /col -->
141 <div class="col-lg-3 col-6">
142 <!-- small card -->
143 <div class="small-box bg-success">
144 <!-- Loading (remove the following to stop the loading)-->
145 <div class="inner">
146 <h3>{{ $nivelA->count() }} </h3>
147 <p><a href="/admin/nivel" class="small-box-footer">
148 Mas Información <i class="fas fa-arrow-circle-right"></i>
149 </a>
150 </div>
151 </div>
152 </div>
153 </div>
154 <!-- /col -->
155 <div class="col-lg-3 col-6">
156 <!-- small card -->
157 <div class="small-box bg-success">
158 <!-- Loading (remove the following to stop the loading)-->
159 <div class="inner">
160 <h3>{{ $horarioA->count() }} </h3>
161 <p><a href="/admin/horario" class="small-box-footer">
162 Mas Información <i class="fas fa-arrow-circle-right"></i>
163 </a>
164 </div>
165 </div>
166 </div>
167 </div>
168 <!-- /col -->
169 <div class="col-lg-3 col-6">
170 <!-- small card -->
171 <div class="small-box bg-success">
172 <!-- Loading (remove the following to stop the loading)-->
173 <div class="inner">
174 <h3>{{ $seccionA->count() }} </h3>
175 <p><a href="/admin/seccion" class="small-box-footer">
176 Mas Información <i class="fas fa-arrow-circle-right"></i>
177 </a>
178 </div>
179 </div>
180 </div>
181 </div>
182 </div>
183 </div>
184 </div>
185 </div>
186 </div>
187 </div>
188 </div>
189 </div>

```

Nota. Código PHP de Vistas de Layouts (Diseños) (home.blade.php).

Figura 156: Vistas de Layouts (Diseños) (home.blade.php).

```

191 <div class="icon">
192 <i class="ion ion-stats-bars"></i>
193 </div>
194 <a href="/admin/seccion" class="small-box-footer">
195 Mas Información <i class="fas fa-arrow-circle-right"></i>
196 </a>
197 </div>
198 </div>
199 <!-- /col -->
200 <!-- /col -->
201 <div class="col-lg-3 col-6">
202 <!-- small card -->
203 <div class="small-box bg-info">
204 <!-- Loading (remove the following to stop the loading)-->
205 <div class="inner">
206 <h3>{{ $matriculaA->count() }} </h3>
207 <p><a href="/admin/matricula" class="small-box-footer">
208 Mas Información <i class="fas fa-arrow-circle-right"></i>
209 </a>
210 </div>
211 </div>
212 </div>
213 </div>
214 </div>
215 </div>
216 </div>
217 </div>
218 </div>
219 </div>
220 <!-- /row -->
221 </div>
222 <!-- =====>
223 @stop
224 @section('css')
225 <link rel="stylesheet" href="/css/admin_custom.css">
226 @stop
227 @section('js')
228 <script> console.log('Hi!'); </script>
229 @stop
230 </div>
231 </div>
232 </div>

```

Nota. Código PHP de Vistas de Layouts (Diseños) (home.blade.php).

Figura 157: Vistas de Layouts (Diseños) (welcome.blade.php).

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <meta name="description" content="Free educational-responsive-web-template-webEdu">
7 <meta name="author" content="webthemz.com">
8 <title>Colegio</title>
9 <link rel="favicon" href="assets/images/favicon.png">
10 <link rel="stylesheet" media="screen" href="http://fonts.googleapis.com/
11 css?family=Open+Sans:300,400,700">
12 <link rel="stylesheet" href="assets/css/bootstrap.min.css">
13 <link rel="stylesheet" href="assets/css/font-awesome.min.css">
14 <link rel="stylesheet" href="assets/css/bootstrap-theme.css" media="screen">
15 <link rel="stylesheet" href="assets/css/style.css">
16 <link rel="stylesheet" id="camera-css" href="assets/css/camera.css" type="text/css"
17 media="all">
18 </head>
19 <body>
20 <!-- Fixed navbar -->
21 <div class="navbar navbar-inverse">
22 <div class="navbar-header">
23 <!-- Button for smallest screens -->
24 <button type="button" class="navbar-toggle" data-toggle="collapse"
25 data-target=".navbar-collapse"><span class="icon-bar"></span><span class="
26 icon-bar"></span><span class="icon-bar"></span></button>
27 <a class="navbar-brand" href="{ asset('/') }">
28 
29 </div>
30 <div class="navbar-collapse collapse">
31 <ul class="nav navbar-nav pull-right mainNav">
32 <li class="active"><a href="index.html">Inicio</a></li>
33 <li><a href="acerca/colegio">Acerca de </a></li>
34 <li><a href="contacto/colegio">Contacto</a></li>
35 <li><a href="{ asset('home') }">Iniciar Sesión</a></li>
36 <li><a href="/contacto/colegio"></a></li>
37 </ul>
38 </div>
39 </div>
40 <!-- /.navbar -->
41
42 <!-- Header -->
43 <header id="header">
44 <div class="container">
45 <div class="heading-text">
46 <h1 class="animated flipInY delay1">Institución Educativa N°
47 16498 Tupac Amaru II - Huarango</h1>
48 <p>>TUPACAMARINO HOY, MAÑANA Y SIEMPRE</p>
49 </div>
50 <div class="fluid_container">
51 <div class="camera_wrap camera_emboss pattern_1" id="camera_wrap_4">
52 <div data-thumb=assets/images/slides/thumbs/imagen2.jpg"
53 data-src=assets/images/slides/imagen2.jpg">
54 <h2>We develop.</h2>
55 </div>
56 <div data-thumb=assets/images/slides/thumbs/imagen1.jpg"
57 data-src=assets/images/slides/imagen1.jpg">
58 </div>
59 <div data-thumb=assets/images/slides/thumbs/imagen3.jpg"
60 data-src=assets/images/slides/imagen3.jpg">
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70
71 <!-- /Header -->
72
73 <!-- Main Content -->
74 <div class="container">
75 <div class="row">
76 <div class="col-md-4">
77 <div class="title-box clearfix">
78 <h2 class="
79 title-box-primary">Sobre nosotros</h2>
80 <div class="col-md-4">
81 <div class="list styled custom-list">
82 <li><a href="#">Matemática</a>
83 <li><a href="#">Ciencias Exactas</a>
84 <li><a href="#">Comunicación</a>
85 <li><a href="#">Educación religiosa</a>
86 <li><a href="#">Historia</a>
87 </li>
88 </div>
89 <div class="news-box top-margin">
90 <div class="row">
91 <div class="col-lg-4 col-md-4 col-sm-12">
92 <div class="thumbnail">
93 <img alt="Thumbnail 1" data-src=assets/images/slides/imagen1.jpg" />
94 <div class="caption maxheight2">
95 <div class="box">
96 <h3 class="title">Información</h3>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>

```

Nota. Código PHP de Vistas de Layouts (Diseños) (welcome.blade.php).

Figura 158: Vistas de Layouts (Diseños) (welcome.blade.php).

```

67 <section class="container">
68 <div class="row">
69 <div class="col-md-4">
70 <div class="title-box clearfix">
71 <h2 class="
72 title-box-primary">Sobre nosotros</h2>
73 <div class="col-md-4">
74 <div class="list styled custom-list">
75 <li><a href="#">Matemática</a>
76 <li><a href="#">Ciencias Exactas</a>
77 <li><a href="#">Comunicación</a>
78 <li><a href="#">Educación religiosa</a>
79 <li><a href="#">Historia</a>
80 </li>
81 </div>
82 <div class="news-box top-margin">
83 <div class="row">
84 <div class="col-lg-4 col-md-4 col-sm-12">
85 <div class="thumbnail">
86 <img alt="Thumbnail 1" data-src=assets/images/slides/imagen1.jpg" />
87 <div class="caption maxheight2">
88 <div class="box">
89 <h3 class="title">Información</h3>
90 </div>
91 </div>
92 </div>
93 </div>
94 </div>
95 </div>
96 </div>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
115 </div>
116 </div>
117 </div>
118 </div>
119 </div>

```

Nota. Código PHP de Vistas de Layouts (Diseños) (welcome.blade.php).

Figura 159: Vistas de Layouts (Diseños) (welcome.blade.php).

```

120 </div>
121 </div>
122 </div>
123 </div>
124 <div class="col-lg-4 col-md-4 col-sm-12">
125 <div class="newsBox">
126 <div class="thumbnail">
127 <figure>
128 </figure>
129 <div class="caption maxheight2">
130 <div class="box_inner">
131 <div class="box">
132 <p class="title"><h5></h5></p>
133 <p></p>
134 </div>
135 </div>
136 </div>
137 </div>
138 </div>
139 </div>
140 </div>
141 </section>
142 </div>
143 <section class="news-box top-margin">
144 <div class="container">
145 <h2><span>Contactanos</span></h2>
146 <div class="row">
147 <!-- PARA RELLENAR EL FORMULARIO DE CONTACTANOS-->
148 <div class="col-lg-7 mt-5 mt-lg-0 d-flex align-items-stretch">
149 <div class="info">
150 <div class="row">
151 <div class="col-md-6">
152 <div class="col-md-6">
153 </div>
154 </div>
155 </div>
156 </div>
157 </div>
158 </section>
159 </div>
160 <div class="container">
161 <div class="row">
162 <div class="row">
163 <div class="col-md-3 col-sm-6">
164 <div class="col-md-3 col-sm-6">
165 <div class="col-md-3 col-sm-6">
166 <div class="col-md-3 col-sm-6">
167 <div class="col-md-3 col-sm-6">
168 <div class="col-md-3 col-sm-6">
169 <div class="col-md-3 col-sm-6">
170 <div class="col-md-3 col-sm-6">
171 <div class="col-md-3 col-sm-6">
172 <div class="col-md-3 col-sm-6">
173 <div class="col-md-3 col-sm-6">
174 <div class="col-md-3 col-sm-6">
175 <div class="col-md-3 col-sm-6">
176 <div class="col-md-3 col-sm-6">
177 <div class="col-md-3 col-sm-6">
178 <div class="col-md-3 col-sm-6">
179 <div class="col-md-3 col-sm-6">
180 <div class="col-md-3 col-sm-6">
181 <div class="col-md-3 col-sm-6">
182 <div class="col-md-3 col-sm-6">
183 <div class="col-md-3 col-sm-6">
184 <div class="col-md-3 col-sm-6">

```

Nota. Código PHP de Vistas de Layouts (Diseños) (welcome.blade.php).

Figura 160: Vistas de Layouts (Diseños) (welcome.blade.php).

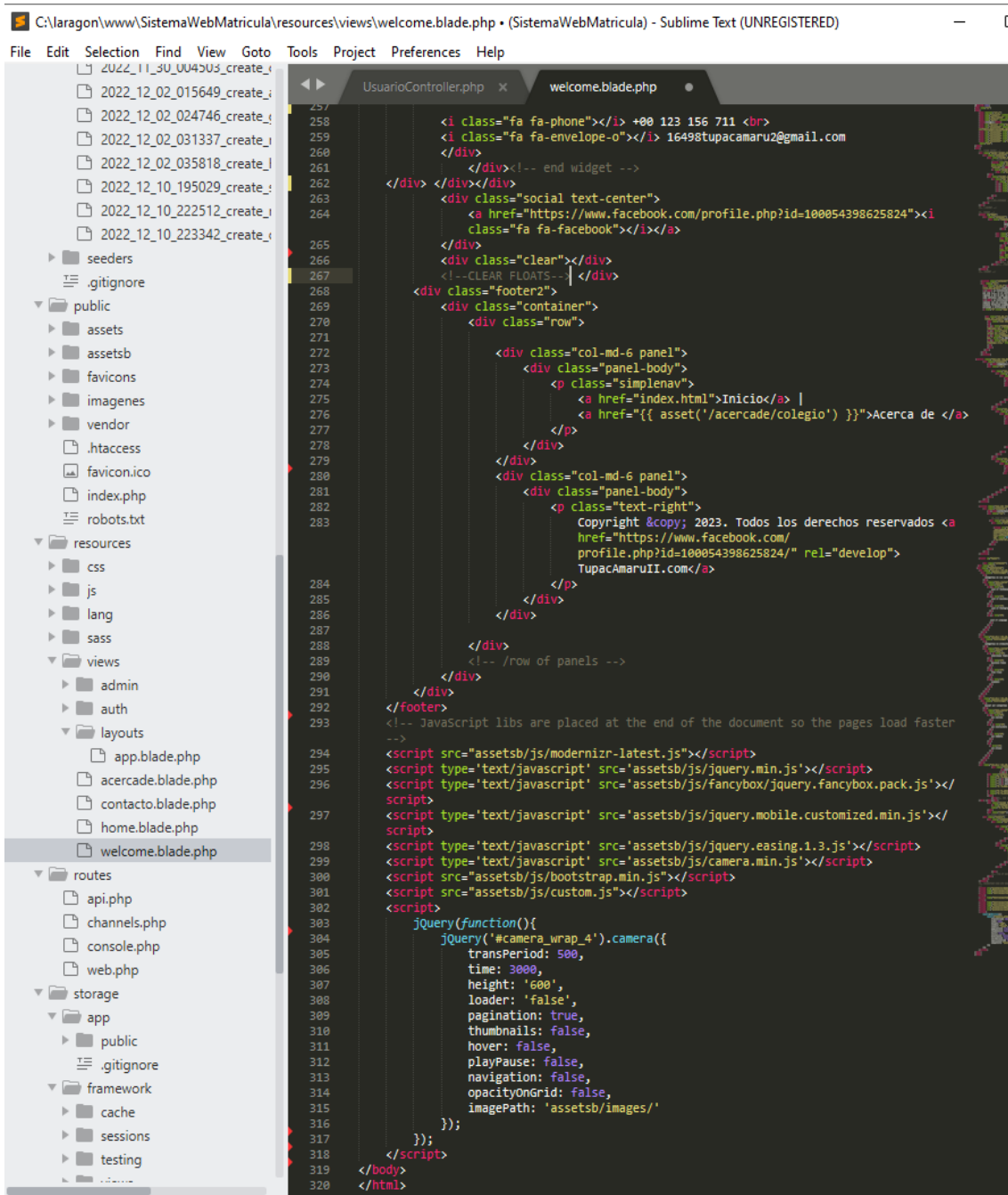
```

186 <div class="col-md-3 col-sm-6">
187 <div class="col-md-3 col-sm-6">
188 <div class="col-md-3 col-sm-6">
189 <div class="col-md-3 col-sm-6">
190 <div class="col-md-3 col-sm-6">
191 <div class="col-md-3 col-sm-6">
192 <div class="col-md-3 col-sm-6">
193 <div class="col-md-3 col-sm-6">
194 <div class="col-md-3 col-sm-6">
195 <div class="col-md-3 col-sm-6">
196 <div class="col-md-3 col-sm-6">
197 <div class="col-md-3 col-sm-6">
198 <div class="col-md-3 col-sm-6">
199 <div class="col-md-3 col-sm-6">
200 <div class="col-md-3 col-sm-6">
201 <div class="col-md-3 col-sm-6">
202 <div class="col-md-3 col-sm-6">
203 <div class="col-md-3 col-sm-6">
204 <div class="col-md-3 col-sm-6">
205 <div class="col-md-3 col-sm-6">
206 <div class="col-md-3 col-sm-6">
207 <div class="col-md-3 col-sm-6">
208 <div class="col-md-3 col-sm-6">
209 <div class="col-md-3 col-sm-6">
210 <div class="col-md-3 col-sm-6">
211 <div class="col-md-3 col-sm-6">
212 <div class="col-md-3 col-sm-6">
213 <div class="col-md-3 col-sm-6">
214 <div class="col-md-3 col-sm-6">
215 <div class="col-md-3 col-sm-6">
216 <div class="col-md-3 col-sm-6">
217 <div class="col-md-3 col-sm-6">
218 <div class="col-md-3 col-sm-6">
219 <div class="col-md-3 col-sm-6">
220 <div class="col-md-3 col-sm-6">
221 <div class="col-md-3 col-sm-6">
222 <div class="col-md-3 col-sm-6">
223 <div class="col-md-3 col-sm-6">
224 <div class="col-md-3 col-sm-6">
225 <div class="col-md-3 col-sm-6">
226 <div class="col-md-3 col-sm-6">
227 <div class="col-md-3 col-sm-6">
228 <div class="col-md-3 col-sm-6">
229 <div class="col-md-3 col-sm-6">
230 <div class="col-md-3 col-sm-6">
231 <div class="col-md-3 col-sm-6">
232 <div class="col-md-3 col-sm-6">
233 <div class="col-md-3 col-sm-6">
234 <div class="col-md-3 col-sm-6">
235 <div class="col-md-3 col-sm-6">
236 <div class="col-md-3 col-sm-6">
237 <div class="col-md-3 col-sm-6">
238 <div class="col-md-3 col-sm-6">
239 <div class="col-md-3 col-sm-6">
240 <div class="col-md-3 col-sm-6">
241 <div class="col-md-3 col-sm-6">
242 <div class="col-md-3 col-sm-6">
243 <div class="col-md-3 col-sm-6">
244 <div class="col-md-3 col-sm-6">
245 <div class="col-md-3 col-sm-6">
246 <div class="col-md-3 col-sm-6">
247 <div class="col-md-3 col-sm-6">
248 <div class="col-md-3 col-sm-6">
249 <div class="col-md-3 col-sm-6">
250 <div class="col-md-3 col-sm-6">
251 <div class="col-md-3 col-sm-6">
252 <div class="col-md-3 col-sm-6">
253 <div class="col-md-3 col-sm-6">
254 <div class="col-md-3 col-sm-6">
255 <div class="col-md-3 col-sm-6">
256 <div class="col-md-3 col-sm-6">

```

Nota. Código PHP de Vistas de Layouts (Diseños) (welcome.blade.php).

Figura 161: Vistas de Layouts (Diseños) (welcome.blade.php).



```
C:\laragon\www\SistemaWebMatricula\resources\views/welcome.blade.php • (SistemaWebMatricula) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
257 <i class="fa fa-phone"></i> +00 123 156 711 <br>
258 <i class="fa fa-envelope-o"></i> 16498tupacamaru2@gmail.com
259 </div>
260 </div><!-- end widget -->
261 </div></div></div>
262 <div class="social text-center">
263 <a href="https://www.facebook.com/profile.php?id=100054398625824"><i
264 class="fa fa-facebook"></i></a>
265 </div>
266 <div class="clear"></div>
267 <!-- CLEAR FLOATS --></div>
268 <div class="footer2">
269 <div class="container">
270 <div class="row">
271 <div class="col-md-6 panel">
272 <div class="panel-body">
273 <p class="simplenav">
274 <a href="index.html">Inicio</a> |
275 <a href="{{ asset('/acercade/colegio') }}">Acerca de </a>
276 </p>
277 </div>
278 </div>
279 <div class="col-md-6 panel">
280 <div class="panel-body">
281 <p class="text-right">
282 Copyright &copy; 2023. Todos los derechos reservados <a
283 href="https://www.facebook.com/profile.php?id=100054398625824/" rel="develop">
284 TupacAmaruII.com</a>
285 </p>
286 </div>
287 </div>
288 </div>
289 <!-- /row of panels -->
290 </div>
291 </div>
292 </footer>
293 <!-- Javascript libs are placed at the end of the document so the pages load faster -->
294 <script src="assetsb/js/modernizr-latest.js"></script>
295 <script type="text/javascript" src="assetsb/js/jquery.min.js"></script>
296 <script type="text/javascript" src="assetsb/js/fancybox/jquery.fancybox.pack.js"></
297 script>
298 <script type="text/javascript" src="assetsb/js/jquery.mobile.customized.min.js"></
299 script>
300 <script type="text/javascript" src="assetsb/js/jquery.easing.1.3.js"></script>
301 <script type="text/javascript" src="assetsb/js/camera.min.js"></script>
302 <script src="assetsb/js/custom.js"></script>
303 <script>
304 jquery(function(){
305 jquery('#camera_wrap_4').camera({
306 transPeriod: 500,
307 time: 3000,
308 height: '600',
309 loader: 'false',
310 pagination: true,
311 thumbnails: false,
312 hover: false,
313 playPause: false,
314 navigation: false,
315 opacityOnGrid: false,
316 imagePath: 'assetsb/images/'
317 });
318 </script>
319 </body>
320 </html>
```

Nota. Código PHP de Vistas de Layouts (Diseños) (home.blade.php).

Figura 162: *Routes (api.php)*.



```
1 <?php
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Route;
5
6 /*
7  |-----|
8  | API Routes
9  |-----|
10
11 | Here is where you can register API routes for your application. These
12 | routes are loaded by the RouteServiceProvider within a group which
13 | is assigned the "api" middleware group. Enjoy building your API!
14
15 */
16
17 Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
18     return $request->user();
19 });
20
```

Nota. Código PHP de *Routes (api.php)*.

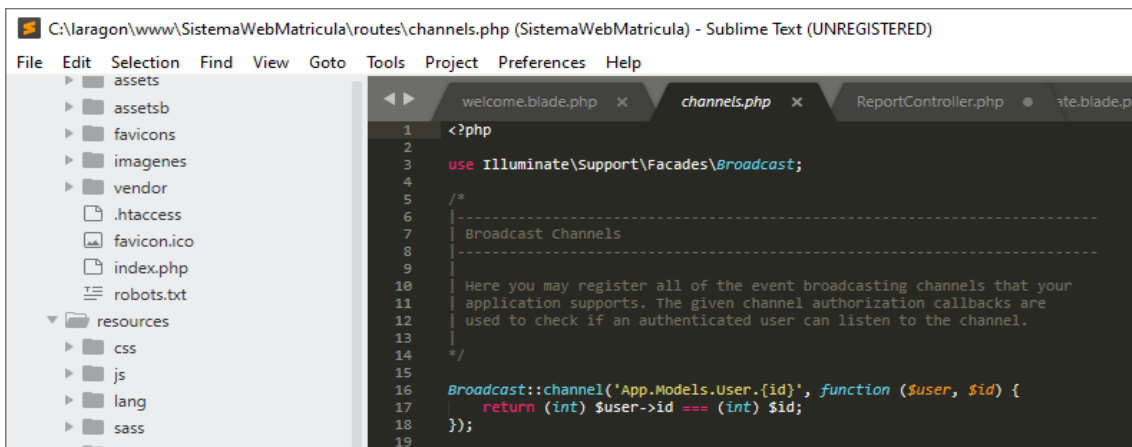
Figura 163: *Routes (api.php)*.



```
1 <?php
2
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\Route;
5
6 /*
7  |-----|
8  | API Routes
9  |-----|
10
11 | Here is where you can register API routes for your application. These
12 | routes are loaded by the RouteServiceProvider within a group which
13 | is assigned the "api" middleware group. Enjoy building your API!
14
15 */
16
17 Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
18     return $request->user();
19 });
20
```

Nota. Código PHP de *Routes (api.php)*.

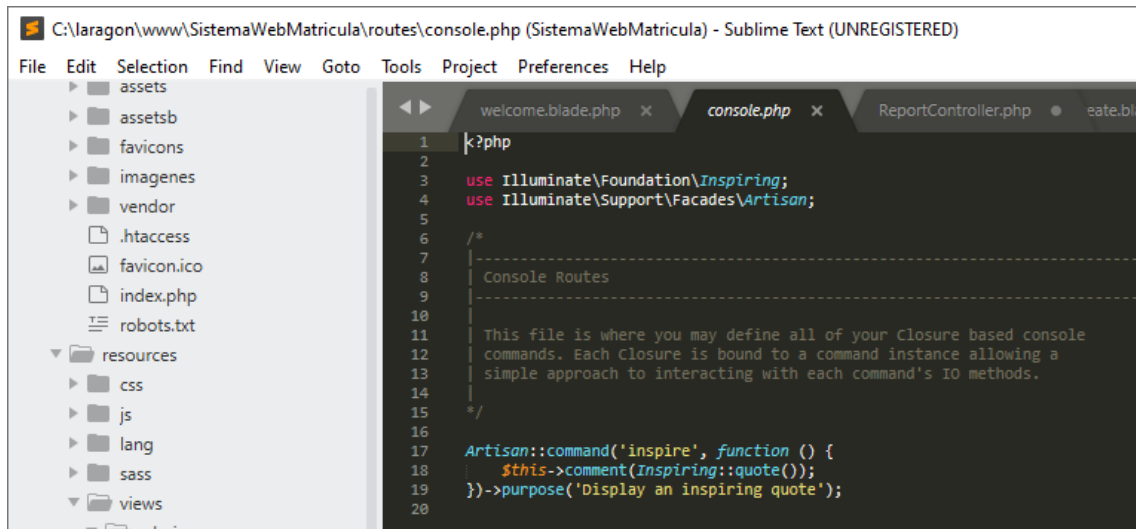
Figura 164: *Routes (channels.php)*.



```
1 <?php
2
3 use Illuminate\Support\Facades\Broadcast;
4
5 /*
6  |-----|
7  | Broadcast Channels
8  |-----|
9
10 | Here you may register all of the event broadcasting channels that your
11 | application supports. The given channel authorization callbacks are
12 | used to check if an authenticated user can listen to the channel.
13
14 */
15
16 Broadcast::channel('App.Models.User.{id}', function ($user, $id) {
17     return (int) $user->id === (int) $id;
18 });
19
```

Nota. Código PHP de *Routes (channels.php)*.

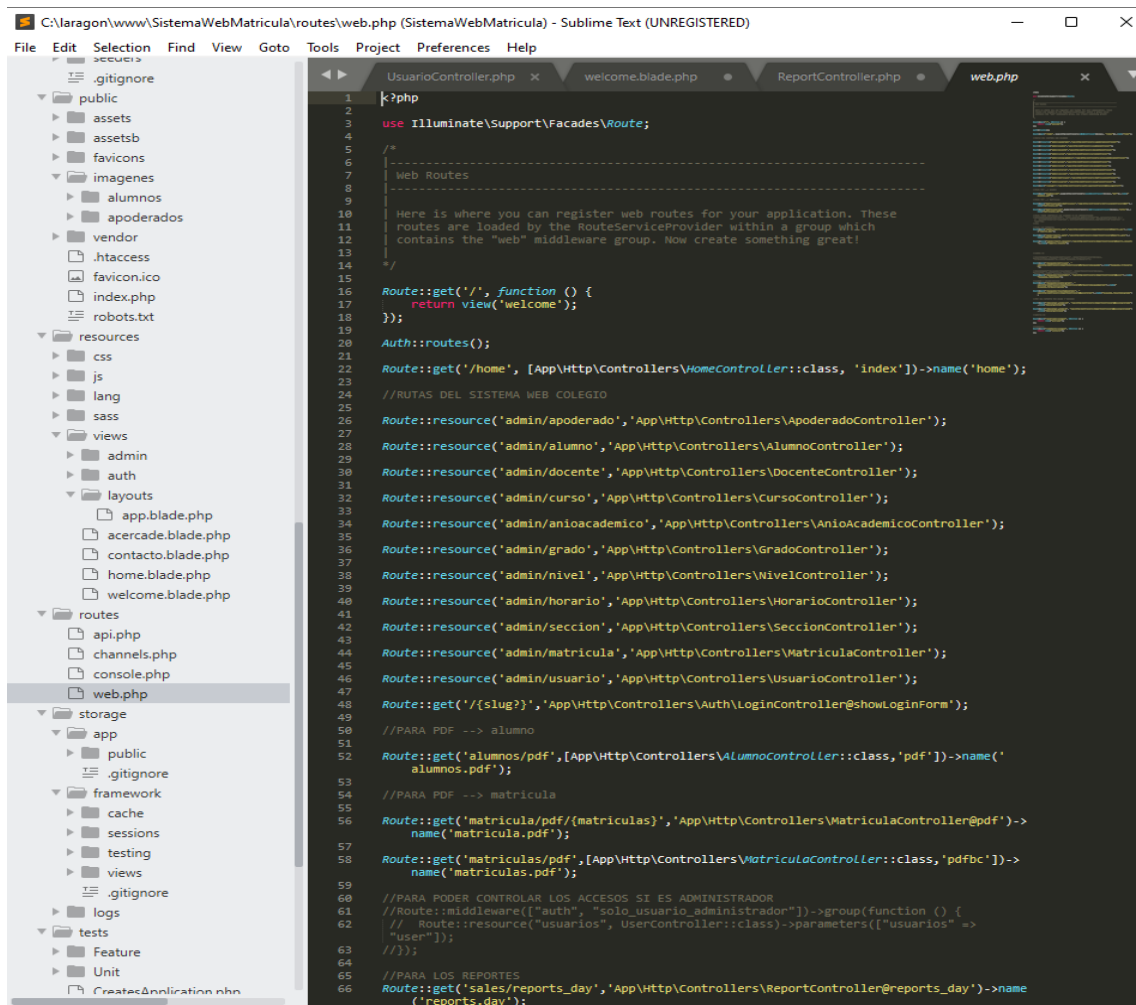
Figura 165: Routes (console.php).



```
1 |<?php
2
3 | use Illuminate\Foundation\Inspiring;
4 | use Illuminate\Support\Facades\Artisan;
5
6 | /*
7 |  |-----|
8 |  | Console Routes
9 |  |-----|
10 |
11 |  | This file is where you may define all of your Closure based console
12 |  | commands. Each Closure is bound to a command instance allowing a
13 |  | simple approach to interacting with each command's IO methods.
14 |  |
15 |  | */
16
17 | Artisan::command('inspire', function () {
18 |     $this->comment(Inspiring::quote());
19 | })->purpose('Display an inspiring quote');
```

Nota. Código PHP de Routes (console.php).

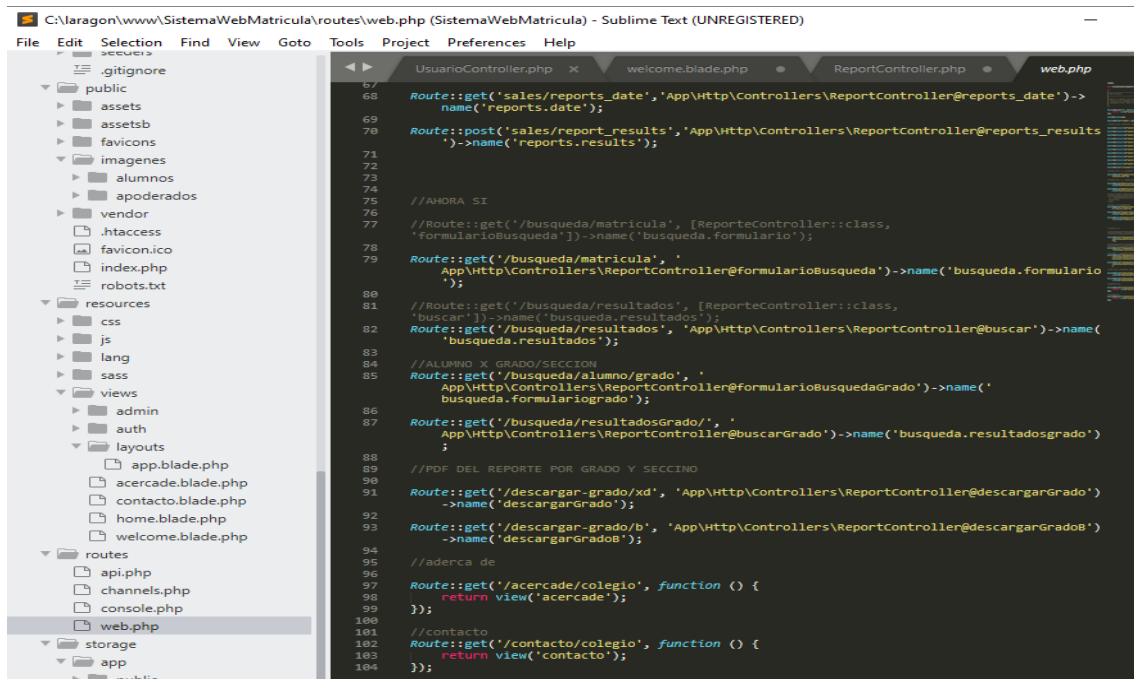
Figura 166: Routes (web.php).



```
1 |<?php
2
3 | use Illuminate\Support\Facades\Route;
4
5 | /*
6 |  |-----|
7 |  | Web Routes
8 |  |-----|
9 |
10 |  | Here is where you can register web routes for your application. These
11 |  | routes are loaded by the RouteServiceProvider within a group which
12 |  | contains the "web" middleware group. Now create something great!
13 |  |
14 |  | */
15
16 | Route::get('/', function () {
17 |     return view('welcome');
18 | });
19
20 | Auth::routes();
21
22 | Route::get('/home', [App\Http\Controllers\HomeController::class, 'index']->name('home'));
23
24 | //RUTAS DEL SISTEMA WEB COLEGIO
25
26 | Route::resource('admin/apoderado', 'App\Http\Controllers\ApoderadoController');
27
28 | Route::resource('admin/alumno', 'App\Http\Controllers\AlumnoController');
29
30 | Route::resource('admin/docente', 'App\Http\Controllers\DocenteController');
31
32 | Route::resource('admin/curso', 'App\Http\Controllers\CursoController');
33
34 | Route::resource('admin/anioacademico', 'App\Http\Controllers\AnioAcademicoController');
35
36 | Route::resource('admin/grado', 'App\Http\Controllers\GradoController');
37
38 | Route::resource('admin/nivel', 'App\Http\Controllers\NivelController');
39
40 | Route::resource('admin/horario', 'App\Http\Controllers\HorarioController');
41
42 | Route::resource('admin/seccion', 'App\Http\Controllers\SeccionController');
43
44 | Route::resource('admin/matricula', 'App\Http\Controllers\MatriculaController');
45
46 | Route::resource('admin/usuario', 'App\Http\Controllers\UsuarioController');
47
48 | Route::get('/{slug?}', 'App\Http\Controllers\Auth\LoginController@showLoginForm');
49
50 | //PARA PDF --> alumno
51
52 | Route::get('alumnos/pdf', [App\Http\Controllers\AlumnoController::class, 'pdf']->name('alumnos.pdf'));
53
54 | //PARA PDF --> matricula
55
56 | Route::get('matricula/pdf/{matriculas}', 'App\Http\Controllers\MatriculaController@pdf')->name('matricula.pdf');
57
58 | Route::get('matriculas/pdf', [App\Http\Controllers\MatriculaController::class, 'pdfbc']->name('matriculas.pdf'));
59
60 | //PARA PODER CONTROLAR LOS ACCESOS SI ES ADMINISTRADOR
61 | //Route::middleware(['auth', 'solo_usuario_administrador']->group(function () {
62 | //    Route::resource('usuarios', UserController::class)->parameters(['usuarios' => 'user']);
63 | //});
64
65 | //PARA LOS REPORTES
66 | Route::get('sales/reports_day', 'App\Http\Controllers\ReportController@reports_day')->name('reports.day');
```

Nota. Código PHP de Routes (web.php).

Figura 167: Routes (web.php).



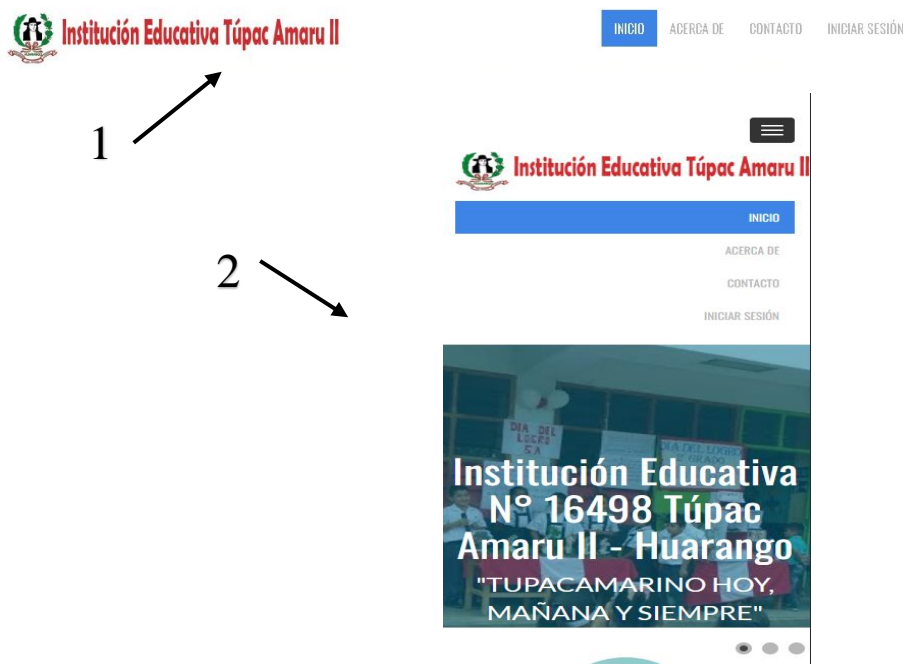
```
67 Route::get('sales/reports_date', 'App\Http\Controllers\ReportController@reports_date')->
68 name('reports.date');
69
70 Route::post('sales/report_results', 'App\Http\Controllers\ReportController@reports_results')->
71 name('reports.results');
72
73 //AHORA SI
74
75 //Route::get('/busqueda/matricula', [ReporteController::class,
76 'formularioBusqueda']->name('busqueda.formulario'));
77
78 Route::get('/busqueda/matricula', '
79 App\Http\Controllers\ReportController@formularioBusqueda')->name('busqueda.formulario
80 ');
81
82 //Route::get('/busqueda/resultados', [ReporteController::class,
83 'buscar']->name('busqueda.resultados'));
84
85 Route::get('/busqueda/resultados', 'App\Http\Controllers\ReportController@buscar')->name(
86 'busqueda.resultados');
87
88 //ALUMNO X GRADO/SECCION
89 Route::get('/busqueda/alumno/grado', '
90 App\Http\Controllers\ReportController@formularioBusquedaGrado')->name('
91 busqueda.formulariogrado');
92
93 Route::get('/busqueda/resultadosGrado/', '
94 App\Http\Controllers\ReportController@buscarGrado')->name('busqueda.resultadosgrado')
95 );
96
97 //PDF DEL REPORTE POR GRADO Y SECCION
98 Route::get('/descargar-grado/xd', 'App\Http\Controllers\ReportController@descargarGrado')
99 ->name('descargargrado');
100
101 Route::get('/descargar-grado/b', 'App\Http\Controllers\ReportController@descargarGradoB')
102 ->name('descargargradoB');
103
104 //acerca de
105 Route::get('/acercade/colegio', function () {
106     return view('acercade');
107 });
108
109 //contacto
110 Route::get('/contacto/colegio', function () {
111     return view('contacto');
112 });
```

Nota. Código PHP de Routes (web.php).

Interfaz gráfica: Usuario

- ✓ **Barra de navegación:** Son los diferentes botones que permiten acceder a información vinculada con el sitio web.

Figura 168: Barra de navegación para (1) Pc y (2) Teléfono.



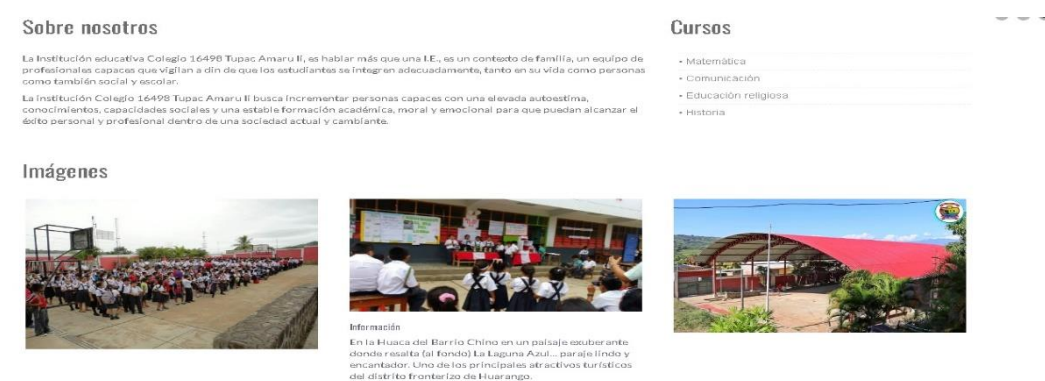
- ✓ **Inicio:** en el inicio del sistema académico web, aparecerán distintos datos de la Institución correspondiente, como el nombre de la institución, su insignia, frase propia, imagen propia, sobre nosotros, cursos, imágenes, contáctanos.

Figura 169: Inicio.



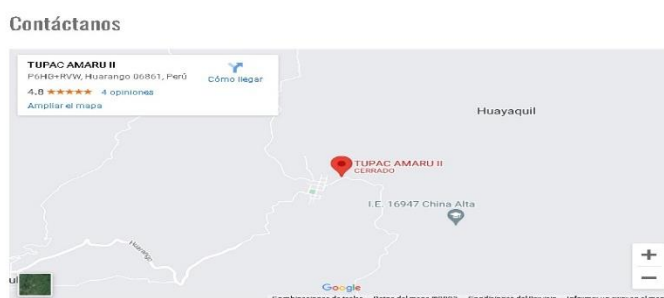
Nota. Página de inicio.

Figura 170: Inicio.



Nota. Página de inicio.

Figura 171: Inicio.



Nota. Página de inicio.

- ✓ **Acerca de:** al hacer clic en botón acerca de, nos mostrará una ventana donde aparecerá una descripción de la propia Institución donde detallará sus datos.

Figura 172: *Acerca de*

Institución Educativa Túpac Amaru II

INICIO **ACERCA DE** CONTACTO INICIAR SESIÓN

COLEGIO 16498 TUPAC AMARU II HUARANGO – HUARANGO

Es esta sección podrás encontrar información del colegio!

La Institución educativa Colegio 16498 Tupac Amaru II se localiza en el poblado de Huarango, provincia de Huarango, tal IE es supedita por la UGEL SAN IGNACIO quien comprueba el servicio educativo, y esta última corresponde a la Dirección regional de educación DRE CAJAMARCA.

La Institución educativa Colegio 16498 Tupac Amaru II, es hablar más que una I.E., es un contexto de familia, un equipo de profesionales capaces que vigilan a día de que los estudiantes se integren adecuadamente, tanto en su vida como personas como también social y escolar.

La Institución Colegio 16498 Tupac Amaru II busca incrementar personas capaces con una elevada autoestima, conocimientos, capacidades sociales y una estable formación académica, moral y emocional para que puedan alcanzar el éxito personal y profesional dentro de una sociedad actual y cambiante.

En institución educativa.info discernimos que en Huarango esta IE busca una educación de la más alta calidad en un ambiente de estudio seguro, en donde nuestros alumnos desarrollen:

Datos sobre de la I.E. Colegio 16498 Tupac Amaru II

- DRE CAJAMARCA.
- UGEL SAN IGNACIO
- Nombre/ Nº de la I. E.: 16498 Tupac Amaru II
- Nivel / Modalidad: Secundaria
- Distrito: Huarango
- Departamento: Cajamarca
- Dirección: CALLE TUPACAMARU CS/N
- Email: 16498tupacamaru2@gmail.com
- Tipo de gestión: Pública de gestión directa
- Género de los alumnos: Mixto
- Forma de atención: Escolarizada

Nota. Botón acerca de.

Figura 173: *Acerca de*

total crecimiento espiritual, intelectual, moral, físico, social y emocional.

En conjunto estos elementos forman parte el Sistema de Colegio 16498 Tupac Amaru II, que es una propuesta educativa y metodológica guiada a conquistar lo mejor de cada alumnos.

Los objetivos en Escuela 16498 Tupac Amaru II mediante el aprendizaje:

- Obtener una aprendizaje de los conocimientos en consonancia con la armonía de aprendizaje.
- Desarrollar saber de análisis y raciocinio crítico.
- Invitar la curiosidad en práctica.
- Amplificar las soltura de comunicación.

FOTOS

Imágenes de nuestro colegio.

Nota. Botón acerca de.

- ✓ **Contacto:** Al hacer clic en el botón Contacto nos mostrará datos del personal en general de la Institución quienes laboran en ello.

Figura 174: Contacto

Personal directivo y administrativo :

1. Wilmer Huanca Rivera - Director
2. Angel Leodan Peña León - Sub Director
3. Rogelio Felix Tenorio Mera - Secretario
4. Amancio Nilo Cera Rodríguez - Auxiliar de Educación
5. Rogger Miky Ortega Brito - Bibliotecario
6. Ana Leydi Calle Flores - Personal de Servicio
7. Manuel Bacerra Llamo - Personal Administrativo

Personal docente nivel primaria:

Nota. Botón de contacto.

Figura 175: Contacto

Personal docente nivel primaria:

1. Daisy Cabrera Larraldeagui
3. Erica Abad Ascurra
4. Nery Huachas Condoy
5. Elvia Izquierdo Cisneros
6. Ceci Edith Tocto Concha
7. Edilys Mena Rubio
8. Ronald Vianey Flores Solano
9. Melly Tocto Núñez
10. Maura Rivera Ramos

PERSONAL DOCENTE NIVEL SECUNDARIO

1. David Llanos Dávila
2. Fortunato Coronel Cabrera
3. Carlos Heredia Huamán
4. Yaque Marley Mundaca Ramos
5. Janet Vallajos Silva
6. Félix Andrés López Flores
7. Arcely Medalla Sánchez Sánchez
8. Luis Alfonso Barrocal Cotrina
9. Carlos David Vilcheraz Cerna

Nota. Botón de contacto.

Figura 176: Contacto

PERSONAL DOCENTE NIVEL SECUNDARIO

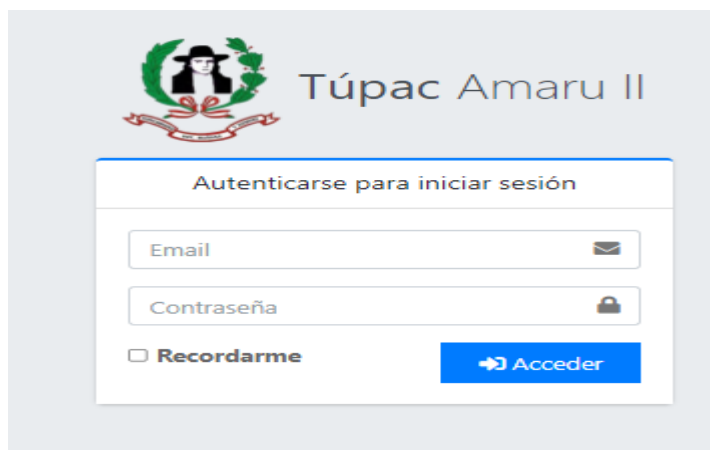
1. David Llanos Dávila
2. Fortunato Coronel Cabrera
3. Carlos Heredia Huamán
4. Yaque Marley Mundaca Ramos
5. Janet Vallajos Silva
6. Félix Andrés López Flores
7. Arcely Medalla Sánchez Sánchez
8. Luis Alfonso Barrocal Cotrina
9. Carlos David Vilcheraz Cerna
10. Aldemar Martínez Paz
11. Carmen Soña Meniz Guevara
12. Francisco López Sobarón
13. María Luzolina Vázquez Pérez
14. Maritza Rodríguez Mendoza
15. Nelson Roger Limay Carrasco
16. Tony Norman Santos Gallardo
17. Ramiro Silva Vicente

FOTOS

Nota. Botón de contacto

- ✓ **Iniciar sesión:** al hacer clic en el botón de inicio de sesión nos llevará a una ventana donde se podrá visualizar un eslogan con el nombre la institución, su insignia, un email válido, una contraseña y acceder.

Figura 177: Iniciar sesión



Nota. Botón de iniciar sesión.

- ✓ **Panel principal:** una vez ingresado un correo y una contraseña válida, nos mostrará la siguiente ventana de panel principal, donde estarán todos los botones de gestionar usuarios, apoderados, alumnos, docentes, cursos, año académico, grados, niveles, horarios, secciones y matricula que se podrán editar, eliminar, agregar.

Figura 178: Panel Principal



Nota. Botón de panel principal.

- ✓ **Registrar Matrícula:** al hacer clic en el botón de registrar matrícula se podrá registrar la matrícula del alumno con todos sus datos respectivos y necesarios.

Figura 179: *Formulario de Registrar Matrícula*

Gestionar Nueva Matrícula

Agregar Estudiante

Alumno: Miler Alvarez Sarmiento

Tipo Comprobante: Boleta Número de Comprobante: []

Serie Comprobante: [] Fecha Actual: 26/02/2023

AGREGAR CURSO

Curso: [Seleccione una opción]

Año académico: 2023 Horario: Lunes 10:45:00 - 12:00:00 Estado: Nuevo

Sección: [Seleccione una opción] Grado: [] Nivel: [Nivel]

Costo: [Ingrese el costo a pagar S/.] Cantidad: [Cantidad]

Agregar

Opciones	Curso	Año Académico	Seccion	Horario	Estado	Costo	SubTotal
Total							S/. 0.00

Nota. Formulario de registrar matrícula.

Figura 180: *Registrar Matrícula*

Listado de Matriculas

Nueva Matricula **Generar PDF**

Mostrar 10 registros por página Buscar []

ID	Fecha - Hora	Alumno Matriculado	Tipo de Comprobante	Total de Matricula	Opciones
1	2023-02-26 03:46:39	Miler Alvarez Sarmiento	Boleta: 1-1	S/. 50.00	Imprimir Detalles Anular

Mostrando página 1 de 1 Anterior 1 siguiente

Nota. Listado de matrículas registradas.

- ✓ **Perfil Usuario:** al hacer clic en el botón de perfil usuario aparecerán los usuarios disponibles y por agregar un nuevo usuario con los respectivos datos a llenar en los formularios.

Figura 181: *Formulario Perfil Usuario*

The image shows a registration form titled 'Crear una nueva cuenta' for the 'Túpac Amaru II' system. The form contains the following elements:

- Nombre completo:** A text input field with a person icon on the right.
- Email:** A text input field with an envelope icon on the right.
- Contraseña:** A text input field with a lock icon on the right.
- Confirmar la contraseña:** A text input field with a lock icon on the right.
- Tipo de Usuario:** A dropdown menu with a person icon on the right.
- Registrarse:** A prominent blue button with a person icon and a plus sign.
- Ya tengo una cuenta:** A blue link at the bottom of the form.

Nota. Formulario de perfil Usuario.

Figura 182: *Perfil Usuario*

The image shows the 'Listado de Usuarios' (User List) page in the 'Túpac Amaru II' system. The page features a sidebar on the left with navigation options, a search bar, and a main content area with the following elements:

- Header:** 'Túpac Amaru II' logo and user profile 'Miler Miler'.
- Section:** 'Listado de Usuarios' with a 'Nuevo Usuario' button.
- Status:** 'Hay 2 usuarios registrados.'
- Filters:** 'Mostrar 10 registros por página' and a search bar.
- Table:** A table with the following data:

Id	Nombre	Email	Tipo de empleado	Opciones
1	Miler	Mileradmin@gmail.com	admin	[Editar] [Eliminar]
2	Israel	israel123@gmail.com	miembro	[Editar] [Eliminar]
- Pagination:** 'Mostrando página 1 de 1' and 'Anterior 1 siguiente'.

Nota. Listado de perfil usuarios registrados.

- ✓ **Registrar Apoderado:** al hacer clic en el botón de registrar apoderado se podrá registrar un apoderado (padre de familia de estudiante) con los respectivos datos.

Figura 183: Formulario de Registrar Apoderado

Nuevo Apoderado

En este formulario se registrarán los padres de los distintos alumnos

Nombre:
Ingrese nombre

Apellido:
Ingrese apellido

Sexo:
Masculino

DNI:
Ingrese DNI

Dirección:
Ingrese direccion

Email address:
Enter email

Celular:
Ingrese celular

Imagen Ninguno archivo selec.

Nota. Formulario de apoderado.

Figura 184: Listado de Apoderados

Listado de apoderados

Mostrar registros por página

Nombre	Sexo	DNI	Dirección	Email	Celular	Imagen	Opciones
Elizandro Alvarez	Masculino	87654321	San Antonio 123	Elizandro123@gmail.com	987654321		<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>

Mostrando página 1 de 1

Nota. Listado de apoderados registrados.

- ✓ **Registrar Alumnos:** al hacer clic en el botón de registrar alumnos se podrá registrar un nuevo alumno con los respectivos datos en el cual se podrá editar y eliminar.

Figura 185: Formulario de Registrar Alumno

En este formulario se registrará a los alumnos que pertenecen a dicho colegio

Nombre
Ingrese nombre

Apellido
Ingrese apellido

Apoderado
Elizandro Alvarez

Email address
Enter email

DNI
Ingrese DNI

Fecha:
26/02/2023

Sexo
Masculino

Dirección
Ingrese direccion

Celular
Ingrese celular

Imagen
Seleccionar archivo Ninguno archivo selec.

Guardar Cancelar

Nota. Formulario de registrar alumno.

Figura 186: Listado de Alumnos registrados

Listado de alumnos

Nuevo Alumno Generar PDF

Mostrar 10 registros por página Buscar

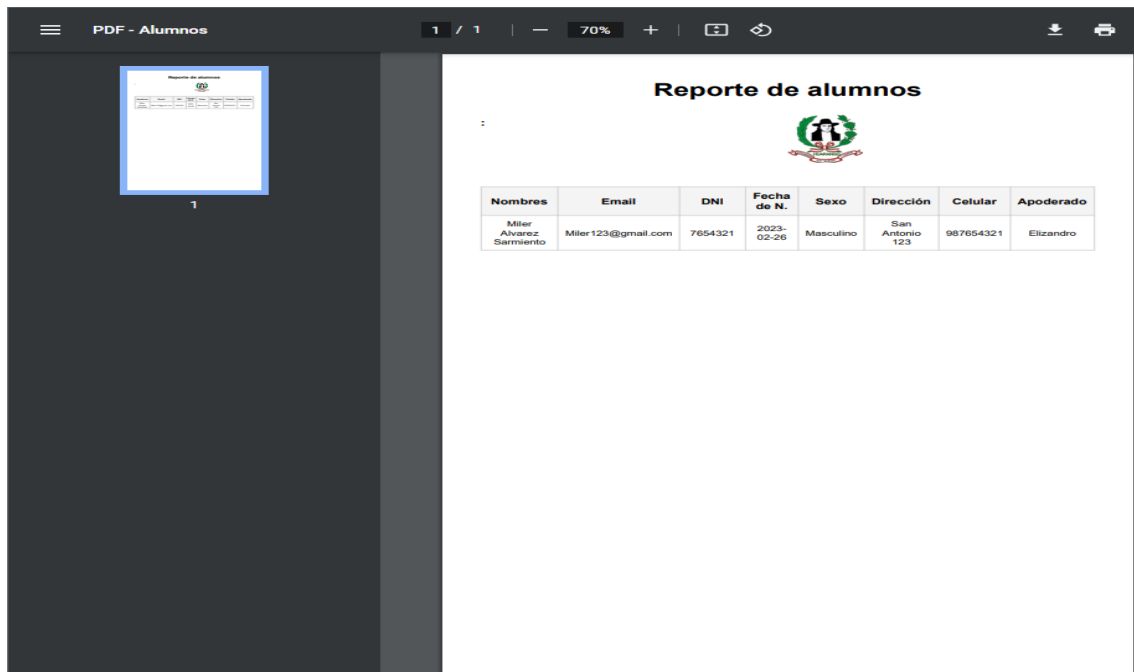
Nombres	Email	DNI	Fecha Nacimiento	Sexo	Dirección	Celular	Imagen	Apoderado
Miler Alvarez Sarmiento	Miler123@gmail.com	7654321	2023-02-26	Masculino	San Antonio 123	987654321		Elizandro

Mostrando página 1 de 1 Anterior 1 siguiente

Nota. Listado de alumnos registrados.

- ✓ **Reporte de listado de Alumnos:** al hacer clic en el botón de generar PDF, se obtendrá un resultado de un listado de alumnos registrados en PDF.

Figura 187: Reporte de listado de alumnos PDF.



Nota. Listado de alumnos registrados en PDF.

- ✓ **Registrar Docentes:** al hacer clic en el botón de registrar docente, aparecerá un formulario para llenar datos propios del docente.

Figura 188: Formulario de Registrar Docente

The screenshot shows a web application interface. On the left is a dark sidebar with a search bar and a menu. The main content area is titled 'Nuevo Docente' and contains a form for registering a teacher. The form fields are:

- Nombre:** Ingrese nombre
- Apellido:** Ingrese apellido
- Sexo:** Masculino (dropdown menu)
- Email address:** Enter email
- Descripción:** Ingrese direccion
- Celular:** Ingrese celular

At the bottom of the form are two buttons: 'Guardar' (blue) and 'Cancelar' (red).

Nota. Formulario de registrar docentes.

Figura 189: Listado de Docentes

The screenshot displays the 'Listado de docentes' (Teachers List) interface. On the left is a dark sidebar with navigation options: Inicio (Nuevo), Registrar Matricula, Panel del Estudiante (Perfil de Usuario, Registrar Apoderado, Registrar Alumnos), Registrar Docentes (highlighted), Registrar Cursos, Gestionar Horario, Gestionar Año Academico, Gestionar Secciones, and Reportes (Reporte por dia, Reporte por fecha, Reporte por grado,seccion,nivel, Reporte por seccion). The main content area has a header 'Listado de docentes' and a 'Nuevo Docente' button. Below is a search bar and a table with columns: Id, Nombres, Apellidos, Sexo, Email, Descripción, Telefono, and Opciones. The table contains one row for David Llanos Olivera. At the bottom, it shows 'Mostrando página 1 de 1' and navigation buttons for 'Anterior', '1', and 'siguiente'.

Nota. Listado de docentes registrados.

- ✓ **Registrar Cursos:** al hacer clic en el botón cursos, nos permitirá agregar respectivos cursos dependiendo a los que se enseñen en dicha institución.

Figura 190: Formulario de Registrar Cursos

The screenshot shows the 'Nuevo Curso' (New Course) form. The sidebar is identical to the previous figure. The main content area has a header 'Nuevo Curso' and a blue banner stating 'En este formulario se registrará todos los cursos que pertenecen a una determinada matricula'. The form contains four input fields: 'Nombre del curso' (with placeholder 'Ingrese el nombre del curso'), 'Cantidad de alumnos de Curso' (with placeholder 'Ingrese cantidad'), 'Docente de Curso' (a dropdown menu with 'David Llanos Olivera' selected), and 'Descripción del curso' (with placeholder 'Ingrese descripción'). At the bottom are 'Guardar' and 'Cancelar' buttons.

Nota. Formulario de registrar cursos.

Figura 191: Listado de Cursos

Listado de cursos

Nuevo Curso

Mostrar 10 registros por página

Id	Nombre Curso	Cantidad	Descripción	Docente	Opciones
1	Educación Religiosa	50	Educación Religiosa	David	Editar Eliminar

Mostrando página 1 de 1

Anterior 1 siguiente

Nota. Listado de cursos registrados.

- ✓ **Gestionar Horarios:** al hacer clic en el botón gestionar horario, el administrador podrá ingresar un nuevo horario de acuerdo a los cursos que se creen.

Figura 192: Formulario de Gestionar Horario

Nuevo Horario

En este formulario se registrará todos los horarios que pertenecen a un determinado curso

Día de la semana:
Ingrese el dia

Hora de Inicio: --:--

Hora de Fin: --:--

Guardar Cancelar

Nota. Formulario de gestionar horario.

Figura 193: Listado de Horarios

The screenshot displays the 'Listado de horarios' interface. On the left is a dark sidebar with a search bar and a list of menu items including 'Inicio', 'Registrar Matricula', 'Panel del Estudiante', 'Gestionar Horario' (highlighted), and 'Reportes'. The main content area has a header 'Listado de horarios' and a 'Nuevo Horario' button. Below this is a 'Mostrar' dropdown set to '10' and a 'Buscar' input field. A table with a blue header contains the following data:

ID	Dia	Hora de inicio	Hora de fin	Opciones
1	Lunes	10:45:00	12:00:00	Editar Eliminar

At the bottom of the table area, it says 'Mostrando página 1 de 1' and has navigation buttons for 'Anterior', '1', and 'siguiente'.

Nota. Listado de horarios registrados.

- ✓ **Gestionar Año Académico:** al hacer clic en el botón gestionar año académico, el administrador podrá ingresar el año presente para llevar a cabo las matrículas.

Figura 194: Formulario de Gestionar Año Académico

The screenshot shows the 'Nuevo Año Académico' form. The sidebar on the left is the same as in Figure 193, with 'Gestionar Año Académico' highlighted. The main content area has a header 'Nuevo Año Académico' and a blue banner that says 'En este formulario se registrará los distintos años académicos'. Below the banner are three input fields:

- Año:** Ingrese año académico
- Fecha de Inicio:** 26/02/2023
- Fecha de Fin:** 26/02/2023

At the bottom of the form are two buttons: 'Guardar' (blue) and 'Cancelar' (red).

Nota. Formulario de gestionar Año Académico.

Figura 195: Listado de Gestionar Año Académico

The screenshot displays the 'Listado de años académicos' interface. At the top, there is a 'Nuevo Año Académico' button. Below it, a control for 'Mostrar 10 registros por página' and a search box are visible. The main content is a table with the following data:

Id	Descripción	Fecha Inicio	Fecha Fin	Opciones
1	2023	2023-03-06	2023-12-19	Editar Eliminar

At the bottom of the table, it indicates 'Mostrando página 1 de 1' and navigation buttons for 'Anterior', '1', and 'siguiente'.

Nota. Listado de gestionar año académico registrados.

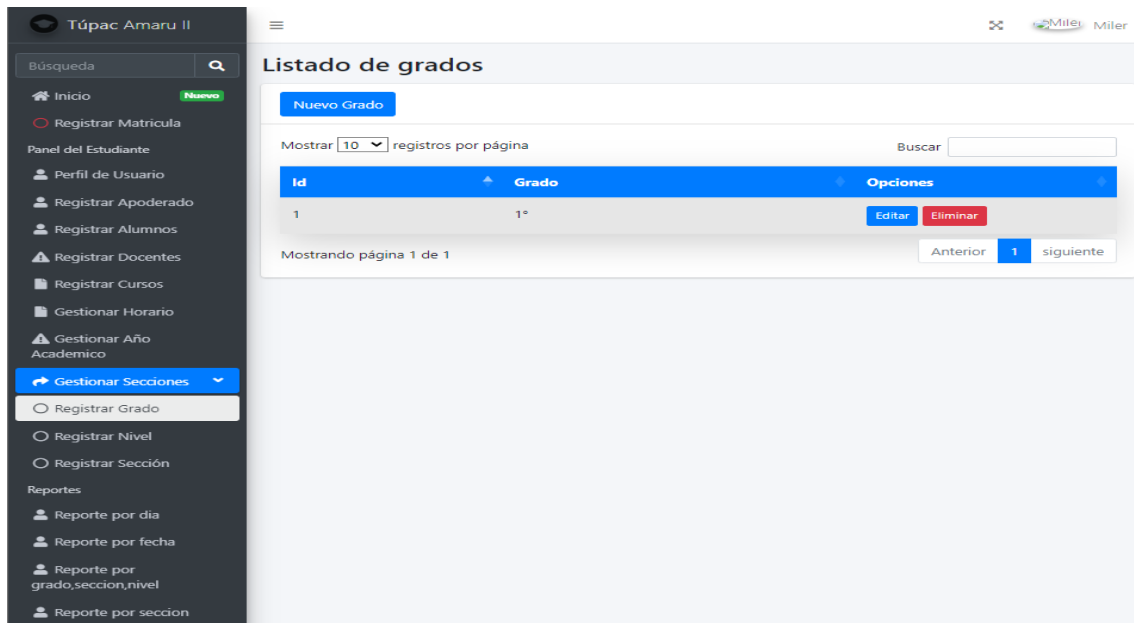
- ✓ **Gestionar Secciones (Registrar Grado):** al hacer clic en el botón gestionar secciones (Registrar Grado), el administrador podrá ingresar los respectivos grados de la institución.

Figura 196: Formulario de Gestionar Secciones (Registrar Grado).

The screenshot shows the 'Nuevo Grado' form. It features a blue header with the title 'Nuevo Grado' and a subtitle: 'En este formulario se registrará los distintos grados que tiene el colegio de acuerdo a un nivel académico'. Below this, there is a section labeled 'Grado:' with a text input field containing the placeholder 'Ingrese grado'. At the bottom of the form, there are two buttons: 'Guardar' (blue) and 'Cancelar' (red).

Nota. Formulario de grados.

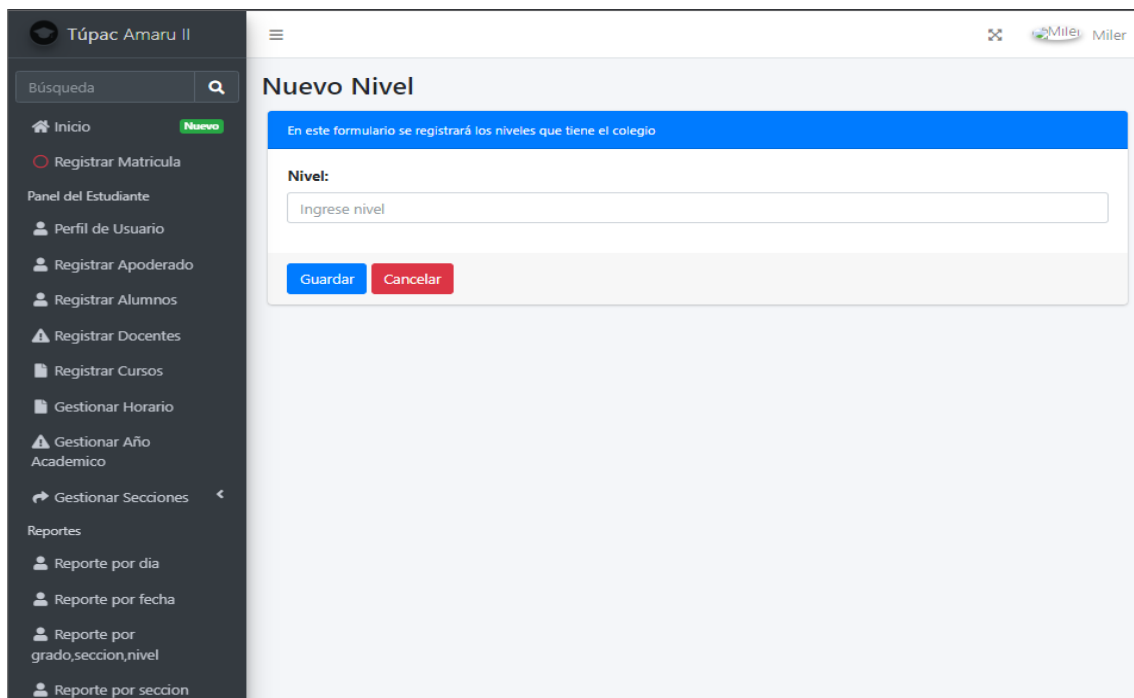
Figura 197: Listado de Gestionar Secciones (Registrar Grado).



Nota. Listado de grados registrados.

- ✓ **Gestionar Secciones (Registrar Nivel):** al hacer clic en el botón gestionar secciones (Registrar Nivel), el administrador podrá ingresar los respectivos niveles (primaria y secundaria) de la institución.

Figura 198: Formulario de Gestionar Secciones (Registrar Nivel).



Nota. Formulario de niveles.

Figura 199: Listado de Gestionar Secciones (Registrar Nivel).

The screenshot displays the 'Listado de Niveles' interface. On the left, a dark sidebar contains a search bar and a menu with 'Gestionar Secciones' expanded to show 'Registrar Nivel'. The main content area has a 'Nuevo Nivel' button, a 'Mostrar 10 registros por página' dropdown, and a search box. Below is a table with columns 'Id', 'Nivel', and 'Opciones'. The table lists two levels: '1 PRIMARIO' and '2 SECUNDARIA', each with 'Editar' and 'Eliminar' buttons. At the bottom, it shows 'Mostrando página 1 de 1' and navigation buttons 'Anterior', '1', and 'siguiente'.

Nota. Listado de niveles registrados.

- ✓ **Gestionar Secciones (Registrar Sección):** al hacer clic en el botón gestionar secciones (Registrar Sección), el administrador podrá ingresar las respectivas secciones que cuente la institución.

Figura 200: Formulario de Gestionar Secciones (Registrar Sección).

The screenshot shows the 'Nueva Sección' form. The sidebar is identical to Figure 199. The main content area has a blue header with the text 'En este formulario se registrará las secciones de acuerdo a su nivel y grado académico'. Below are four form fields: 'Grado' (dropdown with '1°'), 'Nombre de la Sección' (text input with placeholder 'Ingrese el nombre de la seccion'), 'Nivel' (dropdown with 'PRIMARIO'), and 'Cantidad de alumnos (esto ya no va):' (text input with placeholder 'Ingrese la cantidad'). At the bottom are 'Guardar' and 'Cancelar' buttons.

Nota. Formulario de sección.

Figura 201: Listado de Gestionar Secciones (Registrar Sección).

Listado de secciones

Nueva Sección

Mostrar 10 registros por página

ID Sección	Grado	Nombre de la Sección	Nivel	Cantidad	Opciones
1	1°	A	SECUNDARIA	30	Editar Eliminar

Mostrando página 1 de 1 Anterior 1 siguiente

Nota. Listado de secciones registradas.

- ✓ **Reporte de listado de Alumnos por día:** al hacer clic en el botón de generar PDF, se obtendrá un resultado de un listado de alumnos registrados en PDF por día.

Figura 202: Reportes por día de alumnos matriculados por día.

Reporte de Matriculas por dia actual

Generar PDF

Fecha de consulta: 26/02/23 Cantidad de registros: 1 Total de ingresos: S/. 50

Mostrar 10 registros por página

ID	Fecha - Hora	Total	Acciones
1	2023-02-26 03:46:39	S/. 50.00	Imprimir Detalles Anular

Mostrando página 1 de 1 Anterior 1 siguiente

Nota. Listado de alumnos matriculados por día.

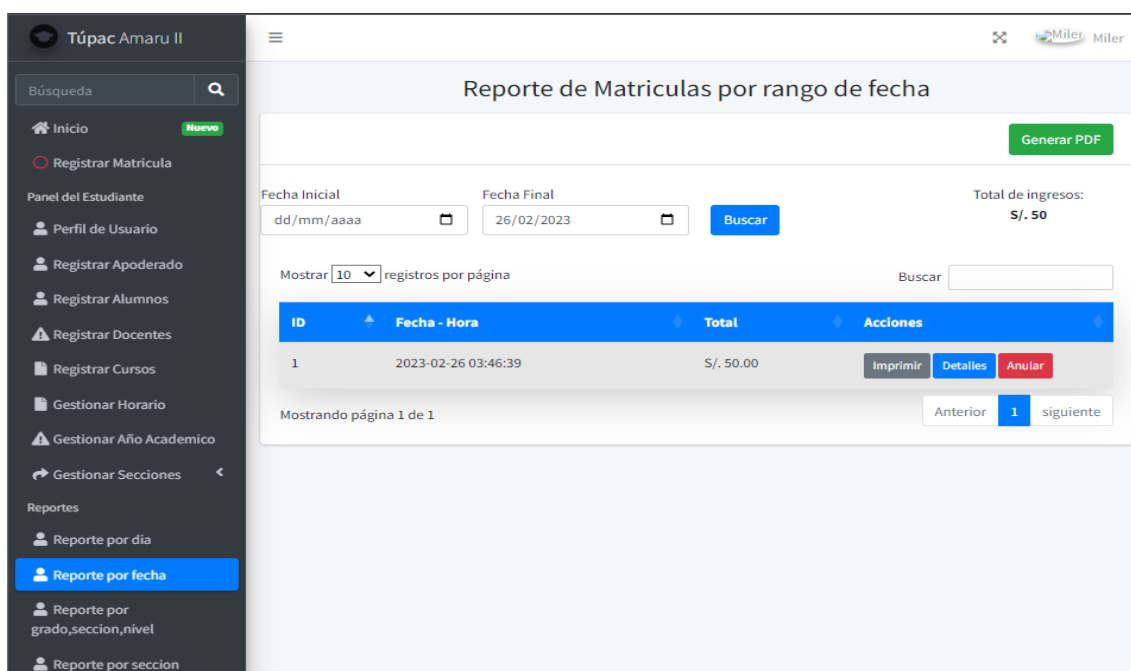
Figura 203: PDF de reportes por día de alumnos matriculados.



Nota. Reporte diario de alumnos matriculados.

- ✓ **Reporte de listado de Alumnos por rango de fecha:** al hacer clic en el botón de generar PDF, se obtendrá un resultado de un listado de alumnos registrados en PDF. Por rango de fecha.

Figura 204: Reportes por rango de fecha de alumnos matriculados.



Nota. Listado de reporte de rango por fecha de alumnos matriculados.

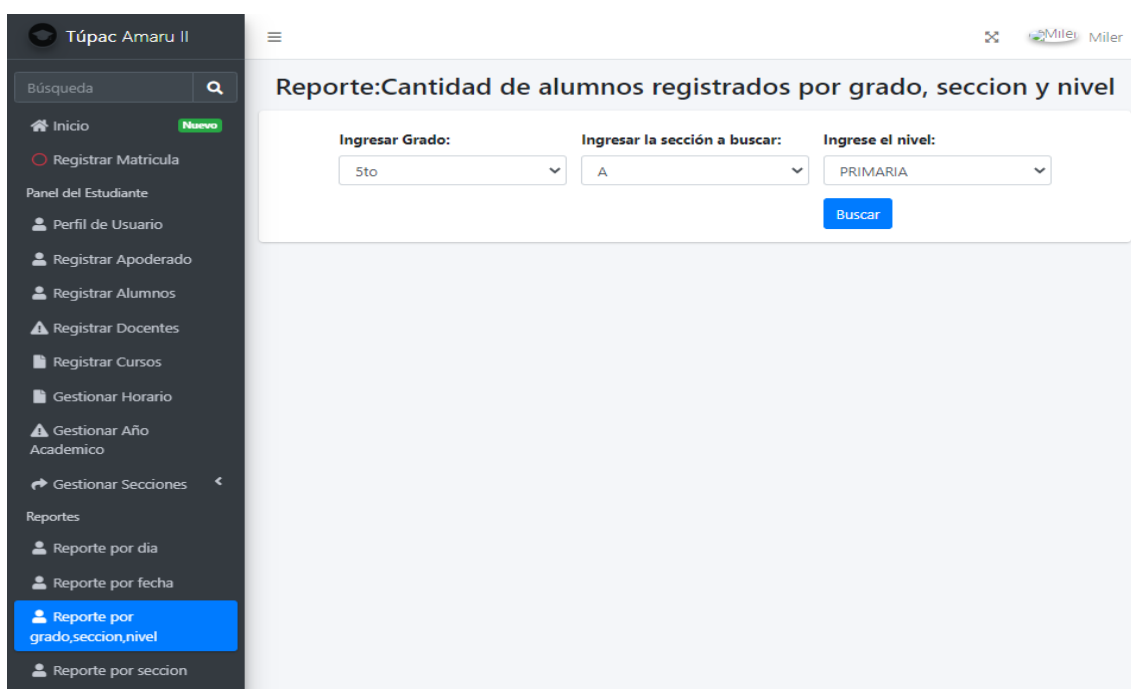
Figura 205: PDF de reportes por rango de fecha de alumnos matriculados.



Nota. Reporte de rango por fecha de alumnos matriculados PDF.

- ✓ **Reporte de listado de Alumnos por grado, sección y nivel:** al hacer clic en el botón de generar PDF, se obtendrá un resultado de un listado de alumnos por grado, sección y nivel registrados en PDF.

Figura 206: Buscar por grado, sección y nivel de alumnos matriculados.



Nota. Buscar por grado, sección y nivel de alumnos matriculados.

Figura 207: Reportes por grado, sección y nivel de alumnos matriculados.

The screenshot shows a web application interface for 'Túpac Amaru II'. The sidebar on the left contains navigation options such as 'Inicio', 'Registrar Matricula', 'Panel del Estudiante', and 'Reportes'. The main content area is titled 'Reporte de Grado por Alumno'. It displays 'Cantidad de alumnos registrados: 1' and a 'Descargar PDF' button. Below this is a table with the following data:

Grado Seleccionado	Sección	Nombre del alumno	Apellido del alumno	Nivel
5to	A	Miler	Alvarez Sarmineto	PRIMARIA

Nota. Reportes por grado, sección y nivel de alumnos matriculados registrados.

Figura 208: Descargar PDF de reportes por grado, sección y nivel de alumnos matriculados.

The screenshot shows a PDF viewer interface for 'matriculas (2).pdf'. The PDF content is titled 'Reporte de Grado por alumno 5to, Sección A y Nivel PRIMARIA'. It contains a table with the following data:

Grado seleccionado	Sección	Nombre	Apellido	Nivel
5to	A	Miler	Alvarez Sarmineto	PRIMARIA

Nota. Descargar PDF de reportes por grado, sección y nivel de alumnos matriculados registrados.

- ✓ **Reporte de listado de Alumnos por sección:** al hacer clic en el botón de reporte por sección, se obtendrá un resultado de un listado de alumnos por sección registrados.

Figura 209: *Buscar cantidad de secciones registradas por alumno.*

Reporte: Cantidad de secciones registradas por alumno

Ingresar el alumno a buscar:

Miler Alvarez Sarmiento

Buscar

Nota. *Buscar cantidad de secciones registradas por alumno.*

Figura 210: *Listado de cantidad de secciones registradas por alumno.*

Reporte de Matriculas de Miler

Cantidad de alumnos por sección:
1

ID	Alumno	Apellidos	Secciones
1	Miler	Alvarez Sarmiento	A

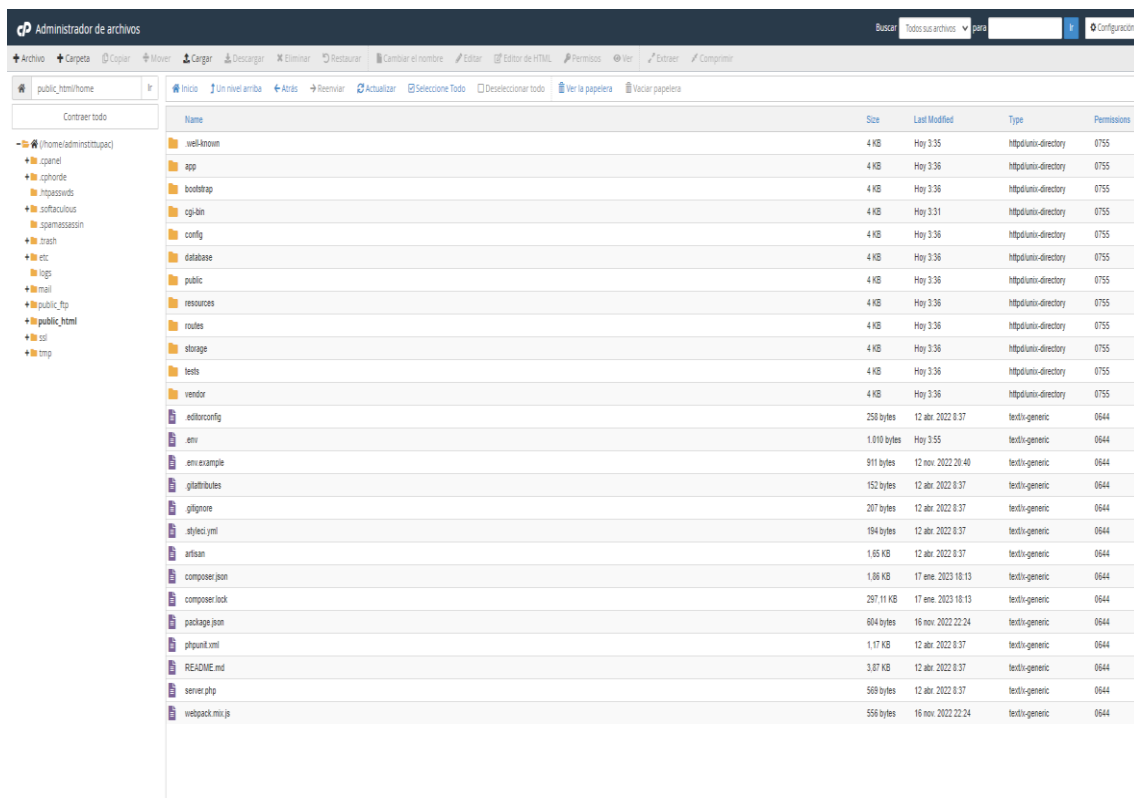
Nota. *Listado de cantidad de secciones registradas por alumno.*

Figura 211: Capturas de prueba de proyecto guardado en hosting y dominio en cPanel.





Nota. Cuenta de cPanel registrada.

Figura 212: Administrador de archivos cPanel.



Nota. Administrador de archivos cPanel.

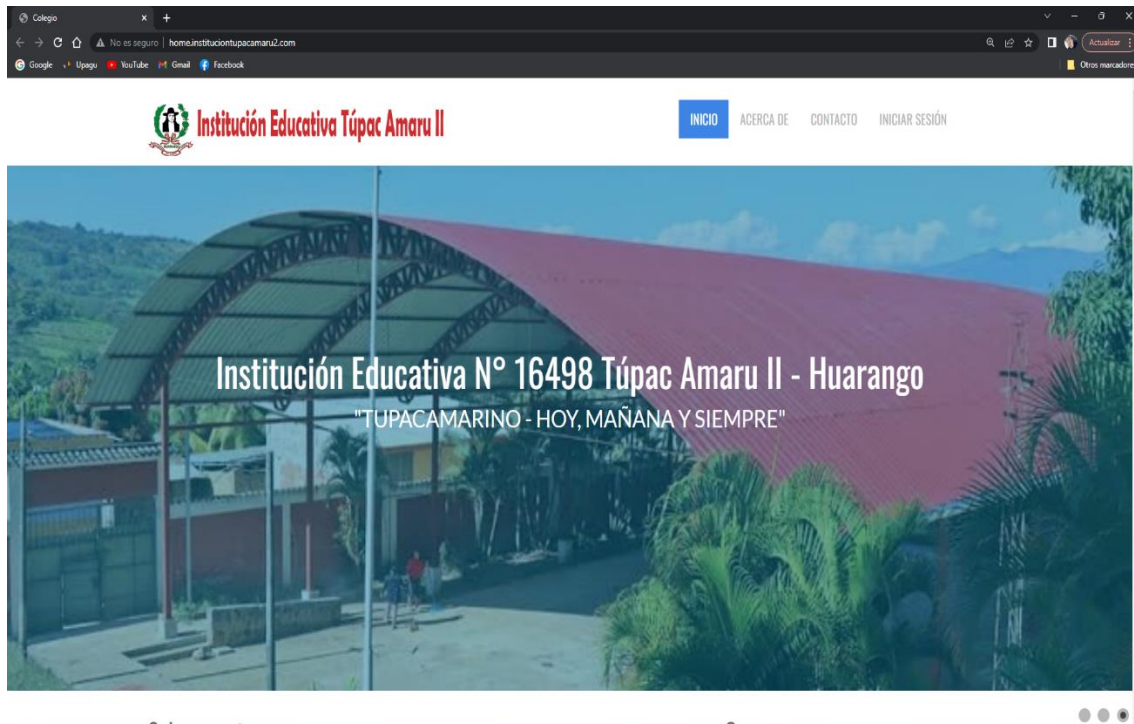
Figura 213: Información general.

Información General	
Usuario Actual	adminstittupac
Dominio Principal (DV Certificate)	 instituciontupacamaru2.com 
Shared IP Address	167.114.218.76
Directorio Principal	/home/adminstittupac
Last Login IP Address	190.116.90.123
Tema	<input type="text" value="paper_lantern"/>
Información del servidor	

Nota. Información general.

Figura 214: Link de ingreso (subdominio).

Link: home.instituciontupacamaru2.com



Nota. Link de ingreso (subdominio).

5. RESULTADOS Y DISCUSIÓN

5.1. RESULTADOS

5.1.1. Resultados del cuestionario del sitio web dirigido a expertos.

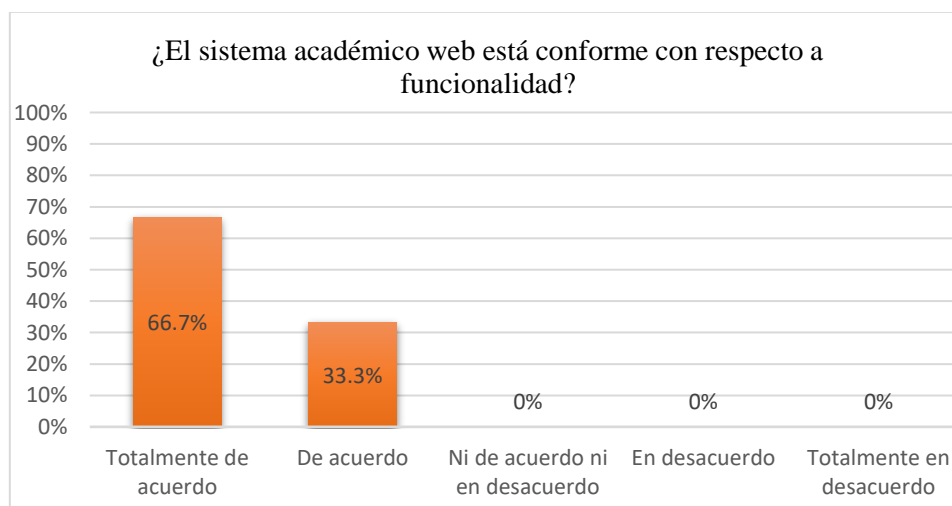
Se debe mencionar que hemos utilizado un cuestionario que fue validado por los expertos, este menciona los estándares de la norma ISO 25010. Esto que lleva a que podemos obtener la evaluación sobre el sistema académico web en la institución, se les genero algunas interrogantes:

Se obtuvieron datos por cada dimensión:

✓ **FUNCIONALIDAD**

En el gráfico N.º 1 se muestra los resultados a la pregunta: ¿El sistema académico web está conforme con respecto a funcionalidad? Se puede ver que el 66.7% de expertos encuestados consideran estar totalmente de acuerdo y el 33.3% están de acuerdo que el sistema académico web está conforme con respecto a su funcionalidad.

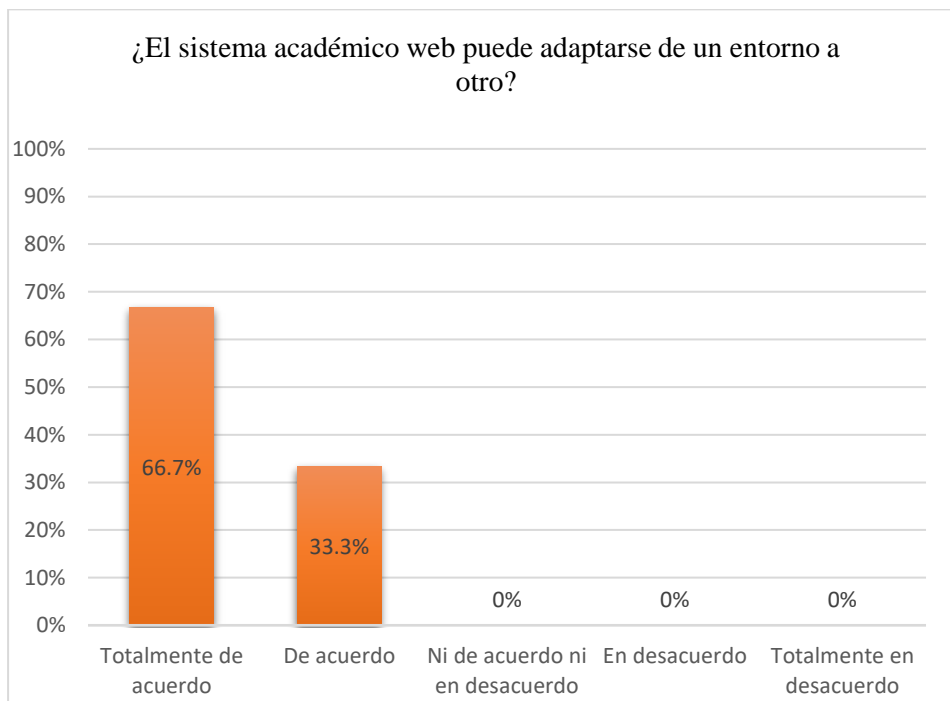
Gráfico 1: ¿El sistema académico web está conforme con respecto a funcionalidad?



PORTABILIDAD

En el gráfico N.º 2 se muestra los resultados a la pregunta: ¿El sistema académico web puede adaptarse de un entorno a otro? Se puede ver que el 66.7% de expertos encuestados consideran estar totalmente de acuerdo y el 33.3% están de acuerdo que el sistema académico web puede adaptarse de un entorno a otro.

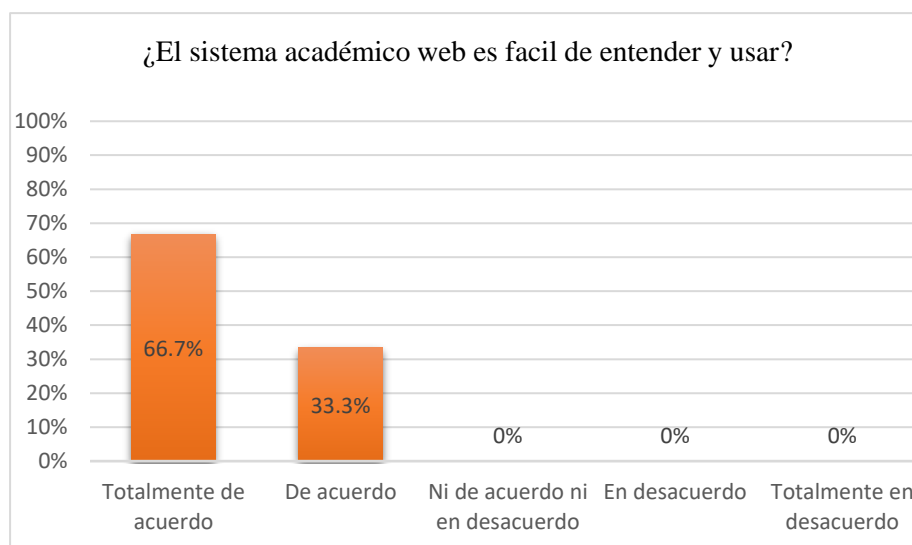
Gráfico 2: ¿El sistema académico web puede adaptarse de un entorno a otro?



USABILIDAD

En el gráfico N.º 3 se muestra los resultados a la pregunta: ¿El sistema académico web es fácil de entender y usar? Se puede ver que el 66.7% de expertos encuestados consideran estar totalmente de acuerdo y el 33.3% están de acuerdo de que el sistema académico web es fácil de entender y usar según el cuestionario realizado.

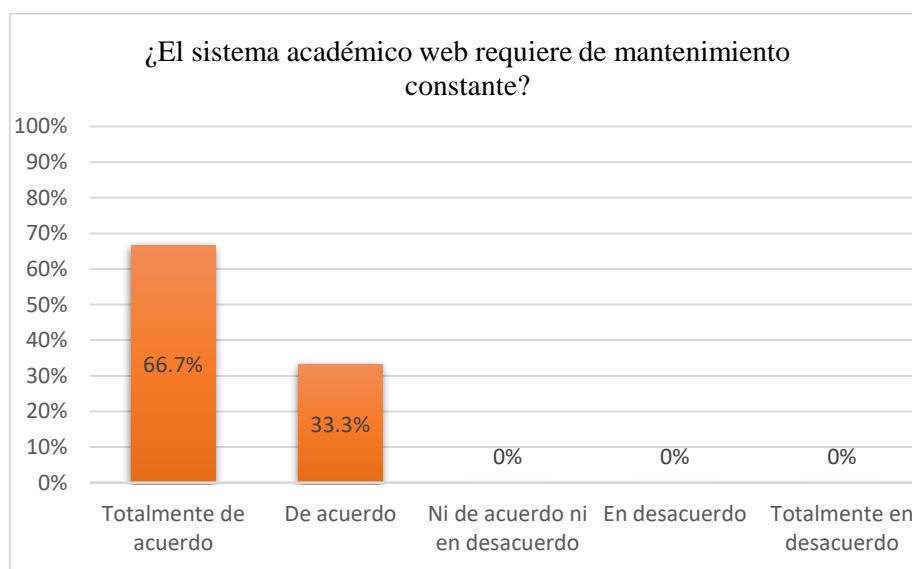
Gráfico 3: ¿El sistema académico web es fácil de entender y usar?



MANTENIBILIDAD

En el gráfico N.º 4 se muestra los resultados a la pregunta: ¿El sistema académico web requiere de mantenimiento constante? Se puede ver que el 66.7% de expertos encuestados consideran estar totalmente de acuerdo y el 33.3% están de acuerdo que el sistema académico web requiere de mantenimiento constante.

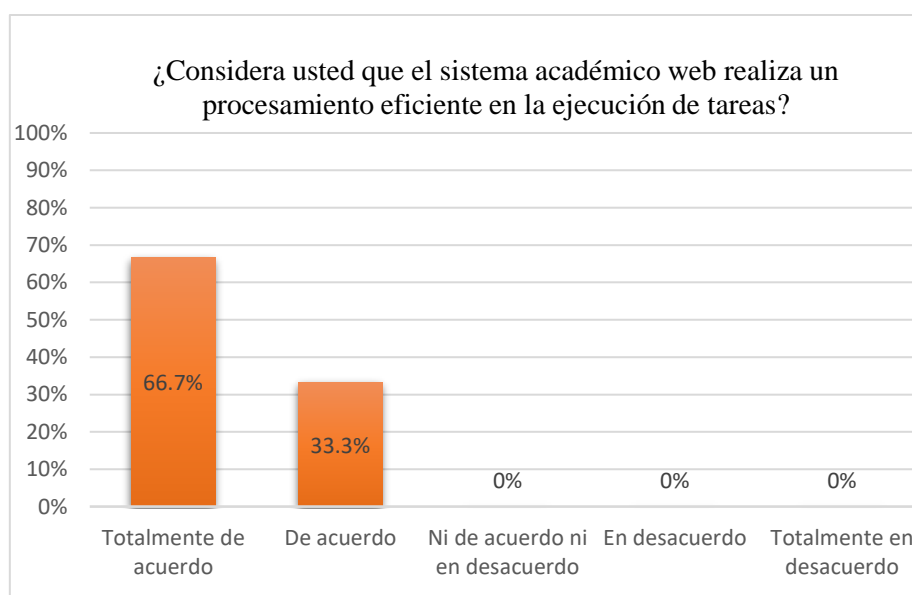
Gráfico 4: ¿El sistema académico web requiere de mantenimiento constante?



TIEMPO

En el gráfico N.º 5 se muestra los resultados a la pregunta: ¿Considera usted que el sistema académico web realiza un procesamiento eficiente en la ejecución de tareas? Se puede ver que el 66.7% de expertos encuestados consideran estar totalmente de acuerdo y el 33.3% están de acuerdo de que el sistema académico web realiza un procesamiento eficiente en la ejecución de tareas.

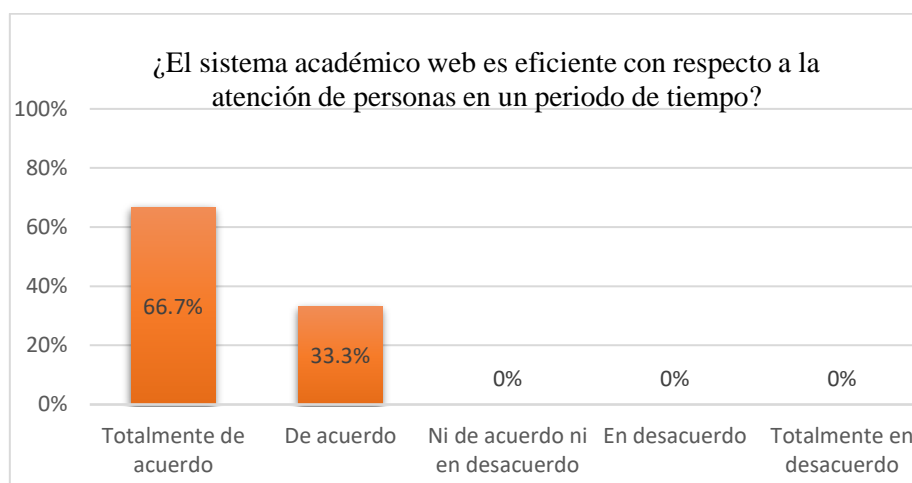
Gráfico 5: ¿Considera usted que el sistema académico web realiza un procesamiento eficiente en la ejecución de tareas?



EFICIENCIA

En el gráfico N.º 6 se muestra los resultados a la pregunta: ¿El sistema académico web es eficiente con respecto a la atención de personas en un periodo de tiempo? Se puede ver que el 66.7% de expertos encuestados consideran estar totalmente de acuerdo y el 33.3% están de acuerdo de que el sistema académico web es eficiente con respecto a la atención de personas en un periodo de tiempo.

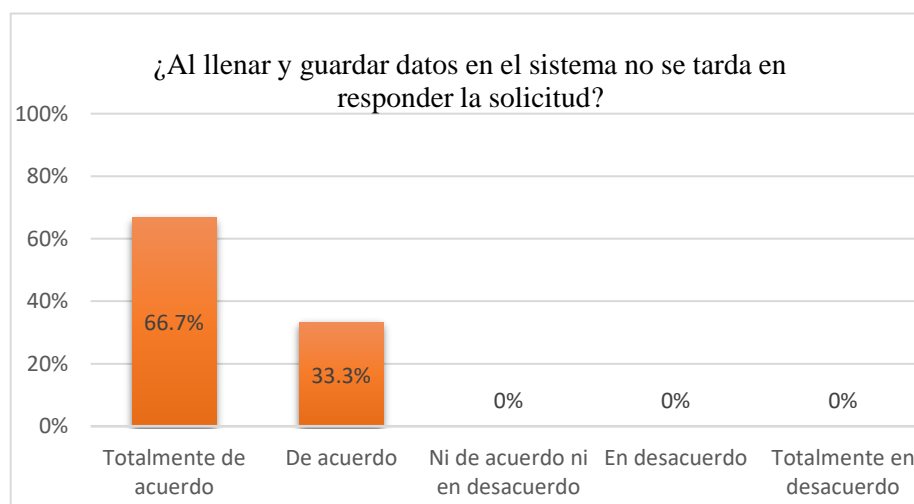
Gráfico 6: ¿El sistema académico web es eficiente con respecto a la atención de personas en un periodo de tiempo?



RELEVANCIA

En el gráfico N.º 7 se muestra los resultados a la pregunta: ¿Al llenar y guardar datos en el sistema no se tarda en responder la solicitud? Se puede ver que el 66.7% de expertos encuestados consideran estar totalmente de acuerdo y el 33.3% están de acuerdo de que el sistema académico web al llenar y guardar datos en el sistema no se tarda en responder la solicitud.

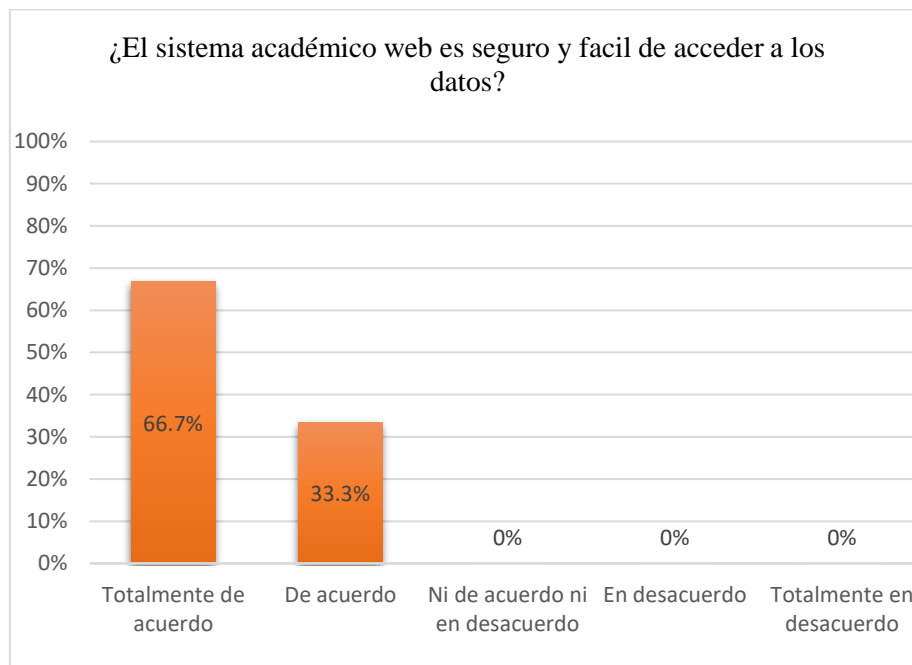
Gráfico 7: ¿Al llenar y guardar datos en el sistema no se tarda en responder la solicitud?



ORGANIZACIONAL

En el gráfico N.º 8 se muestra los resultados a la pregunta: ¿El sistema académico web es seguro y fácil de acceder a los datos? Se puede ver que el 66.7% de expertos encuestados consideran estar totalmente de acuerdo y el 33.3% de acuerdo de que el sistema académico web es seguro y fácil de acceder a los datos.

Gráfico 8: ¿El sistema académico web es seguro y fácil de acceder a los datos?



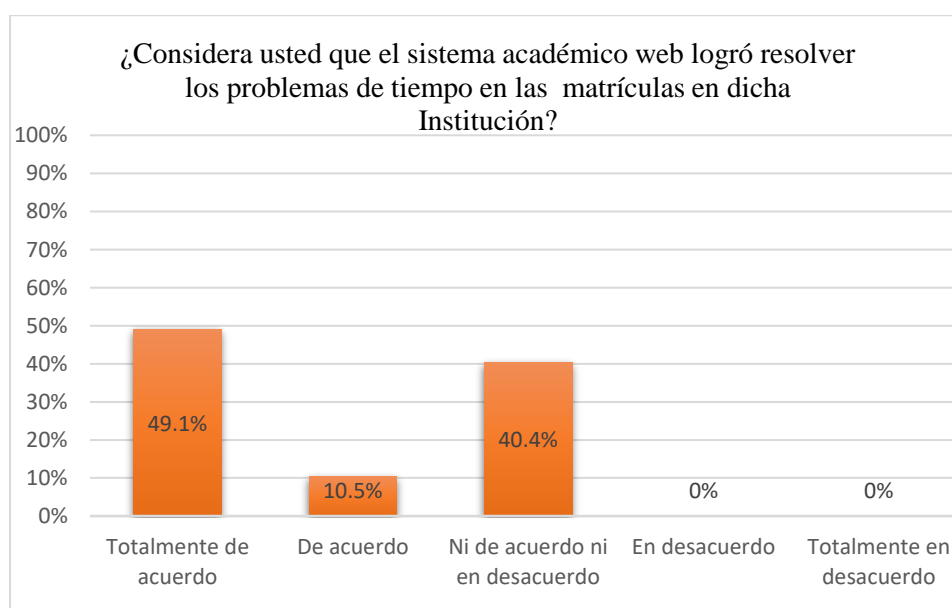
5.1.2. Resultados de cuestionario del sistema web dirigido a la comunidad educativa

Las preguntas fueron relacionadas a los padres de familia, la cual se obtuvo resultados que a continuación vamos a presentar:

TIEMPO

En el gráfico N.º 9 se muestra los resultados a la pregunta: ¿Considera usted que el sistema académico web logró resolver los problemas de tiempo en las matrículas en dicha Institución? Se puede ver que el 49.1% de docentes y padres de familia encuestados consideran estar totalmente de acuerdo y el 10.5% señalan estar de acuerdo mientras que el 40.4% están ni de acuerdo ni en desacuerdo, es por ello que se logra un valor positivo en cuanto al tiempo que tardaba haciéndolo manualmente y en comparación haciéndolo mediante una computadora una matrícula en la institución educativa Tupac Amaru II N.º 16498.

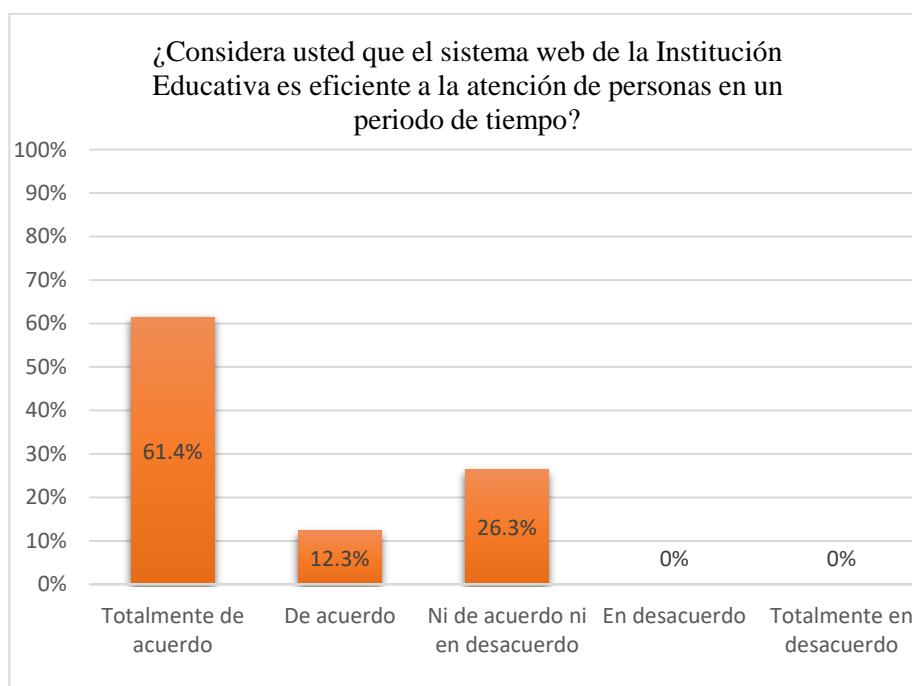
Gráfico 9: ¿Considera usted que el sistema académico web logró resolver los problemas de tiempo en las matrículas en dicha Institución?



EFICIENCIA

En el gráfico N.º 10 se muestra los resultados a la pregunta: ¿Considera usted que el sistema web de la Institución Educativa es eficiente a la atención de personas en un periodo de tiempo? Se puede ver que el 61.4% de docentes y padres de familia encuestados consideran estar totalmente de acuerdo y el 12.3% señalan estar de acuerdo mientras que el 26.3% están ni de acuerdo ni en desacuerdo, se obtiene un valor positivo ya que ahorra el tiempo y se realiza en un menor tiempo que haciendo la matrícula manualmente en la institución educativa Tupac Amaru II N° 16498.

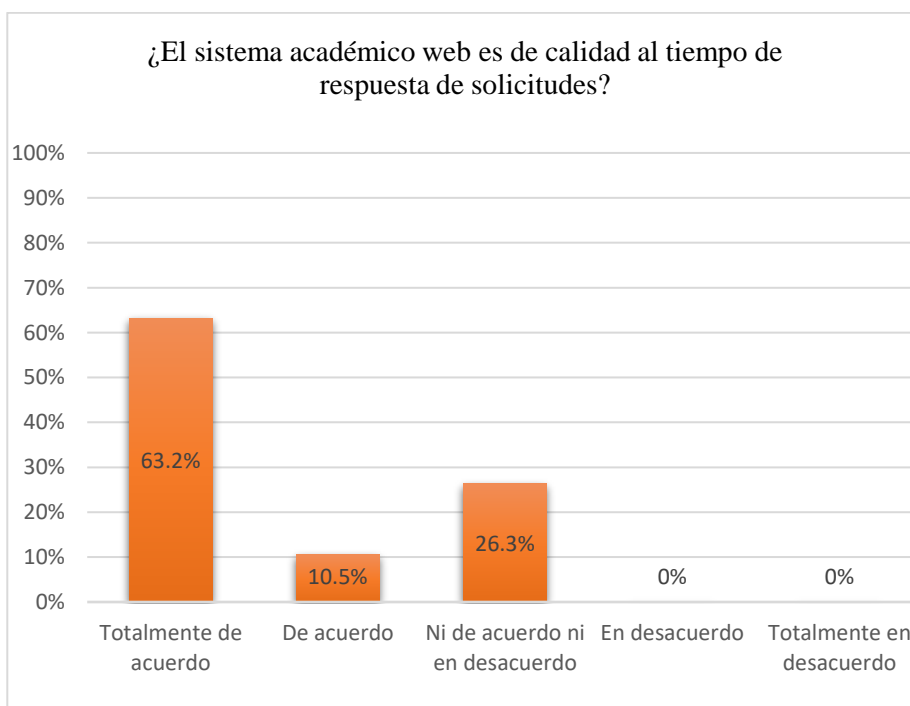
Gráfico 10: ¿Considera usted que el sistema web de la Institución Educativa es eficiente a la atención de personas en un periodo de tiempo?



RELEVANCIA

En el gráfico N.º 11 se muestra los resultados a la pregunta: ¿El sistema académico web es de calidad al tiempo de respuesta de solicitudes? Se puede ver que el 63.2% de docentes encuestados consideran estar totalmente de acuerdo y el 10.5% señalan estar de acuerdo mientras que el 26.3% están ni de acuerdo ni en desacuerdo, conlleva a que el sistema académico web ahorra tiempo necesario y da respuesta concisa al solicitar una respuesta por parte de una matrícula de un estudiante dentro de la institución educativa Tupac Amaru II N.º 16498.

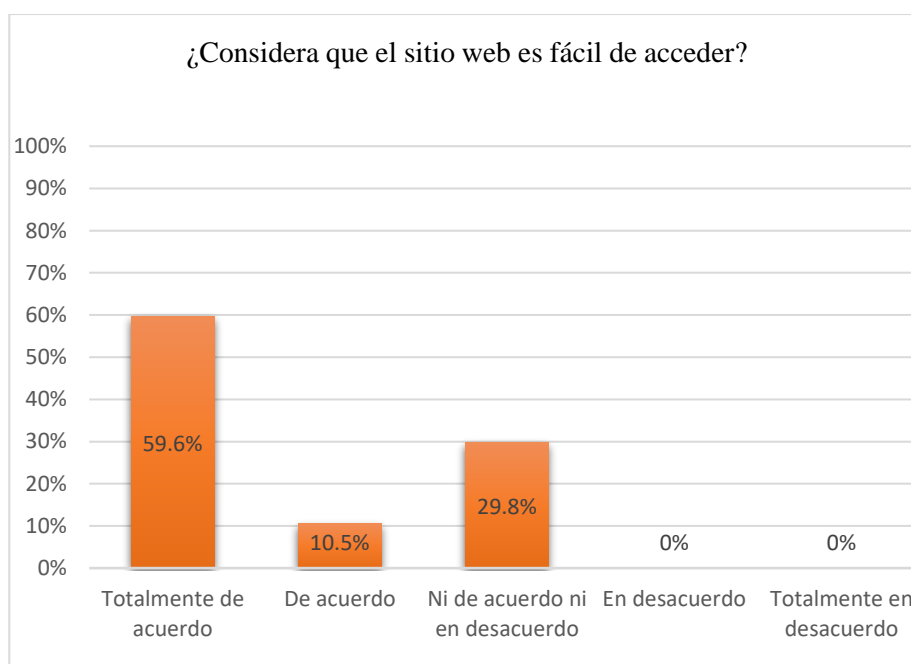
Gráfico 11: ¿El sistema académico web es de calidad al tiempo de respuesta de solicitudes?



ORGANIZACIONAL

En el gráfico N.º 12 se muestra los resultados a la pregunta: ¿Considera que el sitio web es fácil de acceder? Se puede ver que el 59.6% de docentes encuestados consideran estar totalmente de acuerdo y el 10.5% señalan estar de acuerdo mientras que el 29.8% están ni de acuerdo ni en desacuerdo, se obtiene un porcentaje llamativo que el sistema académico web es de fácil de acceder mediante el link de ingreso en la página de la institución educativa Tupac Amaru II N° 16498.

Gráfico 12: ¿Considera que el sitio web es fácil de acceder?



5.2. Contrastación de hipótesis

Para la hipótesis:

Sistema académico web y su influencia para la mejora en el proceso de matrículas en la institución educativa Tupac Amaru II – Huarango – San Ignacio – Cajamarca, se resolvió mediante la técnica de Chi-cuadrado o “ χ^2 ”, ya que esta facilita una medida de la discrepancia entre la frecuencia observada y la frecuencia esperada.

HIPOTESIS:

Modelo Lógico

H1: La influencia del sistema académico web mejoró a optimizar el tiempo de matrículas de la institución educativa Tupac Amaru II N° 16498, Huarango-San Ignacio-Cajamarca.

H0: La influencia del sistema académico web no mejoró a optimizar el tiempo de matrículas de la institución educativa Tupac Amaru II N° 16498, Huarango-San Ignacio-Cajamarca.

Modelo estadístico

$$X^2 = \sum \frac{(O - E)^2}{E}$$

Selección de nivel de significancia

Para la verificación de la hipótesis se utilizará el nivel de significancia “ α ” de 0.005 o 5% con un nivel de confiabilidad del 95%.

Especificación del estadístico

Para calcular el Chi cuadrado se recurre a la siguiente fórmula:

$$X^2 = \sum \frac{(O - E)^2}{E}$$

Donde:

X^2 = Chi – cuadrado, valor a calcularse

\sum = *Sumatoria*

O= Frecuencia observada, datos de la investigación

E= Frecuencia esperada.

Especificación de las regiones de aceptación y rechazo

Se llevará a cabo el proceso de determinación de los grados de libertad, enfatizando que el cuadro tiene 3 filas: las cuales son de acuerdo a la escala de Likert, Totalmente de acuerdo, De acuerdo y Ni de acuerdo ni en desacuerdo.

Se empleará la siguiente fórmula para determinar el grado de libertad:

$$gl = (K - 1)$$

Donde:

gl= grados de libertad.

K=Categorías o grupos a comparar.

$$gl = (K - 1)$$

$$gl = (3 - 1)$$

$$gl = 2$$

Resultó con 2 grados de libertad y un nivel de significancia de 0.005, y como respuesta de la tabla de chi-cuadrado se obtiene el valor de X^2 tabulado=5.9915.

Figura 215: *Tabla de chi-cuadrado, valor tabulado.*

v/p	0,001	0,0025	0,005	0,01	0,025	0,05	0,1	0,15
1	10,8274	9,1404	7,8794	6,6349	5,0239	3,8415	2,7055	2,0722
2	13,8150	11,9827	10,5965	9,2104	7,3778	5,9915	4,6052	3,7942
3	16,2660	14,3202	12,8381	11,3449	9,3484	7,8147	6,2514	5,3170
4	18,4662	16,4238	14,8602	13,2767	11,1433	9,4877	7,7794	6,7449
5	20,5147	18,3854	16,7496	15,0863	12,8325	11,0705	9,2363	8,1152
6	22,4574	20,2901	18,5474	16,9110	14,4494	12,5916	10,6446	9,4861

Nota. *Fuente: Elaboración propia.*

Recolectamos datos y cálculo de los estadísticos.

Análisis de variables

Frecuencias observadas

Tabla 30: *Frecuencia observada, datos de la investigación.*

	Totalmente de acuerdo	De acuerdo	Ni de acuerdo ni en desacuerdo	TOTAL
Muestra	34	22	4	60

Nota. *Fuente: Elaboración propia.*

Frecuencias esperadas

Tabla 31: *Frecuencia esperada.*

	Totalmente de acuerdo	De acuerdo	Ni de acuerdo ni en desacuerdo	TOTAL
Muestra	20	20	20	60

Nota. *Fuente: Elaboración propia.*

CÁLCULO DE CHI CUADRADO

X^2 Calculado = **22.80**

Tabla 32: *Cálculo de chi-cuadrado.*

Categoría	FO	FE	FO - FE	(FO - FE) ²	(FO - FE) ² / FE
Totalmente de acuerdo	34	20	14	196	9.8
De acuerdo	22	20	2	4	0.2
Ni de acuerdo ni en desacuerdo	4	20	-16	256	12.8
Total	60	60	0		22.8

Nota. Fuente: *Elaboración propia.*

REGLA DE DECISIÓN

Si X^2 tabulado $>$ X^2 calculado Se acepta H_0

Si X^2 tabulado $<$ X^2 calculado Se rechaza H_0 y se acepta H_1

El valor de X^2 Tabulado = 5,9915 $<$ X^2 Calculado = 22.80 y según en la regla de decisión, se rechaza la hipótesis nula y se acepta la hipótesis alternativa, con un nivel de confianza del 95% se confirma que el Sistema Académico Web mejoró a optimizar el tiempo de matrículas en la institución educativa Tupac Amaru II N° 16498, Huarango-San Ignacio-Cajamarca.

5.3. DISCUSIÓN DE RESULTADOS

De acuerdo a los resultados obtenidos por Díaz Fernández (2021) en su investigación **“Sistema de información para el proceso de matrículas, control de asistencias y gestión de notas académicas en la institución educativa departamental integrada Antonio Nariño del municipio de Apulo”**, tuvo como objetivo general Desarrollar e implementar un sistema de información donde se lleve a cabo los procesos de matrículas, control de asistencias y gestión de notas de los estudiantes para mejorar los procesos administrativos de la institución educativa Antonio Nariño del municipio de Apulo departamento de Cundinamarca, menciona que las matrículas, los datos de los estudiantes se realizaban de manera física y es por ello que se debe implementar un sistema web para mejorar el proceso de los datos y tenerlos guardados de manera confiable y segura. El sistema académico web posee una información detallada del proceso de matrículas, está de manera organizada, y se puede guardar de manera segura por lo que cuenta con el fácil acceso a ello.

De acuerdo a los resultados obtenidos por Guevara Ascoy (2021) en su investigación **“Diseño e implementación de un sistema de matrícula y agenda virtual para mejorar la eficiencia administrativa de la i.e. 10824 “san Nicolás”, Chiclayo 2022**, nos menciona en una de sus conclusiones que **“El sistema de matrícula y agenda virtual es un software de calidad, accesibilidad segura y manejo sencillo lo cual se logró un impacto positivo en los administrativos de la institución educativa 10824 “San Nicolás”**. El sistema académico web se logró cumplir para ayudar a priorizar las matrículas de la IE y ahorra tiempo necesario para realizar a cabo otras actividades, es de fácil

manejo y muy sencilla, es importante que cualquier encargado pueda realizar el aprendizaje de este sistema, cumpliendo con el objetivo propuesto de implementar un sistema académico web.

De acuerdo a los resultados obtenidos por Moyán y Palomino (2021) en su investigación **Implementación de un sitio web como medio de difusión de contenidos de la Institución Educativa N°028 “Santa Elena”- Cajamarca-2021**, tuvo como objetivo general la Implementación de un sitio web como medio de difusión de contenidos de la Institución Educativa, nos menciona en una de sus conclusiones que la implementación del sitio web es significativamente influyente ya que permite mostrar información más organizada, muestra contenido multimedia sin límite de almacenamiento, se accede desde cualquier dispositivo lo que facilita su acceso, logrando de esta manera que la Institución Educativa N°028 “Santa Elena” disponga de una herramienta tecnológica rápida, confiable y eficaz”. Así mismo el presente proyecto de investigación fue realizado para aprovechar las tecnologías de información y crear un sistema académico web para agilizar los procesos de matrículas en la IE de una forma más organizada posible.

Conforme a los resultados obtenidos por Osorio Álvarez (2016) en su investigación **“Diseño e implementación de un sistema de matrícula web usando software libre en el centro educativo “España”, distrito – Breña 2013”**, teniendo como objetivo general, Diseñar e implementar un sistema de matrícula web usando software libre en el centro educativo España del distrito

de Breña, nos dice en una de sus conclusiones que, “Se logró implementar el módulo de reportes con el cual el uso del sistema de matrícula web permite una mayor viabilidad, usabilidad y seguridad en la información, así como el uso de los recursos (hardware, software, trabajadores) en el centro educativo. De igual manera, se optimiza la toma de decisiones ya que cuenta con reportes personalizados”. En la presente investigación se realizó la implementación adecuada para realizar procesos de matrículas de los estudiantes y con ello lograr la satisfacción positiva del personal administrativo y contar con su propio sistema de la IE.

De acuerdo a los resultados obtenidos por Nava Alarcón (2018) con su tesis presentada **“Mejoramiento del proceso de control de pagos y matrícula de la institución educativa privada Ramón Castilla a través de un sistema de información desktop”**, nos dice en una de sus conclusiones que “Los requerimientos de información satisfechos mensualmente con el desarrollo e implementación del sistema de información desktop “Software a Medida v1.0 – Ramón Castilla” aumento en un 20% de 11 - 20 a 21 - 30 y 60% de 21 - 30 a más de 31; por lo tanto; la satisfacción del usuario ha mejorado sustancialmente en relación al manejo de información de matrículas y pagos; exactitud de la información y la satisfacción propiamente dicha de los medios utilizados para la obtención de la información de 60% a 100%”. En la presente investigación realizada y desarrollada se hicieron encuestas para probar la satisfacción de los padres de familia y personal administrativo, logrando un porcentaje adecuado de usabilidad adecuado de 31 personas de 60 estando totalmente de acuerdo (26) y de acuerdo (5) alcanzando un porcentaje de más

del 50 %. Lo cual aportamos un valor importante para la mejora en el proceso de matrículas.

Por otro lado Ortiz Garay, con su tesis presentada a la universidad nacional de Cajamarca, para optar por el título profesional de ingeniero de sistemas, titulada, *Influencia de un sistema de información web en la gestión académica de la institución educativa particular maría de Nazaret de Cajamarca*, nos dice en una de sus conclusiones que “En esta investigación no se han llegado a automatizar todos los procesos académicos, ya que se presentaron ciertas limitantes, como el hecho de que no se podía implementar una plataforma de pagos ya que requería costos adicionales, que la institución no creyó conveniente realizar, por cierta desconfianza en cuanto a la seguridad de los medios de pago. Esto solo demuestra que aún existe cierta resistencia a usar las tecnologías de la información, sobre todo, cuando el riesgo es un mayor y las pérdidas que se pueden presentar son económicas”. (Ortiz Garay, 2020, pág. 79).

De acuerdo a los resultados obtenidos por Ortiz Garay (2020) con su tesis presentada **Influencia de un sistema de información web en la gestión académica de la institución educativa particular maría de Nazaret de Cajamarca**, nos dice en una de sus conclusiones que “En esta investigación no se han llegado a automatizar todos los procesos académicos, ya que se presentaron ciertas limitantes, como el hecho de que no se podía implementar una plataforma de pagos ya que requería costos adicionales, que la institución

no creyó conveniente realizar, por cierta desconfianza en cuanto a la seguridad de los medios de pago. Esto solo demuestra que aún existe cierta resistencia a usar las tecnologías de la información, sobre todo, cuando el riesgo es un mayor y las pérdidas que se pueden presentar son económicas”. Así mismo en el presente proyecto se realizó la implementación correcta de procesos de matrículas, mejorando en primer lugar a la IE y segundo a los padres de familia al realizar la matrícula de sus hijos, ahorrando tiempo necesario y fácil de acceder y matricular para el personal administrativo.

6. CONCLUSIONES

- Se logró implementar el sistema académico web para la mejora en el proceso de matrículas en la “IE Túpac Amaru II N° 16498, Huarango – San Ignacio- Cajamarca, para ello hemos utilizado la programación XP, lo cual nos ha servido como base para la realización de la implementación, se usó diferentes programas como Laragon, Laravel (Framework de desarrollo), Composer (Gestor de Dependencias de PHP), PhpMyAdmin (Sistema gestor de base de datos), Sublime Text (editor de texto y editor de código fuente) y desarrollado bajo la arquitectura MVC (Modelo-Vista-Controlador), y por último se realizó una compra de un dominio y hosting por parte de los bachilleres para publicar los resultados del sistema académico web a internet, obteniendo un sitio web oficial.

- Se logró implementar el sistema académico web utilizando los requerimientos obtenidos mediante un listado, emitidas por el personal administrativo de la IE Tupac Amaru II, lo cual tomamos mucho énfasis en las necesidades que se estaban presentando, por último, logramos cumplir con todos los requerimientos y poderlos plasmar en el sistema académico web obteniendo un resultado positivo (ANEXO IV).

- Se logró la validación del sistema académico web mediante la aplicación de cuestionarios dirigidos a expertos lo cual están las dimensiones funcionalidad, portabilidad, usabilidad y mantenibilidad (ANEXO II).

7. RECOMENDACIONES

- Al personal administrativo futuro tener en cuenta que la Institución Educativa cuenta con su propio sistema académico web para realizar procesos de matrículas de cursos y docentes.
- Al director de la Institución Educativa, aprovechar el sistema académico web en el próximo año al matricular a los estudiantes.
- Tener en cuenta las diferentes herramientas que nos ofrece la tecnología para realizar nuestras actividades académicas y llevarlas de forma dinámica para los estudiantes.
- Una recomendación importante es la actualización de los datos de los docentes en el sistema académico web y el personal en general.
- Realizar un manejo constante del sistema académico web para complementar mejoras posibles.
- Tomar en cuenta que el sistema académico web se abre la venta en un dispositivo móvil y en una Pc (Computadora Personal).
- Comunicar si existe alguna duda o explicación al personal que llevo a cabo la creación del sistema.
- Se recomienda al personal administrativo realizar el guardado de boletas que emita el sistema en una carpeta segura.

8. REFERENCIAS

(s.f.). Obtenido de grupoconsultorefe:
<https://grupoconsultorefe.com/servicio/tecnologias-de-la-informacion/sistemas-web>

(s.f.). Obtenido de <https://tiposinvestigacion.com/instrumentos-investigacion/>

(s.f.). Obtenido de <https://ima.udg.edu/~sellares/einf-es2/present1011/metodopesadesrup.pdf>

Ahumada, J. (s.f.). *controlacademic*. Obtenido de <https://controlacademic.com.co/?p=1327>

Alan Neill, David; Cortez Suárez, Liliana;. (2017). *Procesos y Fundamentos de la Investigación Científica*. Ecuador : Utmach.

apen. (s.f.). Obtenido de apen: <https://apen.es/glosario-de-informatica/microsoft-excel/>

applicatta. (s.f.). Obtenido de <https://www.applicatta.cl/index.php/soluciones/metodologia-applicatta/implementacion-de-software>

asmarketero. (s.f.). Obtenido de <https://www.asmarketero.com/web-para-colegios-peru/#:~:text=Un%20sistema%20web%20para%20colegios,mayor%20disponibilidad%20y%20ahorros%20sustanciales>.

Calvo, D. (07 de Abril de 2018). *diegocalvo*. Obtenido de diegocalvo: <https://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/>

Celina Oviedo, Heidi; CampoArias, Adalberto;. (2005). *Aproximación al uso del coeficiente alfa*. colombia .

Choquesillo Sánchez, F. A. (2012). *DESARROLLO E IMPLEMENTACION DE UN PORTAL WEB COMO ALTERNATIVA DE SOLUCION PARA MEJORAR LA CALIDAD DE SERVICIO DEL CEMENTERIO GENERAL EN LA PROVINCIA DE CHINCHA*. Chicha: Universidad Privada Alda Byron S.AC.

clumsygames. (s.f.). Obtenido de clumsygames:
<https://clumsygames.com/diccionario-gamer/admin/>

Coefficiente de correlación lineal de pearson. (s.f.).

Coppola, M. (01 de Agosto de 2022). *Hubspot*. Obtenido de Hubspo:
<https://blog.hubspot.es/website/que-es-servidor-web>

Diaz Fernandez, J. M. (2021). *Sistema de información para el proceso de matrículas, control de asistencias y gestión de notas académicas en la institución educativa departamental integrada Antonio Nariño del municipio de Apulo*. Colombia: unipiloto.

Galán Amador, M. (29 de mayo de 2009). Obtenido de
<https://manuelgalan.blogspot.com/2009/05/la-entrevista-en-investigacion.html>

García Muñoz, T. (2003). *El cuestionario como instrumento de investigación/evaluación*.

Gómez Fuentes, M. d. (2013). *Base de datos*. México: UNIVERSIDAD AUTÓNOMA METROPOLITANA.

Guevara Ascoy , C. B. (2021). *Diseño e implementación de un sistema de matrícula y agenda virtual para mejorar la eficiencia administrativa de la i.e. 10824 “san Nicolás”, Chiclayo 2021*. Colombia : umb.

Howard White;Shagun Sabarwal . (2014). *Diseño y métodos cuasiexperimentales*.

Iso. (2022). *iso25000*. Obtenido de iso25000:
<https://iso25000.com/index.php/normas-iso-25000/iso-25010>

iso25000. (2022). Obtenido de iso25000: <https://iso25000.com/index.php/normas-iso-25000>

Jimeno Flores, J. V., & Visitación Castillo, R. R. (2019). “*DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA GESTIÓN DEL FLUJO DE INFORMACIÓN EN EL TALLER AUTOMOTRIZ AUTOSERVICIOS AGUILAR*”. Lima: Universidad Tecnológica del Perú.

Juan. (s.f.). *Mauronet*. Obtenido de <https://www.mauronet.com/>

laragon. (2019). Obtenido de laragon: <https://laragon.org/docs/index.html>

López Fuentes , F. d. (2015). *Sistemas Distribuidos* . colombia .

López Menéndez de Jiménez, R. E. (s.f.). *Metodologías Ágiles de Desarrollo de Software Aplicadas a la*.

Maldonado Luna, S. M. (2007). *Manual práctico para el diseño de la Escala Likert*.

ministerio de educacion . (11 de Marzo de 2022). *minedu*. Obtenido de minedu:
<https://sites.minedu.gob.pe/sigec/2022/03/01/cual-es-la-finalidad-del-proceso-de-matricula/>

- Nava Alarcón , G. F. (2018). *Mejoramiento del proceso de control de pagos y matrícula de la institución educativa privada Ramón Castilla a través de un sistema de información desktop*. Cajamarca : unc.
- Ortiz Garay, H. G. (2020). *Influencia de un sistema de información web en la gestión académica de la institución educativa particular maría de Nazaret de Cajamarca*. Cajamarca: unc.
- Osorio Álvarez, N. Á. (2016). *Diseño e implementación de un sistema de matrícula web usando software libre en el centro educativo “España”, distrito – breña 2013*. Lima: uch.
- Pérez Porto, J., & Merino, M. (24 de Julio de 2022). *Implementar*. Obtenido de <https://definicion.de/implementar/>
- php*. (2023). Obtenido de php: <https://www.php.net/manual/es/intro-what-is.php>
- Proaño Castro, Milton Felipe; Orellana Contreras, Shirley Yésica; Martillo Pazmiño, Italo Omar;. (2018). *Los sistemas de información y su importancia en la transformación digital de la empresa actual*.
- questionpro*. (2023). Obtenido de *questionpro*: <https://www.questionpro.com/es/una-encuesta.html>
- Quevedo Ricardi, F. (01 de diciembre de 2011). *medwave*. Obtenido de medwave: <http://doi.org/10.5867/medwave.2011.12.5266>
- Rodríguez Carbajal , A. D. (2019). *Sistema académico web y la mejora en el proceso de matrículas y pagos de la institución educativa Virgen de la Asunción del Porvenir – Trujillo*. Trujillo : Biblioteca Digital.
- Rovira, S. (2022). *Tecnologías digitales para un nuevo futuro* . Santiago .

Stsepanets, A. (17 de enero de 2023). *ganttpro*. Obtenido de ganttpro:

<https://blog.ganttpro.com/es/metodologia-de-cascada/>

Torres, P. (2023). *scribd*. Obtenido de scribd.

Vargas Cordero, Z. R. (2009). *La investigación aplicada: una forma de conocer las realidades con evidencia*. Costa Rica.

ANEXOS

ANEXO 1: VALIDACIÓN DE INSTRUMENTO

✓ VALIDACIÓN N° 1

✓

VALIDACIÓN DE INSTRUMENTO

1. Identificación del experto.

Nombres y Apellidos: Karim Ivette Cruzado Villar

Título profesional: Ingeniero de Sistemas

Grado: Doctora en Administración

DNI: 26706593

Fecha: 26/04/2023

2. Título de la investigación.

Sistema académico web y su influencia para la mejora en el proceso de matrículas en la Institución educativa Tupac Amaru II Huarango – San Ignacio – Cajamarca.

3. Instrucciones.

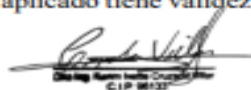
Estimado(a) especialista, para evaluar dicho instrumento, marcar con un aspa (x) una de las categorías contempladas en el siguiente cuadro.

1: Deficiente 2: Regular 3: Bueno 4: Muy Bueno 5: Excelente

4. Juicio de expertos.

INDICADORES	CRITERIOS	CATEGORÍA				
		1	2	3	4	5
1	CLARIDAD					X
2	OBJETIVIDAD					X
3	ACTUALIDAD					X
4	ORGANIZACIÓN					X
5	SUFICIENCIA					X
6	INTENCIONALIDAD					X
7	CONSISTENCIA					X
8	COHERENCIA					X
9	METODOLOGÍA					X
10	CONVENIENCIA					X
Puntaje parcial						50
Puntaje total		50				

Mediante la presente, fidelizo que lo aplicado tiene validez y confiabilidad. Según el juicio de mi experiencia profesional.



KARIM IVETTE CRUZADO VILLAR
CIP 8132

FIRMA DEL EXPERTO

✓ VALIDACIÓN N° 2

VALIDACIÓN DE INSTRUMENTO

1. Identificación del experto.

Nombres y Apellidos: FREDDY WILMER CERVERA ESTELA

Título profesional: INGENIERO INFORMÁTICO Y DE SISTEMAS

Grado: MAGISTER

DNI: 26685133

Fecha: 25/04/2023

2. Título de la investigación.

Sistema académico web y su influencia para la mejora en el proceso de matrículas en la Institución educativa Tupac Amaru II Huarango – San Ignacio – Cajamarca.

3. Instrucciones.

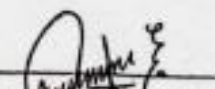
Estimado(a) especialista, para evaluar dicho instrumento, marcar con un aspa (x) una de las categorías contempladas en el siguiente cuadro.

1: Deficiente 2: Regular 3: Bueno 4: Muy Bueno 5: Excelente

4. Juicio de expertos.

	INDICADORES	CRITERIOS	CATEGORÍA				
			1	2	3	4	5
1	CLARIDAD	Esta formulado con lenguaje claro y apropiado.					X
2	OBJETIVIDAD	Cumple con el criterio de objetividad.					X
3	ACTUALIDAD	Instrumento actualizado.					X
4	ORGANIZACIÓN	Se muestra de manera organizada.					X
5	SUFICIENCIA	El instrumento es suficiente para que se mida el objetivo.					X
6	INTENCIONALIDAD	Adecuado para valorar los aspectos del estudio.					X
7	CONSISTENCIA	Basado en aspectos teóricos y científicos respecto al tema de estudio.					X
8	COHERENCIA	Está formulado correctamente.					X
9	METODOLOGÍA	La planificación responde al propósito del estudio.					X
10	CONVENIENCIA	Las preguntas no constituyen riesgo para el encuestado.					X
Puntaje parcial							50
Puntaje total							50

Mediante la presente, fidelizo que lo aplicado tiene validez y confiabilidad. Según el juicio de mi experiencia profesional.


FIRMA DEL EXPERTO
 Freddy Wilmer Cervera Estela
 INGENIERO INFORMÁTICO Y DE SISTEMAS
 Reg. CIP N° 138166

✓ VALIDACIÓN N° 3

VALIDACIÓN DE INSTRUMENTO

1. Identificación del experto.

Nombres y Apellidos: Evelyn Soneth Gutierrez Fernandez

Título profesional: Ing. sistemas

Grado: Magister en Administración

DNI: 41151916

Fecha: 29-04-2023

2. Título de la investigación.

Sistema académico web y su influencia para la mejora en el proceso de matrículas en la Institución educativa Tupac Amaru II Huarango – San Ignacio – Cajamarca.

3. Instrucciones.

Estimado(a) especialista, para evaluar dicho instrumento, marcar con un aspa (x) una de las categorías contempladas en el siguiente cuadro.

1: Deficiente 2: Regular 3: Bueno 4: Muy Bueno 5: Excelente

4. Juicio de expertos.

INDICADORES	CRITERIOS	CATEGORÍA				
		1	2	3	4	5
1	CLARIDAD					X
2	OBJETIVIDAD				X	
3	ACTUALIDAD				X	
4	ORGANIZACIÓN				X	
5	SUFICIENCIA				X	
6	INTENCIONALIDAD				X	
7	CONSISTENCIA				X	
8	COHERENCIA				X	
9	METODOLOGÍA				X	
10	CONVENIENCIA					X
Puntaje parcial					32	10
Puntaje total					42	

Mediante la presente, fidelizo que lo aplicado tiene validez y confiabilidad. Según el juicio de mi experiencia profesional.


FIRMA DEL EXPERTO

ANEXO II: CUESTIONARIO DIRIGIDO A EXPERTOS

✓ EXPERTO N° 1

CUESTIONARIO DIRIGIDO A EXPERTOS

Buen día estimado(a), solicitamos su colaboración para una investigación sobre: Sistema académico web y su influencia para la mejora en el proceso de matrículas en la institución educativa Tupac Amaru II – Huarango – San Ignacio – Cajamarca.

Su participación es necesaria para conseguir los objetivos de este estudio. En tal sentido, le solicitamos su apreciada colaboración para que responda a cada una de las preguntas, con la finalidad de validar el sistema académico web.

DATOS DEL EXPERTO:

Nombres y Apellidos: Karim Ivette Cruzado Villar

Título Profesional: Ingeniero de Sistemas

DNI: 26706593

Fecha: 26/04/2023

INSTRUCCIONES: Marque una sola alternativa con una "X" en las siguientes preguntas:

• **FUNCIONALIDAD**

¿El sistema académico web está conforme con respecto a funcionalidad?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- Totalmente de acuerdo.

• **PORTABILIDAD**

¿El sistema académico web puede adaptarse de un entorno a otro?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- Totalmente de acuerdo.

- **USABILIDAD**

¿El sistema académico web es fácil de entender y usar?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- Totalmente de acuerdo.

- **MANTENIBILIDAD**

¿El sistema académico web requiere de mantenimiento constante?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- Totalmente de acuerdo.

- **TIEMPO**

¿Considera usted que el sistema académico web realiza un procesamiento eficiente en la ejecución de tareas?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- Totalmente de acuerdo.

- **EFICIENCIA**

¿El sistema académico web es eficiente con respecto a la atención de personas en un periodo de tiempo?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- Totalmente de acuerdo.

- **RELEVANCIA**

¿Al llenar y guardar datos en el sistema no se tarda en responder la solicitud?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- Totalmente de acuerdo.

- **ORGANIZACIONAL**

¿El sistema académico web es seguro y fácil de acceder a los datos?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- Totalmente de acuerdo.

Mediante la presente, fidelizo que lo aplicado tiene validez y confiabilidad. Según el juicio de mi experiencia profesional.



FIRMA DEL EXPERTO

✓ VALIDACIÓN N° 2

CUESTIONARIO DIRIGIDO A EXPERTOS

Buen día estimado(a), solicitamos su colaboración para una investigación sobre: Sistema académico web y su influencia para la mejora en el proceso de matrículas en la institución educativa Tupac Amaru II – Huarango – San Ignacio – Cajamarca.

Su participación es necesaria para conseguir los objetivos de este estudio. En tal sentido, le solicitamos su apreciada colaboración para que responda a cada una de las preguntas, con la finalidad de validar el sistema académico web.

DATOS DEL EXPERTO:

Nombres y Apellidos: FREDDY WILMER CERVERA ESTELA

Título Profesional: INGENIERO INFORMÁTICO Y DE SISTEMAS

DNI: 26685133

Fecha: 25/04/2023

INSTRUCCIONES: Marque una sola alternativa con una "X" en las siguientes preguntas:

- **FUNCIONALIDAD**

¿El sistema académico web está conforme con respecto a funcionalidad?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **PORTABILIDAD**

¿El sistema académico web puede adaptarse de un entorno a otro?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **USABILIDAD**

¿El sistema académico web es fácil de entender y usar?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **MANTENIBILIDAD**

¿El sistema académico web requiere de mantenimiento constante?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **TIEMPO**

¿Considera usted que el sistema académico web realiza un procesamiento eficiente en la ejecución de tareas?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **EFICIENCIA**

¿El sistema académico web es eficiente con respecto a la atención de personas en un periodo de tiempo?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **RELEVANCIA**

¿Al llenar y guardar datos en el sistema no se tarda en responder la solicitud?


- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **ORGANIZACIONAL**

¿El sistema académico web es seguro y fácil de acceder a los datos?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

Mediante la presente, fidelizo que lo aplicado tiene validez y confiabilidad. Según el juicio de mi experiencia profesional.



Freddy Wilner Corvera Estela
INGENIERO INFORMÁTICA Y DE SISTEMAS
Reg. CIP N° 130166

FIRMA DEL EXPERTO

✓ VALIDACIÓN N° 3

CUESTIONARIO DIRIGIDO A EXPERTOS

Buen día estimado(a), solicitamos su colaboración para una investigación sobre: Sistema académico web y su influencia para la mejora en el proceso de matriculas en la institución educativa Tupac Amaru II – Huarango – San Ignacio – Cajamarca.

Su participación es necesaria para conseguir los objetivos de este estudio. En tal sentido, le solicitamos su apreciada colaboración para que responda a cada una de las preguntas, con la finalidad de validar el sistema académico web.

DATOS DEL EXPERTO:

Nombres y Apellidos: Evelyn Joneth Gutierrez Fernandez

Título Profesional: Ing. de Sistemas

DNI: 41151916

Fecha: 29-04-2023

INSTRUCCIONES: Marque una sola alternativa con una "X" en las siguientes preguntas:

• FUNCIONALIDAD

¿El sistema académico web está conforme con respecto a funcionalidad?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

• PORTABILIDAD

¿El sistema académico web puede adaptarse de un entorno a otro?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **USABILIDAD**

¿El sistema académico web es fácil de entender y usar?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **MANTENIBILIDAD**

¿El sistema académico web requiere de mantenimiento constante?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **TIEMPO**

¿Considera usted que el sistema académico web realiza un procesamiento eficiente en la ejecución de tareas?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **EFICIENCIA**

¿El sistema académico web es eficiente con respecto a la atención de personas en un periodo de tiempo?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **RELEVANCIA**

¿Al llenar y guardar datos en el sistema no se tarda en responder la solicitud?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

- **ORGANIZACIONAL**

¿El sistema académico web es seguro y fácil de acceder a los datos?

- 1) Totalmente en desacuerdo.
- 2) En desacuerdo.
- 3) Ni de acuerdo ni en desacuerdo.
- 4) De acuerdo.
- 5) Totalmente de acuerdo.

Mediante la presente, fidelizo que lo aplicado tiene validez y confiabilidad. Según el juicio de mi experiencia profesional.



Ing. Evelyn J. Gutiérrez Fernández
CIP. 100051

FIRMA DEL EXPERTO

ANEXO III: CUESTIONARIO DIRIGIDO A LOS PADRES DE FAMILIA Y PERSONAL ADMINISTRATIVO

INSTITUCIÓN EDUCATIVA TUPAC AMARU II N° 16498



INSTRUCCIONES

- Las siguientes preguntas van dirigidas al personal de la institución educativa Tupac Amaru II N° 16498.
- El presente cuestionario busca determinar un fin de estudio a evaluar por los bachilleres postulantes.
- Leer cada pregunta y responderla marcando su respuesta dependiendo al logro obtenido.

¿Considera usted que el sistema académico web logró resolver los problemas de tiempo en las matrículas en dicha Institución? *

- Totalmente en desacuerdo
- En desacuerdo.
- Ni de acuerdo ni en desacuerdo.
- De acuerdo.
- Totalmente de acuerdo.

¿Considera usted que el sistema web de la Institución Educativa es eficiente a la atención de personas en un periodo de tiempo? *

- Totalmente en desacuerdo.
- En desacuerdo.
- Ni de acuerdo ni en desacuerdo.
- De acuerdo.
- Totalmente de acuerdo.

¿El sistema académico web es de calidad al tiempo de respuesta de solicitudes? *

- Totalmente en desacuerdo.
- En desacuerdo.
- Ni de acuerdo ni en desacuerdo.
- De acuerdo.
- Totalmente de acuerdo.

¿Considera que el sitio web es fácil de acceder? *

- Totalmente en desacuerdo.
- En desacuerdo.
- Ni de acuerdo ni en desacuerdo.
- De acuerdo.
- Totalmente de acuerdo.

ANEXO IV: LISTADO DE REQUERIMIENTOS DADAS POR PARTE DE LA IE TUPAC AMARU II

SISTEMA ACADÉMICO WEB Y SU INFLUENCIA PARA LA MEJORA EN EL PROCESO DE MATRÍCULAS EN LA INSTITUCIÓN EDUCATIVA TÚPAC AMARU II – HUARANGO –
SAN IGNACIO – CAJAMARCA

REQUERIMIENTOS FUNCIONALES

ID	REQUERIMIENTO	DESCRIPCIÓN	OBSERVACIONES
RF-01	Iniciar sesión	El usuario podrá ingresar al sistema mediante un inicio de sesión mediante un correo y una contraseña.	El sistema proporcionará un correo y contraseña para el inicio de sesión al usuario.
RF-03	Cambiar contraseña	El sistema debe permitir cambiar contraseña de acceso.	El sistema enviará la nueva contraseña al correo electrónico.
RF-04	Matricular alumno(a)	El sistema permitirá matricular al alumno(a) registrando sus datos.	El sistema guardará los datos respectivos del alumno(a).
RF-05	Modificar alumno(a)	El sistema permitirá modificar sus datos al alumno(a).	El sistema podrá modificar los datos del alumno(a) en caso hubiera incorrecciones.
RF-06	Eliminar alumno(a)	El sistema podrá eliminar la información del alumno(a).	En el sistema se podrá eliminar los datos del alumno(a).
RF-07	Buscar alumno(a)	El sistema será capaz de buscar al alumno en sus registros.	El sistema podrá realizar la búsqueda del alumno(a).
RF-08	Listar alumnos(as)	El sistema será capaz de visualizar la lista de alumnos(as).	En el sistema se podrá visualizar una lista de todos los alumnos.
RF-09	Reporte de registro de alumnos matriculados	El sistema hará un reporte de alumnos matriculados.	El administrador podrá imprimir un reporte de alumnos matriculados.
RF-10	Reporte de registro de docentes por cada curso	El sistema hará un reporte de docentes registrados por cada curso.	El administrador podrá imprimir un reporte de docentes registrados por cada curso.
RF-11	Reporte de alumnos registrados por cada grado, sección y año	El sistema hará un reporte de alumnos registrados por cada grado, sección y año.	El administrador podrá imprimir un reporte de alumnos registrados por cada grado, sección y año.
RF-12	Acceso al administrador a una ventana personal	El sistema permitirá poder ingresar a una ventana personal para el administrador.	El sistema podrá mostrar una pantalla diferente con todos los accesos para el administrador.
RF-13	Acceso al tesorero a una ventana personal	El sistema permitirá poder ingresar a una ventana personal para el tesorero.	El sistema podrá mostrar una pantalla diferente con algunos accesos para el tesorero o encargado.

ANEXO V: REUNIONES CON EL PERSONAL ADMINISTRATIVO

