



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Tema:

**SISTEMA WEB BASADO EN LA ARQUITECTURA MODELO VISTA
CONTROLADOR (MVC) PARA LA GESTION DE FICHAS MÉDICAS DE
DOCENTES Y ESTUDIANTES DE LA UNIDAD EDUCATIVA GONZÁLEZ
SUÁREZ DE LA CIUDAD DE AMBATO**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del Título de Ingeniero en Tecnologías de la Información.

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de software

AUTOR: Jonathan Daniel Nuñez Guerrero

TUTOR: Ing. Mg. Dennis Vinicio Chicaiza Castillo.

Ambato - Ecuador

febrero – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA WEB BASADO EN LA ARQUITECTURA MODELO VISTA CONTROLADOR (MVC) PARA LA GESTION DE FICHAS MÉDICAS DE DOCENTES Y ESTUDIANTES DE LA UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ DE LA CIUDAD DE AMBATO desarrollado bajo la modalidad Proyecto de Investigación por el señor Jonathan Daniel Nuñez Guerrero, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2024.

Ing. Mg. Dennis Vinicio Chicaiza Castillo.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: SISTEMA WEB BASADO EN LA ARQUITECTURA MODELO VISTA CONTROLADOR (MVC) PARA LA GESTION DE FICHAS MÉDICAS DE DOCENTES Y ESTUDIANTES DE LA UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ DE LA CIUDAD DE AMBATO es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2024.



Jonathan Daniel Nuñez Guerrero

C.C. 1805120910

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2024.



Jonathan Daniel Nuñez Guerrero

C.C. 1805120910

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Jonathan Daniel Nuñez Guerrero, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA WEB BASADO EN LA ARQUITECTURA MODELO VISTA CONTROLADOR (MVC) PARA LA GESTION DE FICHAS MÉDICAS DE DOCENTES Y ESTUDIANTES DE LA UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ DE LA CIUDAD DE AMBATO , nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, febrero 2024.

Ing. Mg. Elsa Pilar Urrutia Urrutia.
PRESIDENTE DEL TRIBUNAL

Ing. Mg. Leonardo Torres Valverde.
PROFESOR CALIFICADOR

Ing. PhD. Félix Fernández Peña.
PROFESOR CALIFICADOR

DEDICATORIA

Quisiera comenzar expresando mi profunda gratitud a mis padres, Narcisa Guerrero y Jorge Nuñez, cuyo amor incondicional, sacrificio y constante apoyo han sido los pilares fundamentales en mi vida y en la realización de este trabajo. Su aliento y sabias palabras han sido mi inspiración y motivación en cada paso del camino.

A mi querido angelito, Rosa Nuñez, quien desde el cielo ha sido mi luz y mi guía. Su amor eterno y su recuerdo han sido un impulso para seguir adelante, incluso en los momentos más difíciles.

A mis hermanos, cuya presencia siempre ha sido reconfortante y cuyo apoyo inquebrantable ha sido un recordatorio constante de que nunca estoy solo en esta travesía.

Finalmente, quiero expresar mi más sincero agradecimiento a todas las personas que, día tras día, me han brindado su apoyo, ya sea con palabras de aliento, gestos amables o simplemente estando a mi lado en este viaje académico. Su contribución ha sido invaluable y ha enriquecido enormemente mi experiencia.

A todos y cada uno de ustedes, mi más profundo agradecimiento. Sin su amor, apoyo y aliento, este logro no habría sido posible.

Jonathan Daniel Nuñez Guerrero

AGREDECIMIENTO

Primero y, ante todo, doy gracias a Dios por su constante guía, fortaleza y por permitirme cumplir uno de mis sueños más anhelados. Su infinita bondad ha sido mi luz en este camino académico y en cada paso hacia la realización de este proyecto.

Agradezco sinceramente a mi tutor, el Ing. Dennis Chicaiza, por su invaluable orientación, dedicación y apoyo durante todo el proceso de investigación y redacción de esta tesis. Sus conocimientos, paciencia y motivación han sido fundamentales para alcanzar este logro.

Expreso mi gratitud a la Unidad Educativa González Suárez por brindarme la oportunidad de realizar mi proyecto de investigación en sus instalaciones. El ambiente académico y los recursos proporcionados fueron fundamentales para el desarrollo y éxito de este proyecto.

A mi mejor amiga, Valeria Logroño, agradezco su constante apoyo, ánimo y comprensión. Su presencia y palabras de aliento han sido un pilar fundamental en los momentos de duda y dificultad.

A cada una de estas personas y entidades, mi más sincero agradecimiento. Sin su apoyo y contribución, este logro no habría sido posible.

Jonathan Daniel Nuñez Guerrero

ÍNDICE DE CONTENIDO

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGREDECIMIENTO	vii
ÍNDICE DE CONTENIDO	viii
ÍNDICE DE TABLAS	x
ÍNDICE DE FIGURAS	xi
RESUMEN EJECUTIVO.....	xiv
ABSTRACT.....	xv
CAPITULO I.- MARCO TEÓRICO	1
1.1. Tema de Investigación	1
1.1.1. Planteamiento del Problema.....	1
1.2. Antecedentes Investigativos	2
1.3. Fundamentación Teórica.....	4
1.4. Objetivos.....	15
1.4.1. Objetivo General	15
1.4.2. Objetivos Específicos.....	15
CAPÍTULO II.- METODOLOGÍA	16
2.1. Materiales.....	16
2.2. Métodos	20
2.2.1. Modalidad de la Investigación	20
2.2.2. Población y Muestra.....	21
2.2.3. Recolección de Información	23
2.2.4. Procesamiento y Análisis de Datos	37
CAPÍTULO III.- RESULTADOS Y DISCUSIÓN	39
3.1. Análisis y discusión de los resultados.....	39
3.1.1. Procesos para la toma de información de las fichas médicas.....	39
3.1.2. Sistema web basado en la arquitectura MVC	42
3.1.3. Tecnologías web basadas en la arquitectura MVC	44
3.1.4. Motor de base de datos para Laravel.....	48

3.1.5. Metodología ágil para el desarrollo web de un gestor de fichas medicas	50
3.2. Desarrollo de la propuesta	51
3.2.1. Fase 1: Inicio.....	51
3.2.2. Fase 2: Elaboración.....	53
3.2.3. Fase 3: Construcción.....	71
3.2. 4. Fase 4: Transición.....	87
CAPITULO IV.- CONCLUSIONES Y RECOMENDACIONES	98
Referencias Bibliográficas.....	99
ANEXOS	101

ÍNDICE DE TABLAS

Tabla 2.1:	Guía de Entrevista para la persona encargada del departamento médico de la institución.	17
Tabla 2.2:	Población de Estudio.....	21
Tabla 2.3:	Muestra Significativa	22
Tabla 2.4:	Muestra de Estudio.....	23
Tabla 2.5:	Guía de Entrevista aplicada a la persona encargada del departamento médico de la institución	26
Tabla 3.1:	Comparativa entre las mejores tecnologías web basadas en la arquitectura MVC.....	47
Tabla 3.2:	Comparativa de los mejores motores de base de datos para Laravel... 49	49
Tabla 3.3:	Comparativa de la mejor metodología ágil.	50
Tabla 3.2.1:	Entorno de Trabajo.....	52
Tabla 3.2.2:	Detalle Caso de Uso – Inicio de Sesión	54
Tabla 3.2.3:	Detalle Caso de Uso – Gestión Roles.....	55
Tabla 3.2.4:	Detalle Caso de Uso – Gestión Usuarios.....	55
Tabla 3.2.5:	Detalle Caso de Uso – Gestión Tablas Administrativas.....	56
Tabla 3.2.6:	Detalle Caso de Uso – Gestión Docentes	57
Tabla 3.2.7:	Detalle Caso de Uso – Gestión Usuarios.....	58
Tabla 3.2.8:	Detalle Caso de Uso – Gestión Usuarios.....	58
Tabla 3.2.9:	Detalle Caso de Uso – Gestión Usuarios.....	58
Tabla 3.2.10:	Prueba Unitaria Inicio de Sesión.	87
Tabla 3.2.11:	Prueba Unitaria Gestión Roles.....	87
Tabla 3.2.12:	Prueba Unitaria Gestión Usuarios.....	87
Tabla 3.2.13:	Pruebas Unitarias Tablas Administrativas.....	88
Tabla 3.2.14:	Prueba Unitaria Gestión Docentes.....	88
Tabla 3.2.15:	Prueba Unitaria Gestión Estudiantes.	88
Tabla 3.2.16:	Prueba Unitaria Gestión Reporte.	89
Tabla 3.2.17:	Prueba Unitaria Cerrar Sesión.	89
Tabla 3.2.18:	Pruebas caja negra con las pruebas unitarias.	91
Tabla 3.2.19:	Cronograma capacitación del sistema.....	97

ÍNDICE DE FIGURAS

Figura 2.1:	Solicitud de información sobre las historias clínicas.	27
Figura 2.2:	Medio de recolección de información por parte del departamento médico.	28
Figura 2.3:	Conocimiento de la ubicación del Departamento Medico.	28
Figura 2.4:	Instalaciones del Departamento Médico.	29
Figura 2.5:	Calificación del Departamento Medico.....	30
Figura 2.6:	Control de Fichas Medicas.	30
Figura 2.7:	Preguntas referentes a las fichas médicas.....	31
Figura 2.8:	Tiempo de demora para la atención médica institucional.	32
Figura 2.9:	Fichas Médicas Necesarias en la Institución.....	32
Figura 2.10:	Solicitud de información sobre las historias clínicas	33
Figura 2.11:	Medio de recolección de información por parte del departamento médico.	34
Figura 2.12:	Efectividad en el manejo manual de las fichas médicas.....	34
Figura 2.13:	Gestión de fichas medicas mediante un sistema web.	35
Figura 2.14:	Mejoramiento del Departamento Médico.....	36
Figura 2.15:	Inconvenientes con las Fichas Médicas.....	37
Figura 3.1:	Proceso para llenado de fichas médicas de los docentes.....	39
Figura 3.2:	Proceso actualizado para llenado de fichas médicas de los docentes.	40
Figura 3.3:	Proceso para llenar las fichas médicas de los estudiantes.....	41
Figura 3.4:	Proceso actualizado para llenar las fichas médicas de los estudiantes.	42
Figura 3.5:	Arquitectura MVC.....	43
Figura 3.2.1:	Usuarios del Sistema.	53
Figura 3.2.2:	Diagrama de Casos de uso.....	54
Figura 3.2.3:	Diagrama de la Base de Datos para la autenticación de usuarios..	59
Figura 3.2.4:	Tablas de docentes y estudiantes.....	60
Figura 3.2.5:	Ejemplo Tablas Administrativas y Tablas Intermedias (pivot).....	61
Figura 3.2.6:	Diagrama de base de datos para registrar la información de docentes y estudiantes	62
Figura 3.2.7:	Arquitectura MVC con Laravel.....	63
Figura 3.2.8:	Inicio de Sesión.	63
Figura 3.2.9:	Registro de usuario.....	64
Figura 3.2.10:	Vista inicio/home.....	64

Figura 3.2.11:	Roles de usuario.....	65
Figura 3.2.12:	Registro de roles de usuario con sus respectivos permisos.	65
Figura 3.2.13:	Usuarios Registrados	66
Figura 3.2.14:	Registro de usuarios.....	66
Figura 3.2.15:	Docentes o Estudiantes Registrados.	67
Figura 3.2.16:	Registro de docentes o estudiantes.	67
Figura 3.2.17:	Formato ficha médica docentes o estudiantes.....	68
Figura 3.2.18:	Listado de tipos registrados en las tablas administrativas.	68
Figura 3.2.19:	Registrar el tipo en la tabla administrativa.	69
Figura 3.2.20:	Filtrado de datos para los reportes.	69
Figura 3.2.21:	Reporte del estudiante por tipo de sangre y grado o curso.	70
Figura 3.2.22:	Estructura de Laravel.	71
Figura 3.2.23:	Conexión a la base de datos MySQL de XAMMP.....	71
Figura 3.2.24:	Migraciones del Sistema.	72
Figura 3.2.25:	Creación de Migración de la tabla gradocurso con una relación a la tabla niveleducación.....	72
Figura 3.2.26:	Archivo de Configuración AdminLTE.....	73
Figura 3.2.27:	Plantilla general para la creación de archivos nombre_archivo.blade.php.....	73
Figura 3.2.28:	Controladores definidos por Laravel Auth.	74
Figura 3.2.29:	Vistas definidas por Laravel Auth.	74
Figura 3.2.30:	Asignación de proveedor de servicios Spatie.	75
Figura 3.2.31:	Asignación de Middleware Aliases para Spatie.	75
Figura 3.2.32:	Asignación Has Roles al Modelo User.php.....	76
Figura 3.2.33:	Método para asignar un rol al registrar un usuario.	76
Figura 3.2.34:	Ejemplo Seeder para la creación de permisos.	77
Figura 3.2.35:	Edición vista register.blade.php para asignar un rol al usuario. ...	77
Figura 3.2.36:	Controlador Rol y función store para la creación de Rol y sus permisos.	78
Figura 3.2.37:	Controlador Usuario y función store para la creación de usuarios con su rol.	78
Figura 3.2.38:	Middleware en controlador usuario apuntando a una función.....	79
Figura 3.2.39:	Migración de la tabla administrativa género.....	79
Figura 3.2.40:	Modelo tabla Administrativa Género.	80
Figura 3.2.41:	Controlador Género y constructor con permisos.....	80

Figura 3.2.42:	Función index y create del controlador Género.....	80
Figura 3.2.43:	Función store y show del controlador Género.....	81
Figura 3.2.44:	Función edit y update del controlador Género.....	81
Figura 3.2.45:	Función destroy del controlador Género.	82
Figura 3.2.46:	Migración de la tabla gradocurso con su relación.	82
Figura 3.2.47:	Modelo de estudiante con las funciones de las tablas administrativas que están relacionas al modelo.	83
Figura 3.2.48:	Funciones del controlador estudiante para el crud y para la generación de ficha médica.	83
Figura 3.2.49:	Función pdf para obtener los datos de estudiante y generar la ficha médica.	84
Figura 3.2.50:	Función tiposangrea y pdftiposangre para generar reportes por tipo de sangre de los estudiantes de la institución.....	84
Figura 3.2.51:	Función tiposangregrado y pdftiposangregrado para generar reporte del tipo de sangre de los estudiantes por cursos o grados.....	85
Figura 3.2.52:	Función genero y pdfgenero para generar reporte por genero de los estudiantes de la institución.	85
Figura 3.2.53:	Función generogrado y pdfgenerogrado para generar reporte por genero y grado del estudiante.....	86
Figura 3.2.54:	Función edadcurso y pdfedadcurso para genera reporte de estudiantes por edad y curso o grado al que pertenece.	86
Figura 3.2.55:	Contratar plan en Hostinger.es.....	91
Figura 3.2.56:	Contratar dominio en Hostinger.	92
Figura 3.2.57:	Panel principal de Hostinger.....	92
Figura 3.2.58:	Creación de la base de datos.	93
Figura 3.2.59:	Listado de base de datos en Hostinger.....	93
Figura 3.2.60:	Exportación de la base de datos local.	93
Figura 3.2.61:	Importar base de datos a Hostinger.....	94
Figura 3.2.62:	Base de datos importada a Hostinger correctamente.....	94
Figura 3.2.63:	Comando para puesta en producción del proyecto Laravel.....	94
Figura 3.2.64:	Creación archivo .zip del proyecto Laravel.	95
Figura 3.2.65:	Archivo .zip descomprimido en la carpeta public_html.	95
Figura 3.2.66:	Configuración archivo .env.....	96
Figura 3.2.67:	Configuración archivo .htaccess.	96
Figura 3.2.68:	Inicio del sistema con dominio unidadeducativagonzalezsuarez.es 96	

RESUMEN EJECUTIVO

En la actualidad la mayoría de instituciones educativas cuentan con un departamento médico en el cual se documenta información de los docentes y estudiantes en una ficha médica para en caso de emergencia tener un indicio de cómo actuar, ya que algunos docentes o estudiantes pueden tener alergias a medicamentos, patologías, enfermedades importantes que podrían ser críticas para ser tratados dentro de la institución.

El presente proyecto de investigación tiene como objetivo mejorar la gestión de la documentación de fichas médicas de docentes y estudiantes, por lo que se propone implementar un sistema web basado en la arquitectura Modelo, Vista, Controlador (MVC) para gestionar las fichas médicas de docentes y estudiantes de la Unidad Educativa González Suárez ubicada en la ciudad de Ambato.

El sistema web se implementó utilizando el Framework de Laravel, ya que nos ofrece una estructura MVC sólida y coherente, lo que garantiza una separación clara de la lógica de negocio y una mejor organización del código, además posee una amplia gama de herramientas y características que ayuda acelerar el desarrollo. El gestor de base de datos que se utilizó es MySQL, y que garantiza la eficiencia y confiabilidad en el almacenamiento de la información.

La metodología utilizada para el desarrollo del proyecto de investigación es la metodología Agile Unified Process (AUP), en la cual existe cuatro fases. La fase de inicio que se enfoca en establecer los objetivos y alcance del sistema, la fase de elaboración es donde se identifica y priorizan los casos de uso, la fase de construcción se encarga de realizar la implementación de la solución y las pruebas y la fase de transición se realiza la entrega final del sistema y se prepara para su despliegue y uso por parte de los usuarios finales.

Palabras clave: Sistema web, Metodología AUP, Laravel, MySQL, Fichas Médicas.

ABSTRACT

Currently most educational institutions have a medical department in which information of teachers and students is documented in a medical record in case of emergency to have an indication of how to act, since some teachers or students may have allergies to medications, pathologies, important diseases that could be critical to be treated within the institution.

The objective of this research project is to improve the management of the documentation of medical records of teachers and students, so it is proposed to implement a web system based on the Model, View, Controller (MVC) architecture to manage the medical records of teachers and students of the Gonzalez Suarez Educational Unit located in the city of Ambato.

The web system was implemented using the Laravel Framework, since it offers a solid and coherent MVC structure, which guarantees a clear separation of the business logic and a better organization of the code, it also has a wide range of tools and features that help speed up the development. The database manager used is MySQL, which guarantees efficiency and reliability in the storage of information.

The methodology used for the development of the research project is the Agile Unified Process (AUP) methodology, in which there are four phases. The initiation phase focuses on establishing the objectives and scope of the system, the elaboration phase is where the use cases are identified and prioritized, the construction phase is responsible for implementing the solution and testing, and the transition phase is the final delivery of the system and its preparation for deployment and use by end users.

Keywords: Web system, AUP methodology, Laravel, MySQL, Medical records.

CAPITULO I.- MARCO TEÓRICO

1.1. Tema de Investigación

SISTEMA WEB BASADO EN LA ARQUITECTURA MODELO VISTA CONTROLADOR (MVC) PARA LA GESTION DE FICHAS MÉDICAS DE DOCENTES Y ESTUDIANTES DE LA UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ DE LA CIUDAD DE AMBATO

1.1.1. Planteamiento del Problema

A nivel mundial, la utilización de sistemas informáticos ha revolucionado el manejo de la información en todos los ámbitos de la sociedad. Los procesos en el campo de la salud generan diariamente una gran cantidad de información, y las primeras experticias de informatización en este campo se centraron específicamente en los procesos administrativos, pero no mejoraron significativamente la gestión clínica y asistencial. Con posterioridad, el foco cambió hacia la informatización de los sistemas de salud en todos los dominios, tomando como eje la información clínica. Este proceso se denomina “informatización de la capa clínica”, y tiene como objetivo que los actos de cada profesional de la salud y las características de cada paciente estén almacenados en el sistema de información. Por lo que las Historias Clínicas Electrónicas (HCE) están dedicadas a mejorar la eficiencia, calidad y seguridad en el cuidado de la salud. La adopción mundial de las HCE ha demostrado beneficios que incluyen la disminución de errores en medicina, mejoras a nivel de costo/eficacia, aumento de la eficiencia y la posibilidad de brindar un papel activo a los pacientes en la toma de decisiones clínicas. Son el centro de cualquier sistema de información en salud [1].

En Suecia, se cuenta ya con la historia clínica universal, uniforme para todo el sistema, manejada por grandes computadoras, principalmente porque es deseable regular la recolección y procesamiento de datos médicos, salvaguardar la confidencialidad y la seguridad de los datos personales relativos a la salud [2].

En gran parte de Latinoamérica, la Historia Clínica Electrónica no se ha extendido al mismo ritmo que en el resto del mundo, según diferentes autores esto puede ser debido tanto a situaciones económicas como a resistencia por parte del personal de salud. En el año 2006, el Servicio de Salud Metropolitano Occidente de Santiago de Chile implementó un sistema de información en salud para abordar de manera integrada la gestión hospitalaria, centrado en la atención clínica de los pacientes, el sistema de información en salud consta de la ficha clínica electrónica que contiene el historial

clínico del paciente, registro de atenciones, resultados de exámenes de laboratorio e imagenología, prescripción de fármacos; disponible desde cualquier centro de la Red de Salud, lo que asegura la continuidad en la atención, otorgando mayor calidad y seguridad a los pacientes. En cambio, en el Hospital Italiano de Buenos Aires, la implantación de la idea de contar con una Historia Clínica Electrónica (HCE) nació de la búsqueda de la eficiencia y calidad en la atención médica, y a la vez de la necesidad de continuidad en el cuidado del paciente [2].

En el Ecuador ya se habla de implementar sistemas electrónicos en el servicio de salud para brindar al médico un apoyo para poder tomar decisiones clínicas, que permitan además mantener al médico al tanto de su paciente en el lugar que se encuentre. Desde enero del 2009 en el hospital del Instituto Ecuatoriano de Seguridad Social (IESS), se implementó en el servicio de consulta externa el AS400, que se trata de un sistema del equipo de IBM (empresa multinacional estadounidense), de multiusuario, con una interfaz que controla mediante menú y comandos [2].

En la ciudad de Ambato en la Unidad Educativa González Suárez no se lleva un registro de fichas medicas adecuado de docentes y estudiantes por lo cual el departamento medico no puede actuar de manera inmediata en caso de alguna emergencia.

1.2. Antecedentes Investigativos

Después de realizar la revisión y análisis de proyectos de investigación dentro de los repositorios de la Universidades del Ecuador se encontró los siguientes antecedentes: Según Manzano Rojas Edison Javier [3] se puede determinar que:

La arquitectura Modelo –Vista –Controlador (MVC):

- Ayudo en el desarrollo ya que permite tener el código fuente ordenado y que sea reutilizable, también facilito su administración y entendimiento.
- Permite la escalabilidad con la finalidad de implementar mejoras del sistema en el futuro.

Las tecnologías se basan en la modelo vista controlador (MVC) por que permite ser reutilizable.

El Sistema Web basado en el Modelo Vista Controlador:

- Permite garantizar la seguridad de los datos que se obtiene de los procesos que se realizan en el sistema.

- Utilizando software libre, se concluyó que permite una gran cantidad de posibles soluciones dependiendo de los lenguajes que programación.

El uso de la arquitectura MVC permitió separar al sistema web en las capas de presentación, lógica de negocio y acceso a la información, además permitió realizar pruebas unitarias de cada capa lo que facilito el mantenimiento en caso de errores.

Según Chacha Plasencia Karen Vanesa [4] se puede determinar que:

- Utilizar metodologías ágiles es de suma importancia para mantener organizado el trabajo en pequeñas partes y estructuradas
- Es factible utilizar arquitectura MVC combinando la metodología XP y sistemas de producción Kanban.
- La aplicación de la metodología XP en conjunto con MVC para desarrollo de sistemas web mejoró los tiempos en el desarrollo, debido a que XP planifica mediante historias de usuario todos los módulos a generar y MVC organiza el código ya teniendo en cuenta los modelos generados, por lo que resulta más sencillo desarrollar, probar el funcionamiento del aplicativo y solucionar errores.

Según Jiménez Ruiz Edwin Rubén [5] se puede determinar que:

La arquitectura Modelo-Vista-Controlador (MVC)

- Aplicada en el desarrollo del sistema web, permitió escribir código mucho más ordenado de tal manera que las clases y sus componentes tengan una alta cohesión y un bajo acoplamiento. Al utilizar un Framework de desarrollo orientado a una arquitectura como MVC, el equipo de desarrollo puede enfocarse en las tareas propias de la lógica del negocio más no en tareas triviales tales como conexión a la base de datos.
- Facilitó el desarrollo ya que mantiene el código fuente ordenado y permite su reutilización, facilitando la administración y entendimiento.

Según Chanatasig Maigua Edwin Esteban [6] se puede determinar que:

- El uso de la arquitectura MVC benefició a la estructuración del sistema web separando las responsabilidades (accesibilidad a los datos, interfaz de usuario y la lógica de control) haciendo que al aplicativo sea más eficiente y propenso a escalabilidad en un futuro.
- Se evidenció la importancia de la implementación de la arquitectura MVC en el desarrollo del proyecto debido a la mantenibilidad además de la organización

de código haciendo más sencillo identificar posibles errores y posteriormente corregirlos.

- Las actividades que antes se las registraba en papel se la recopiló como información digital evitando riesgos de pérdida de datos en ambientes físicos, además, el control de registro de las citas y visitas a los departamentos de atención se volvió ágil y segura derivando en la satisfacción de los pacientes.

Según Segura Pico Gabriela Prisila [2] se puede determinar que:

- Las Historias Clínicas Electrónicas (HCE) están diseñadas para vencer muchas limitaciones, como así también proveer beneficios adicionales que no pueden ser obtenidos a través de una vista estática de datos.
- La Historia Clínica Electrónica como apoyo en el seguimiento a todo paciente con una enfermedad agudo o crónica, tiene un valor fundamental, pues permite conocer periódicamente el estado de control de la enfermedad, los factores que influyen en su descompensación, así como indicar las medidas y acciones que deben aplicarse para su corrección oportuna. En el proceso interviene el equipo multidisciplinario encargado de la atención del paciente diabético, encabezado por el médico y la enfermera de la familia.

1.3. Fundamentación Teórica

Sistema Web Basado en la Arquitectura MVC

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos[7].

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

- **El Modelo** que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- **La Vista**, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
- **El Controlador**, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.)[7].

El controlador es responsable de:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo " SI Evento Z, entonces Acción W ". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar ()", una de las peticiones al modelo puede ser "Obtener_tiempo_de_entrega (nueva_orden_de_venta) "[7].

Las vistas son responsables de:

- Recibir datos del modelo y los muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes)[7].

Ingeniería en Software

Haciendo una recopilación de todos los conceptos que se han dado sobre la Ingeniería de software, Se define como la disciplina o área de la informática, que hace uso razonable de los principios de ingeniería con el objetivo de obtener soluciones informáticas económicamente factible y que se adapte a las necesidades de las empresas reales, tomando en cuenta los procesos de producción y mantenimiento de software que son desarrollados y modificados en el tiempo y con los costos estimados. Esta ingeniería trata con áreas muy diversas de la informática y de las Ciencias de la

Computación, tales como construcción de compiladores, Sistemas Operativos, o desarrollos Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de Sistema de Información y aplicables a infinidad de áreas (negocios, investigación científica, medicina, producción, logística, banca, etc.)[8].

Algunas definiciones, dadas a través del tiempo son:

- “Ingeniería de Software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas de software”.
- “Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como Desarrollo de Software o Producción de Software”.
- “Ingeniería de Software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales”.
- “Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software”.

En conclusión, se puede decir que los cuatro autores anteriores, de manera diferente describen en sí el principal objetivo de la ingeniería de software, la cual es el establecimiento y puesta en práctica de los principios y metodologías que nos lleven a un desarrollo eficiente de software en todas las etapas desde sus inicios hasta su implementación y mantenimiento[8].

Desarrollo de Software

El desarrollo de software se refiere a un conjunto de actividades informáticas dedicadas al proceso de creación, diseño, despliegue y compatibilidad de software. El software en sí es el conjunto de instrucciones o programas que le dicen a una computadora qué hacer. Es independiente del hardware y hace que las computadoras sean programables. Hay tres tipos básicos:

- Software del sistema para proporcionar funciones básicas como sistemas operativos, administración de discos, servicios, administración de hardware y otras necesidades operacionales.

- Software de programación para brindar a los programadores herramientas como editores de texto, compiladores, enlazadores, depuradores y otras herramientas para crear código.
- Software de aplicación (aplicaciones o apps) para ayudar a los usuarios a realizar tareas. Las suites de productividad de Office, el software de gestión de datos, los reproductores multimedia y los programas de seguridad son algunos ejemplos. Aplicaciones también se refiere a aplicaciones web y móviles como las que se utilizan para comprar en Amazon.com, socializar en Facebook o publicar imágenes en Instagram.
- Un posible cuarto tipo es el software integrado. El software de sistemas integrado se utiliza para controlar máquinas y dispositivos que normalmente no se consideran computadoras, como redes de telecomunicaciones, automóviles, robots industriales y más. Estos dispositivos, y su software, se pueden conectar como parte del Internet de las Cosas (IoT)[9].

El desarrollo de software lo llevan a cabo principalmente programadores, ingenieros de software y desarrolladores de software. Estos roles interactúan y se superponen, y la dinámica entre ellos varía mucho entre los departamentos y comunidades de desarrollo.

Los programadores, o codificadores, escriben el código fuente para programar computadoras para realizar tareas específicas como fusionar bases de datos, procesar pedidos en línea, enrutar comunicaciones, realizar búsquedas o mostrar texto y gráficos. Los programadores suelen interpretar las instrucciones de los desarrolladores e ingenieros de software y utilizan lenguajes de programación como C++ o Java para llevarlas a cabo.

Tecnologías web basadas en MVC.

Las tecnologías web basadas en el patrón de arquitectura Modelo-Vista-Controlador (MVC) son aquellas que siguen este enfoque para estructurar y organizar el desarrollo de aplicaciones web. Algunas de las principales tecnologías web que se basan en el patrón MVC incluyen:

- Laravel: Es un framework de desarrollo web en PHP que utiliza el patrón MVC como su arquitectura base. Proporciona una estructura organizada para el

desarrollo de aplicaciones web y ofrece herramientas y características para facilitar la implementación del patrón MVC.

- Ruby on Rails: Es un framework de desarrollo web en Ruby que también sigue el patrón MVC. Proporciona una estructura modular y convenciones predefinidas para agilizar el desarrollo de aplicaciones web.
- ASP.NET MVC: Es un framework de desarrollo web en el entorno de desarrollo de Microsoft .NET. Permite la creación de aplicaciones web basadas en el patrón MVC, ofreciendo una separación clara entre la lógica de negocio, la presentación y el control.

Motor de base de datos

Es un software diseñado para administrar y organizar grandes cantidades de información de manera eficiente y segura. Es responsable de gestionar el almacenamiento, la recuperación y la manipulación de los datos en una base de datos. Un motor de base de datos proporciona una interfaz para que las aplicaciones puedan interactuar con la base de datos y realizar operaciones como inserción, actualización, eliminación y consulta de datos. Además, se encarga de mantener la integridad de los datos, garantizando que se cumplan las restricciones y reglas definidas en la estructura de la base de datos.

Los motores de base de datos pueden ser de diferentes tipos como, por ejemplo:

- Relacionales: Estos motores gestionan bases de datos relacionales, donde los datos se organizan en tablas y se establecen relaciones entre ellas mediante claves primarias y foráneas. Algunos ejemplos de motores de base de datos relacionales son MySQL, PostgreSQL y Oracle.
- NoSQL: Estos motores gestionan bases de datos no relacionales, que permiten un modelado flexible de los datos y no requieren una estructura fija en forma de tablas. Algunos ejemplos de motores de base de datos NoSQL son MongoDB, Cassandra y Redis.
- Columnares: Estos motores están optimizados para gestionar grandes cantidades de datos organizados en columnas en lugar de filas, lo que los hace especialmente eficientes para consultas analíticas. Un ejemplo de motor de base de datos columnar es Apache HBase.

Metodologías de Desarrollo de Software

Metodologías Clásicas

Al inicio el desarrollo de software era de manera artesanal en su totalidad, la fuerte necesidad de mejorar constantemente el proceso y llevar los proyectos al cumplimiento de los objetivos y meta deseada por las empresas, por lo que se tuvo que incorporar la concepción y fundamentos de diferentes metodologías existentes de otras áreas del conocimiento que fueran adaptadas al desarrollo de software.

Esta nueva etapa de adaptación contenía el progreso dividido en etapas de manera secuencial que de algo mejoraba la necesidad latente en el campo de la ingeniería de software.

Microsoft Solution Framework (MSF)

MSF es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información.

Todo proyecto es separado en cinco principales fases:

- Visión y Alcances.
- Planificación.
- Desarrollo.
- Estabilización.
- Implantación.

Rational Unified Process (RUP)

RUP es un proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por Rational Software, y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. (Customización). Es guiado por casos de uso y centrado en la arquitectura, y utiliza UML como lenguaje de notación.

Fases Las cuatro fases del ciclo de vida son:

- Concepción

- Elaboración
- Construcción
- Transición

Metodologías Ágiles

El método ágil, nace como respuesta a los diferentes problemas presentados y detallados en la metodología tradicional, está se basa en dos aspectos puntuales, el primero, el retrasar las decisiones y segunda, la planificación adaptativa; permitiendo potenciar aún más el desarrollo de software a gran escala. Como resultado de esta nueva teoría se crea un Manifiesto Ágil cuyas principales ideas son:

- Los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados.
- Es más importante crear un producto software que funcione que escribir documentación exhaustiva.
- La colaboración con el cliente debe prevalecer sobre la negociación de contratos.
- La capacidad de respuesta ante un cambio es más importante que el seguimiento estricto de un plan.
- Entre los principales métodos ágiles se tiene: el XP (eXtreme Programming), Scrum, Iconix, Cristal Methods, AUP entre otras[9].

AUP (Agile Unified Process)

El Proceso Unificado Agil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (test driven development - TDD), Modelado Agil, Gestión de Cambios Agil, y Refactorización de Base de Datos para mejorar la productividad.

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados:

- Inception(Concepción): El objetivo de esta fase es obtener una comprensión común clienteequipo de desarrollo del alcance del nuevo sistema y definir una o varias arquitecturas candidatas para el mismo.
- Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- Construcción: Durante la fase de construcción el sistema es desarrollado y probado al completo en el ambiente de desarrollo.
- Transición: el sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.[10]

OOHDM (Object-Oriented Hypermedia Design Method)

Es una metodología de diseño para el desarrollo de aplicaciones hipermedia basadas en objetos. Fue desarrollada por Mark Bernstein y Mark Atkins en la década de 1990 como una extensión de la metodología OMT (Object Modeling Technique).

La metodología OOHDM se centra en el diseño conceptual y navegacional de las aplicaciones hipermedia, poniendo énfasis en la representación y organización de la información, así como en la interacción del usuario con dicha información. A continuación, se describen las principales fases de la metodología OOHDM:

- Análisis de requerimientos: En esta fase se recopilan y analizan los requerimientos del sistema, identificando los objetivos y necesidades de los usuarios. Se definen los casos de uso, se modelan los conceptos clave y se establecen los requisitos funcionales y no funcionales.
- Diseño conceptual: En esta fase se crea un modelo conceptual del sistema, utilizando diagramas de clases y otros diagramas de modelado. Se definen las clases, atributos y relaciones entre los objetos. Se identifican los principales objetos y sus interacciones en el sistema hipermedia.
- Diseño navegacional: En esta fase se diseña la estructura de navegación del sistema, definiendo cómo los usuarios pueden moverse entre los diferentes contenidos y funcionalidades. Se crean diagramas de navegación y se establecen las reglas de navegación.
- Diseño de presentación: En esta fase se diseña la interfaz de usuario del sistema hipermedia. Se define la apariencia visual, la disposición de los elementos y la

interacción con el usuario. Se crean prototipos y se realizan pruebas de usabilidad.

- **Implementación:** En esta fase se realiza la implementación del sistema hipermedia, utilizando lenguajes de programación y herramientas adecuadas. Se desarrollan los componentes y funcionalidades definidas en las fases anteriores. Se realizan pruebas unitarias y de integración para asegurar la calidad del software.
- **Evaluación y mantenimiento:** En esta fase se evalúa el sistema hipermedia en términos de usabilidad, rendimiento y cumplimiento de los requerimientos. Se realizan ajustes y correcciones según los resultados de la evaluación. También se realiza el mantenimiento del sistema para asegurar su correcto funcionamiento a lo largo del tiempo.

WSDM (Web Service Development Methodology)

Es un método de diseño para sitios web, donde el objetivo es detectar y definir los perfiles de usuario. WSDM tiene como finalidad realizar un acercamiento con el usuario para la obtención de requisitos y además modela la aplicación en base a los requerimientos de cada grupo o clases de usuarios.

WSDM se caracteriza por hacer cumplir los procesos del ciclo de vida del software y cabe recalcar que esta metodología realiza un análisis al entorno empresarial en el que se va a instalar el sistema y los procesos que se generará; también permite crear modelos que describan a las aplicaciones web desde diferentes puntos de vista con diferentes niveles de abstracción, siendo considerada una metodología completa para construir y generar software web.

La metodología WSDM, se divide en seis fases del sistema que se describen a

- **continuación.**
Análisis de requerimientos: En esta fase se identifican y comprenden los requerimientos del servicio web. Se definen los objetivos, funcionalidades y restricciones del servicio, así como los casos de uso y los requisitos no funcionales.
- **Diseño de la arquitectura:** En esta fase se define la arquitectura del servicio web. Se establecen los componentes y módulos necesarios, así como la

interacción entre ellos. También se determina la tecnología y los estándares que se utilizarán en el desarrollo del servicio.

- **Diseño del contrato:** En esta fase se especifica el contrato del servicio web, que define los métodos, parámetros y formatos de mensajes que se utilizarán en las interacciones con el servicio. Se utilizan lenguajes como WSDL (Web Services Description Language) para describir el contrato de manera precisa.
- **Implementación:** En esta fase se realiza la implementación del servicio web. Se desarrollan los componentes del servicio, se escriben los códigos necesarios y se integran con otros sistemas si es necesario. Se utilizan tecnologías y frameworks específicos para la implementación de servicios web.
- **Pruebas y depuración:** En esta fase se llevan a cabo pruebas exhaustivas del servicio web para asegurar su correcto funcionamiento. Se realizan pruebas unitarias, pruebas de integración y pruebas de rendimiento para validar el servicio en diferentes escenarios. También se identifican y corrigen errores o problemas encontrados durante las pruebas.
- **Despliegue y mantenimiento:** En esta fase se realiza el despliegue del servicio web en el entorno de producción. Se configuran los servidores y se realizan las configuraciones necesarias para poner el servicio en funcionamiento. Además, se establece un plan de mantenimiento para realizar actualizaciones, correcciones de errores y mejoras continuas del servicio.[11]

Ética Médica y Bioética

Ética

La ética enseña que la razón para actuar los que brindan los servicios de salud es el “paciente”, o cliente externo para otros para lograr su satisfacción y cumplir sus expectativas y su seguridad. Por tanto, los servicios deben ser de alta calidad, lo más solidario y humano posible, así como respetar el derecho de información, la integridad de las personas que los reciben, siempre que no interfiera con la moral del profesional o de otras personas [12].

Bioética

El concepto de bioética surge después de la Segunda Guerra Mundial, y fue definida como: disciplina científica que estudia los aspectos éticos de la biología en general, la medicina, así como de las relaciones entre los hombres y con los restantes seres vivos.

La bioética aspira a proveer los principios orientadores de la conducta humana en el campo biomédico. Etimológicamente proviene del griego bios y ethos: “ética de la vida”, es decir de la ética aplicada a la vida humana y no humana (animales, naturaleza) [12]

Fichas Médicas o Historias Clínicas

Historia Clínica Única

Es un documento médico legal que consigna la exposición detallada y ordenada de todos los datos relativos a un paciente o usuario, incluye la información del individuo y sus familiares, de los antecedentes, estado actual y evolución, además de los procedimientos y de los tratamientos recibidos.

Historia Clínica Activa

Se considera a la historia clínica que tiene un periodo de vigencia hasta cinco años desde la última atención registrada.

Historia Clínica Inactiva

Se considera a la historia clínica que no tiene ningún registro de atención por más de cinco años y por tanto debe ser trasladada al archivo pasivo.

Archivo Clínico

Es el lugar donde se guardan las Historias Clínicas, de manera ordenada y accesible. Debe ser centralizado y está dividido en archivo activo y archivo pasivo, tomando en cuenta la última atención al paciente [13].

Conservación

Es un archivo permanente donde se mantienen las historias clínicas en forma organizada y segura a fin de que estén disponibles y accesibles, clasificadas por su periodo de utilización de acuerdo a la complejidad de la unidad operativa. El ordenamiento de las Historias Clínicas se realizará de acuerdo a las normas vigentes [13].

Etapas de Conservación

Existen dos etapas de conservación de la Historia Clínica: periodo de uso activo, pasivo.

Archivo Activo

Es un archivo permanente donde se conservan las historias clínicas en forma ordenada y accesible que se mantiene en un lapso de tiempo de conservación que permita el uso y consulta frecuente.

Archivo Pasivo

Es un archivo permanente ordenado donde se conservan las historias clínicas de pacientes que no han acudido al establecimiento de salud durante un periodo determinado [13].

1.4. Objetivos

1.4.1. Objetivo General

Implantar un sistema web basado en la arquitectura Modelo, Vista y Controlador (MVC) para la gestión de fichas médicas de los docentes y estudiantes de la Unidad Educativa González Suárez de la ciudad de Ambato.

1.4.2. Objetivos Específicos

- Caracterizar los procesos del manejo de las fichas médicas estudiantiles.
- Identificar las tecnologías más adecuadas para el desarrollo web utilizando el modelo arquitectónico MVC.
- Desarrollar el sistema web de gestión de fichas médicas estudiantiles para la Unidad Educativa González Suárez de Ambato.

CAPÍTULO II.- METODOLOGÍA

2.1. Materiales

Para la recolectar la información del presente proyecto de investigación se utilizó una entrevista dirigida a la encargada del departamento médico de la Unidad Educativa González Suárez y dos encuestas con cuestionario, la primera está dirigida a los estudiantes y la segunda dirigida a los docentes de la institución.

Guía de entrevista aplicada a la encargada del departamento médico de Unidad Educativa González Suárez.

Datos Personales			
Apellidos y Nombres encargada del departamento medico:			
Institución:			
Cargo:			
Objetivo:			
N°	Pregunta	Respuesta	Observación
1	¿Cuáles son los pasos para la creación de fichas médicas de los docentes y estudiantes?		
2	¿En su opinión ha tenido algún problema al registrar las fichas medicas de algún docente o estudiante?		
3	Utiliza algún medio de comunicación para obtener la información de fichas medicas de los docentes y estudiantes.		
4	¿Considera que la manera en la que se lleva a cabo los registros de las fichas médicas de docentes y estudiantes es óptima?		
5	¿Qué tan eficiente considera usted que es tener la información de las fichas medicas de los docentes y estudiantes?		

6	¿Al momento de atención al docente o estudiante, existen percances con las fichas medicas?		
7	¿Cree usted que se debe seguir llevando el control de las fichas medicas de los docentes y estudiantes de forma física?		
8	¿Cree usted que es necesario contar con todo el historial médico de los docentes, por ejemplo, pedidos, imágenes, laboratorio y certificados médicos?		
9	¿En el caso de los estudiantes cree usted que es necesario contar con la autorización de los padres para recibir atención médica en la institución en caso de una emergencia?		
10	¿Cuál es el motivo más frecuente por lo que los estudiantes o docentes asisten al departamento medico?		
11	¿Cuál es la cantidad aproximada de estudiantes que acuden al departamento médico de la institución?		
12	¿De qué cursos nomas cree usted que hay más accidentes dentro de la institución?		
Conclusión:			

Tabla 2.1: Guía de Entrevista para la persona encargada del departamento médico de la institución.

Elaborado por: Investigador.

Encuesta dirigida a los Estudiantes de la Unidad Educativa González Suárez.

Pregunta 1

¿Se le ha solicitado a usted información sobre su historia clínica por parte del departamento médico de la institución?

- Si

- No

Pregunta 2

En caso de que la respuesta sea SI en la pregunta anterior seleccione cuál de estas opciones se ha utilizado por parte del departamento médico y en caso de que la respuesta sea NO seleccione Ninguna.

- Esquela
- WhatsApp
- Facebook
- Página Web
- Ninguna

Pregunta 3

¿En caso de algún accidente usted sabe a dónde acudir?

- Si
- No

Pregunta 4

¿Considera usted que las instalaciones del departamento médico de la institución cuentan con lo necesario en caso de una emergencia?

- Totalmente de acuerdo
- Parcialmente de acuerdo
- Indeciso
- Parcialmente en desacuerdo
- Totalmente en desacuerdo

Pregunta 5

¿Cómo calificaría la atención medica de la institución?

- Muy Satisfecho
- Satisfecho
- Neutral
- Insatisfecho
- Muy Insatisfecho

Pregunta 6

¿Está usted de acuerdo en que, el departamento médico de la institución lleve el registro de fichas medicas de docentes y estudiantes para poder actuar de manera eficiente en caso de un accidente?

- Totalmente de acuerdo
- De acuerdo
- Ni de acuerdo, ni en desacuerdo
- En desacuerdo
- Totalmente en desacuerdo

Pregunta 7

¿El médico de la institución realiza preguntas referentes a su ficha médica?

- Siempre
- Casi Siempre
- Casi Nunca
- Nunca

Pregunta 8

¿Conoce usted cuánto tiempo usualmente se demora el médico de la institución para atender a un docente o estudiante?

- 0-10 minutos
- 10:01-20 minutos
- 20:01-30 minutos
- Más de una hora

Encuesta dirigida a los Docentes de la Unidad Educativa González Suárez.

Pregunta 1

¿Cree usted que las fichas médicas tanto de docentes como estudiantes son necesarias en la institución?

- SI
- NO

Pregunta 2

¿Se le ha solicitado a usted información sobre su historia clínica por parte del departamento médico de la institución?

- SI
- NO

Pregunta 3

¿Qué medio de comunicación ha utilizado el departamento para pedir información sobre su historia clínica?

- WhatsApp

- Facebook
- Esquela
- De Forma Verbal
- Ninguna

Pregunta 4

¿Cree usted que es eficiente en la actualidad que la institución lleve de manera manual las fichas medicas?

- SI
- NO

Pregunta 5

¿Cree usted que un sistema web para gestionar las fichas medicas ayudaría a encontrar la información de manera más rápida y sencilla?

- SI
- NO

Pregunta 6

¿Cree usted que gestionar las fichas médicas de los estudiantes y docentes de manera digital ayude a mejorar al departamento médico?

- SI
- NO

Pregunta 7

¿Ha tenido algún inconveniente con su ficha médica por parte del departamento médico?

Seleccione cual es el inconveniente que se ha presentado por parte del departamento médico.

- Perdida de Ficha Médica
- Duplicidad
- Información Incompleta
- Ninguna

2.2. Métodos

2.2.1. Modalidad de la Investigación

La presente investigación se contextualizó en la modalidad de investigación de campo, documental-bibliográfica y modalidades especiales.

Investigación de Campo

La investigación que se utilizó es la de campo por que se recolectó información del lugar de los hechos, ya sea por medio de encuestas y entrevistas directamente con las personas involucradas. Además. Se recogió datos por parte del departamento médico que aportan a la solución del problema planteado.

Investigación Bibliográfica – Documental

La investigación fue bibliográfica – documental en la obtención de información en tesis, proyectos, revistas científicas, documentos y textos que puedan servir de sustento para el proyecto, indagando acerca de problemas similares y sus respectivas propuestas de solución que se hayan establecido con anterioridad.

2.2.2. Población y Muestra

La presente investigación trabajará con la población referente a los docentes, estudiantes y departamento médico, que son los encargados de brindar la información necesaria para llenar las fichas médicas institucionales.

Debido a que se analizará los criterios adecuados para un buen manejo de fichas médicas se realizará una encuesta tanto a los docentes, estudiantes y una entrevista a la persona encargada del departamento médico de la institución.

Población	Número	Porcentaje
Departamento Medico	1	0,08 %
Docentes	65	4,98%
Estudiantes	1239	94,94%
Total	1305	100%

Tabla 2.2: Población de Estudio

Elaborado por: Investigador

Considerando que la población de representantes calculada es mayor de 100, como resultado se planea realizar el cálculo del tamaño de muestra.

Se utilizará la fórmula con muestra finita con una población de 1239 estudiantes y el tipo de muestreo es probabilístico debido a que la selección será al azar. Por tanto, la fórmula busca encontrar la muestra de cierta población.

Cálculo mediante la presente fórmula:

$$n = \frac{NZ^2PQ}{Ne^2 + Z^2PQ}$$

Para la muestra significativa de estudiantes de la Institución:

Población Estudiantes y representantes	Número	Porcentaje
Estudiantes y representantes de Básica Elemental	239	19.2%
Estudiantes y representantes de Básica Media	500	40.4%
Estudiantes y representantes de Básica Superior	500	40.4%
Total	1239	100%

Tabla 2.3: Muestra Significativa
Elaborado por: Investigador

En donde:

N= Tamaño de la población de estudiantes y representantes equivalente a 1239

e= Error estándar 9% equivalente a 0,09

Z= Nivel de confianza 95% equivalente a 1,96

P= Probabilidad de éxito equivalente a 0,5

Q= Probabilidad de fracaso equivalente a 0,5

n= Tamaño de la muestra

$$(1239)(1,96)^2(0,5)(0,5)$$

$$n = \frac{\quad}{\quad}$$

$$(1239)(0,09)^2 + (1,96)^2(0,5)(0,5)$$

$$= 108$$

El tamaño de la muestra corresponde a 1239 estudiantes.

En el caso del departamento medido se trabajará con el total equivalente a 1 persona, al no ser mayor de 100 la población se trabajará con el total del departamento médico. Mientras de los docentes se trabajará con el total equivalente a 65 personas, al no ser mayor de 100 la población se trabajará con el total de docentes. Por lo tanto, el total del tamaño de la muestra corresponde a 174 personas, los cuáles serán encuestados.

Población	Número	Porcentaje
Departamento Medico	1	0.58%
Docentes	65	37.36%
Estudiantes	108	62.06%
Total	174	100%

Tabla 2.4: Muestra de Estudio
Elaborado por: Investigador

2.2.3. Recolección de Información

Los siguientes resultados que se muestran forman parte de la entrevista aplicada a la persona encargada del departamento médico de la Unidad Educativa González Suárez.

Datos Personales			
Apellidos y Nombres encargada del departamento medico: Licda. Sady Contrado.			
Institución: Unidad Educativa González Suárez.			
Cargo: Departamento Médico.			
Objetivo: Recopilar información sobre el proceso y manejo de fichas médicas de docentes y estudiantes.			
N°	Pregunta	Respuesta	Observación
1	¿Cuáles son los pasos para la creación de fichas médicas de los docentes y estudiantes?	Para la realización de las fichas medicas por parte de los docentes solicitamos el laboratorio. De hecho, aquí en la institución viene el laboratorio que contratamos, le toman las muestras de los docentes y cuando ya tengamos el resultado	Al momento de enviar esquelas los estudiantes tienden a perderlas o no entregar a su representante.

		<p>de ese se le hacen las fichas médicas.</p> <p>Por parte de los estudiantes le pedimos información a través de esquelas a los padres de familia, con lo cual nos llena las esquelas y de esa manera recogemos esos datos.</p>	
2	<p>¿En su opinión ha tenido algún problema al registrar las fichas médicas de algún docente o estudiante?</p>	<p>De los docentes no tenemos problemas porque como son mayores de edad, ellos saben que con eso cuentan con seguro también, pero en cambio los alumnos a través de los padres de familia se nos dificulta obtener la información porque no nos entregan, o entregan con datos insuficientes.</p>	<p>Se presenta inconvenientes en la toma de información para las fichas médicas de los estudiantes.</p>
3	<p>Utiliza algún medio de comunicación para obtener la información de fichas medicas de los docentes y estudiantes.</p>	<p>A los docentes de forma verbal una entrevista directa, pero en cambio a los estudiantes mediante esquela a través de los padres de familia o representantes.</p>	
4	<p>¿Considera que la manera en la que se lleva a cabo los registros de las fichas médicas de docentes y estudiantes es óptima o se podría mejorar?</p>	<p>Yo creo que los docentes no tenemos dificultades, pero en cambio para los estudiantes, si podemos mejorar, porque como les decía, no nos ayudan lo suficientemente los padres de familia.</p>	
5	<p>¿Qué tan eficiente considera usted que es tener la información de las fichas</p>	<p>Sí es bien importante porque podemos hacer un trabajo preventivo y a la vez tener</p>	

	medicas de los docentes y estudiantes?	información para no cometer errores.	
6	¿Al momento de atención al docente o estudiante, existen percances con las fichas medicas?	Sí, sí existen infinidad de percances, porque cada vez que se entra a una ficha, por ejemplo, no obtengo la información suficiente de un estudiante como a que es alérgico, si padece de alguna enfermedad que eso es bien importante en caso de alguna emergencia.	Información Insuficiente en las fichas médicas de los estudiantes.
7	¿Cree usted que se debe seguir llevando el control de las fichas medicas de los docentes y estudiantes de forma física?	Creo que de forma física no tiene tanta validez porque habría que traspapelar demasiado, en cambio, con un sistema nos permitiría ir directamente al estudiante o al docente que necesita atención médica.	Toma mucho tiempo buscar la ficha del docente o estudiante.
8	¿Cree usted que es necesario contar con todo el historial médico de los docentes, por ejemplo, pedidos, imágenes, laboratorio y certificados médicos?	Sí, claro esa forma es informativa y tenemos un registro que nos permite, sí es docente saber. A partir de qué tiempo padece de una enfermedad y si tiene una enfermedad actual.	
9	¿En el caso de los estudiantes cree usted que es necesario contar con la autorización de los padres para recibir atención médica en la institución en caso de una emergencia?	Bueno, la emergencia siempre lo vamos a atender como sea, pero ahí entra detalles de lo que podemos. ¿Cómo podemos actuar ante la emergencia? Si es un niño que presenta una epilepsia, si toma medicamentos por una enfermedad, si es alérgico a algo y eso lo	

		tenemos a través de la información de los padres de familia.	
10	¿Cuál es el motivo más frecuente por lo que los estudiantes o docentes asisten al departamento médico?	Un motivo más grande es por la ingesta de alimentos de días anteriores, que dan lugar a los síntomas digestivos y dolores abdominales, esa es la principal causa por la que los estudiantes vienen al departamento médico.	
11	¿Cuál es la cantidad aproximada de estudiantes que acuden al departamento médico de la institución?	Aquí vienen alrededor de 20 a 25 estudiantes, a veces hasta 30, y créeme que los 20 niños tienen dolor abdominal.	
12	¿De qué cursos nomas cree usted que hay más accidentes dentro de la institución y cuáles son los motivos?	De la sección Básica sobre todo tercero a séptimo grado son los estudiantes que más vienen acá. Por el motivo de caídas, golpes y es que ellos por un rasguño, a veces vienen después 2 días anteriores ya que en el fin de semana en su casa se lastimaron o se dañaron y vienen aquí para que les cure.	
Conclusión: Según la entrevista realizada se puede determinar que el sistema web para la gestión de fichas médicas de docentes y estudiantes sería un aporte importante ya que optimizaría el proceso de búsqueda de la información, la recolección de datos, mantener la información del docente o estudiante disponible en todo momento.			

Tabla 2.5: Guía de Entrevista aplicada a la persona encargada del departamento médico de la institución

Elaborado por: Investigador

Resultados de las encuestas aplicada a los estudiantes de la Unidad Educativa González Suárez.

Pregunta 1

¿Se le ha solicitado a usted información sobre su historia clínica por parte del departamento médico de la institución?

108 respuestas

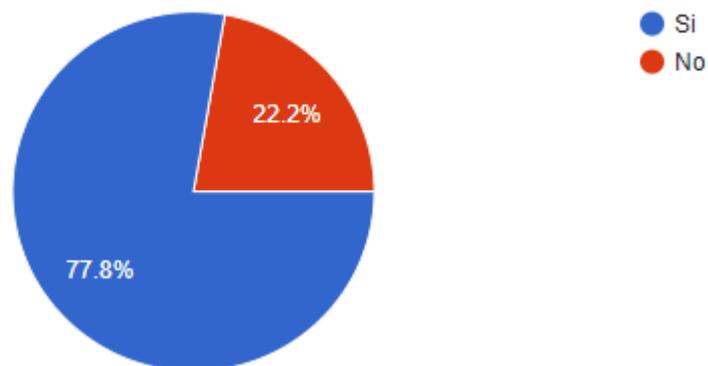


Figura 2.1: Solicitud de información sobre las historias clínicas.

Elaborado por: El investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.1 El 77.8% de los estudiantes que corresponderían a más de la mitad de los encuestados que se le ha solicitado información de las historias clínicas y al 22.2% no se le ha solicitado o no se ha pedido ninguna información. Por lo que se puede identificar que a pesar de que la mayoría de los estudiantes se ha solicitado información de las historias clínicas existe una cierta cantidad de estudiantes a los cuales no se les ha solicitado por lo que se haría difícil al departamento médico llenar sus fichas.

Pregunta 2

En caso de que la respuesta sea SI en la pregunta anterior seleccione cuál de estas opciones se ha utilizado por parte del departamento médico y en caso de que la respuesta sea NO seleccione Ninguna.

108 respuestas

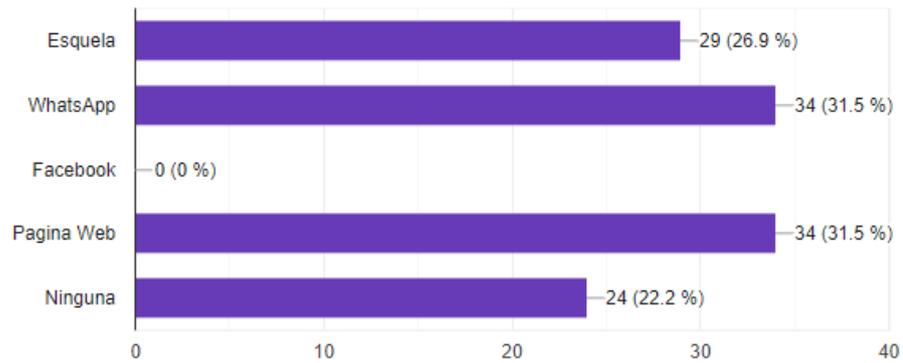


Figura 2.2: Medio de recolección de información por parte del departamento médico.

Elaborado por: Investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.2 el 31.5% de los encuestados se les ha solicitado información sobre su historia clínica mediante WhatsApp y Páginas Web, un 26.9% se le ha solicitado por Esquela y el 22.2% no se le ha solicitado. Lo que evidencia que a la mayoría de estudiantes se les ha solicitado información sobre sus historias clínicas por parte del departamento medico utilizando algún medio de comunicación.

Pregunta 3

¿En caso de algún accidente usted sabe a dónde acudir?

108 respuestas

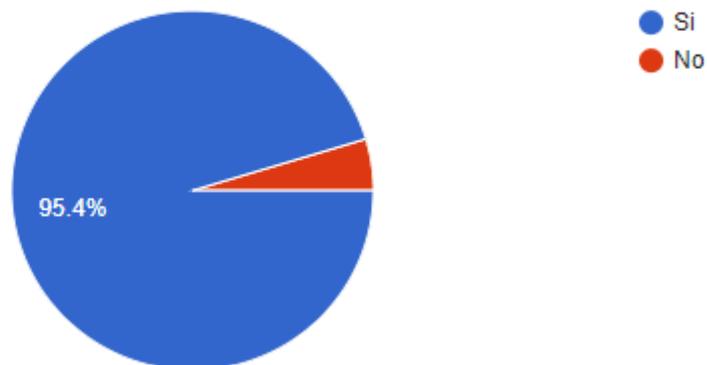


Figura 2.3: Conocimiento de la ubicación del Departamento Medico.

Elaborado por: Investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.3, el 95.4% de los estudiantes saben a dónde acudir en caso de un accidente dentro de la institución y solo el 4.6% de los estudiantes no saben a dónde acudir. Se puede evidenciar que la mayoría de los estudiantes saben a dónde deben ir en caso de algún accidente gracias a las capacitaciones o recorridos que se realiza en la institución, mientras que los estudiantes que ni saben a dónde acudir puede ser porque son nuevos o no tienen conocimiento de la existencia del departamento médico.

Pregunta 4

¿Considera usted que las instalaciones del departamento médico de la institución cuentan con lo necesario en caso de una emergencia?

108 respuestas

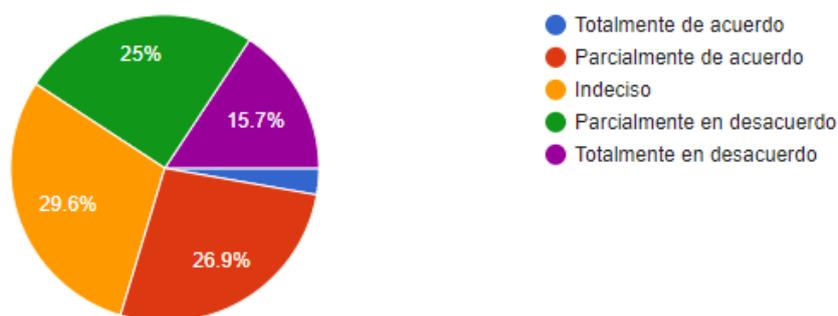


Figura 2.4: Instalaciones del Departamento Médico.
Elaborado por: Investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.4, el 29.6% de los estudiantes están indeciso en que el departamento médico de la institución cuente con lo necesario en caso de alguna emergencia dentro de la institución, mientras el 26.9% están parcialmente de acuerdo, el 25% esta parcialmente en desacuerdo, el 15.7 está totalmente en desacuerdo y solo el 2.8% están totalmente de acuerdo. Por lo tanto, el mayor número de estudiantes sostienen que se podría mejorar e implementar lo necesario al departamento médico para que pueda actuar de manera eficiente en caso de un accidente.

Pregunta 5

¿Cómo calificaría la atención medica de la institución?

108 respuestas

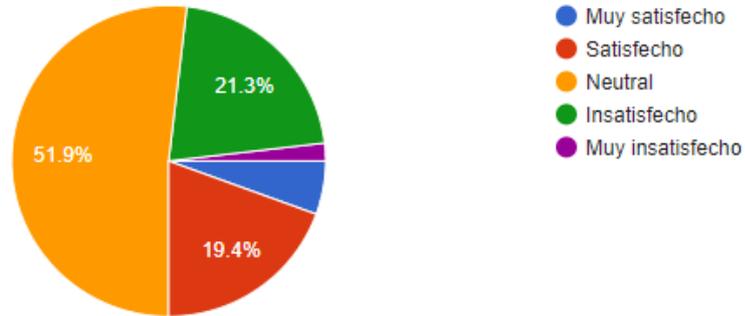


Figura 2.5: Calificación del Departamento Medico.

Elaborado por: Investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.5, el 51.9% de los estudiantes califican la atención medica de la institución de manera Neutral, mientras el 21.3% se encuentra insatisfecha, el 19.4% está satisfecha con la atención, el 5.6% está muy satisfecho y el 1.9% está muy insatisfecho. Por lo tanto, el mayor número de estudiantes sostienen que la atención medica de la institución no es ni buena ni mala, sino que se encuentra de manera neutral cumpliendo con su función.

Pregunta 6

¿Está usted de acuerdo en que, el departamento médico de la institución lleve el registro de fichas medicas de docentes y estudiantes para poder actuar de manera eficiente en caso de un accidente?

108 respuestas

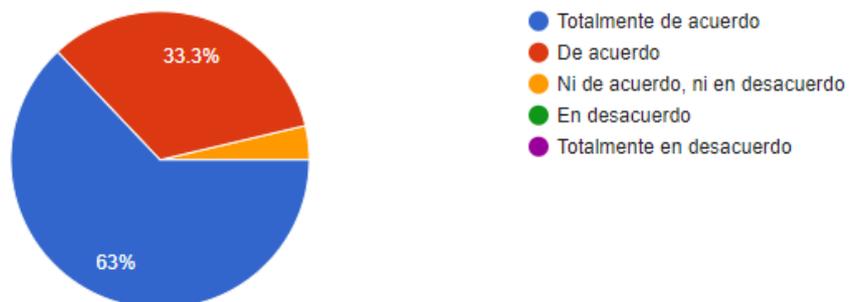


Figura 2.6: Control de Fichas Medicas.

Elaborado por: Investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.6, el 63% de los estudiantes están totalmente de acuerdo que el departamento médico de la institución lleve consigo el registro de fichas médicas, mientras el 33.3% está de acuerdo y solo el 3.7 % está en ni acuerdo, ni en desacuerdo. Por lo tanto, los estudiantes consideran que llevar el registro de las fichas medicas podrían ayudar al departamento médico a actuar de manera más eficiente en caso de algún accidente.

Pregunta 7

¿El médico de la institución realiza preguntas referentes a su ficha médica?

108 respuestas

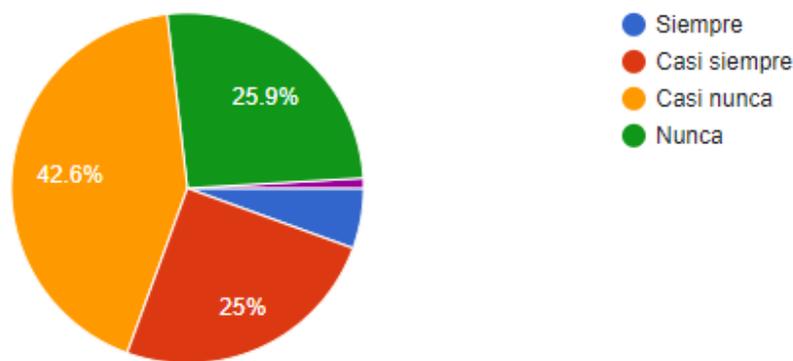


Figura 2.7: Preguntas referentes a las fichas médicas.

Elaborado por: Investigador

Según los resultados de la Figura 2.7, el 42.9% de los estudiantes dicen que casi nunca el médico de la institución les hace preguntas referentes a las fichas médicas, mientras el 25.9% dice que nunca, el 25% dice Casi siempre y solo el 6.8% dice que siempre. Por lo tanto, se puede deducir que el médico de la institución no realiza preguntas sobre las fichas medicas porque estas deben estar registradas, mientras que a los estudiantes que respondieron que siempre o casi siempre es porque el medico no tiene ninguna información de los estudiantes.

Pregunta 8

¿Conoce usted cuánto tiempo usualmente se demora el médico de la institución para atender a un docente o estudiante?

108 respuestas

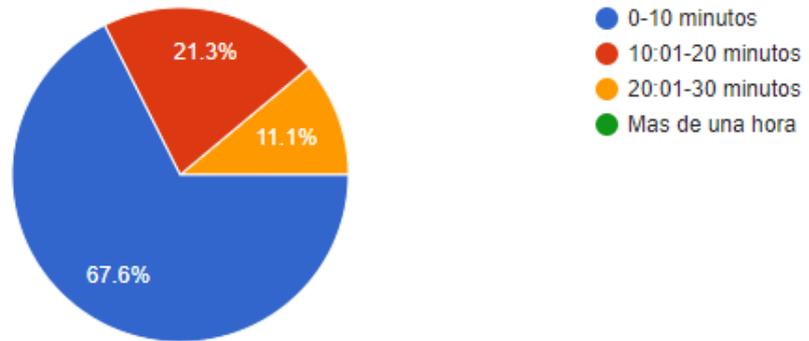


Figura 2.8: Tiempo de demora para la atención médica institucional.

Elaborado por: Investigador

Según los resultados de la Figura 2.8, el 67.6% de los estudiantes respondieron que usualmente el médico para atender a un estudiante se demora de 0-10 minutos, mientras el 21.3% respondió que se demora entre 10:01 – 20 minutos y solo el 11.1% dijo que se demora de 20:01 – 30 minutos. Por lo tanto, el mayor número de estudiantes sostiene que el médico de la institución se demora de 0-10 minutos para atender a un estudiante.

Resultados de las encuestas aplicada a los docentes de la Unidad Educativa González Suárez.

Pregunta 1

¿Cree usted que las fichas médicas tanto de docentes como estudiantes son necesarias en la institución?

65 respuestas

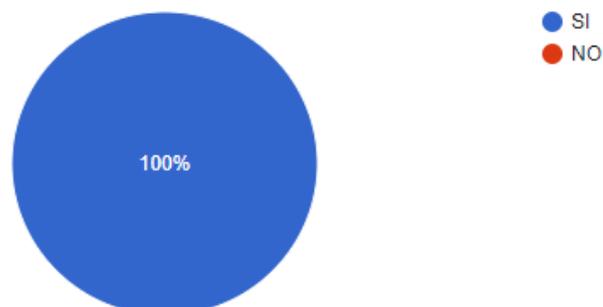


Figura 2.9: Fichas Médicas Necesarias en la Institución.

Elaborado por: Investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.9, el 100% de los docentes creen que son necesarias las fichas médicas tanto de docentes y estudiantes de la institución. Por lo tanto, la totalidad de los docentes sostiene que las fichas medicas son necesarias dentro de la institución.

Pregunta 2

¿Se le ha solicitado a usted información sobre su historia clínica por parte del departamento médico de la institución?

65 respuestas

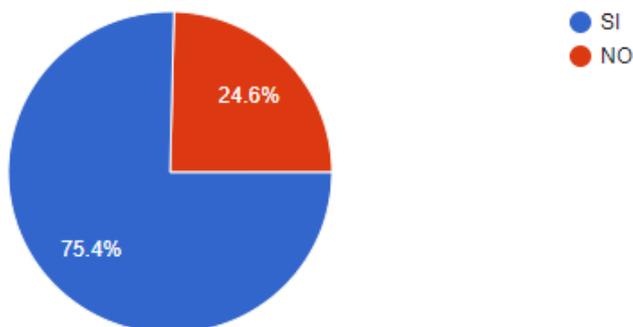


Figura 2.10: Solicitud de información sobre las historias clínicas

Elaborado por: Investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.10. El 75.4% de los docentes que correspondían a más de la mitad de los encuestados se le ha solicitado información de las historias clínicas y al 24.6% no se le ha solicitado o no se ha pedido ninguna información. Por lo que se puede identificar que a pesar de que la mayoría de los docentes se le ha solicitado información de las historias clínicas existe una cierta cantidad de docentes a los cuales no se les ha solicitado por lo que se haría difícil al departamento médico llenar sus fichas.

Pregunta 3.

¿Qué medio de comunicación ha utilizado el departamento para pedir información sobre su historia clínica?

64 respuestas

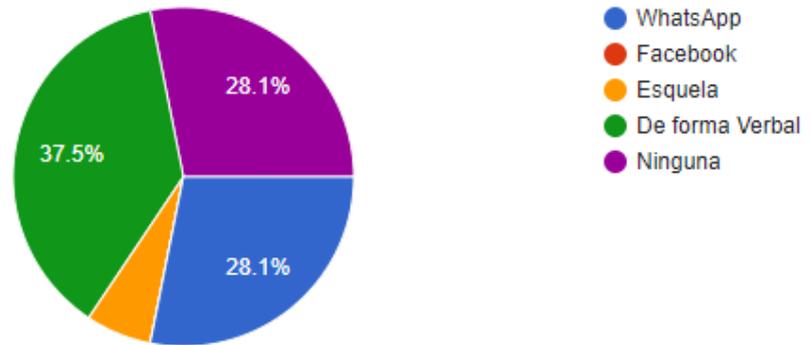


Figura 2.11: Medio de recolección de información por parte del departamento médico.

Elaborado por: Investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.11. El 37.5% de los encuestados se les ha solicitado información sobre su historia clínica de forma verbal, un 28.1% se le ha solicitado por WhatsApp, un 28.1% no se le ha solicitado ninguna información mientras que al 6.3% se le ha solicitado información mediante esquela. Lo que evidencia que a la mayoría de docentes se les ha solicitado información sobre sus historias clínicas por parte del departamento médico utilizando la forma verbal.

Pregunta 4

¿Cree usted que es eficiente en la actualidad que la institución lleve de manera manual las fichas medicas?

65 respuestas

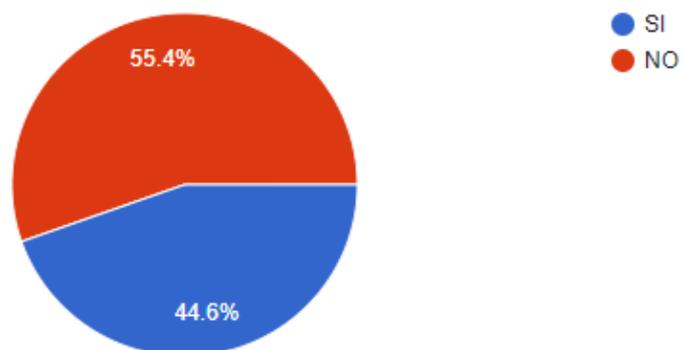


Figura 2.12: Efectividad en el manejo manual de las fichas médicas.

Elaborado por: Investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.12, el 55.4% de los docentes creen no es eficiente que en la actualidad la institución lleve de manera manual las fichas médicas y el 44.6% de los docentes creen que si es eficiente el manejo de las fichas médicas de manera manual. Por lo que se puede evidenciar que la mayoría de los docentes creen que la manera manual que se lleva el control de las fichas médicas en la institución no es eficiente en la actualidad.

Pregunta 5

¿Creó usted que un sistema web para gestionar las fichas médicas ayudaría a encontrar la información de manera más rápida y sencilla?

65 respuestas

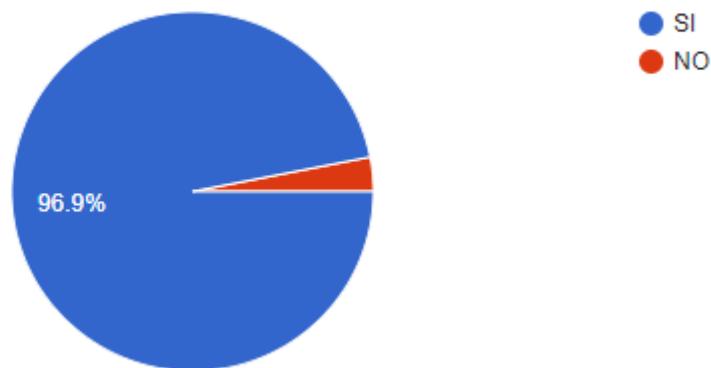


Figura 2.13: Gestión de fichas medicas mediante un sistema web.

Elaborado por: Investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.13, el 96.9% de los docentes creen que un sistema web para la gestión de ficha médicas ayudaría a encontrar la información de una manera más rápida y sencilla y el 3.1% de los docentes creen que un sistema web no ayudaría a encontrar la información de manera rápida y sencilla. Por lo que se puede evidenciar que la mayoría de los docentes creen que un sistema web para la gestión de fichas médicas ayudaría en la búsqueda de la información de manera rápida y sencilla ya que en caso de necesitar la ficha médica de un docente solo se le pediría nombre, apellidos o número de cedula y se obtendría la información y se evitaría la pérdida de

tiempo en buscar de manera manual la ficha médica ya que se tendría que traspapelar todo.

Pregunta 6

¿Creé usted que gestionar las fichas médicas de los estudiantes y docentes de manera digital ayudé a mejorar al departamento médico?

64 respuestas

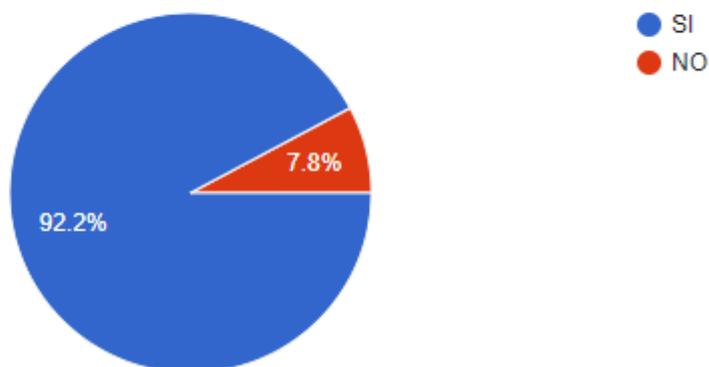


Figura 2.14: Mejoramiento del Departamento Médico.

Elaborado por: Investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.14, el 92.2% de los docentes creen gestionar de las fichas médicas digitalmente ayudara a mejorar al departamento médico de la institución y el 7.8% de los docentes creen que digitalizar las fichas no ayudara en nada al departamento médico. Por lo que se puede evidenciar que la mayoría de los docentes creen que la digitalizar las fichas medicas ayudara al departamento médico a que este lleve un mejor control de las fichas médicas en la institución.

Pregunta 7

¿Ha tenido algún inconveniente con su ficha médica por parte del departamento médico?

Seleccione cual es el inconveniente que se ha presentado por parte del departamento médico.

64 respuestas

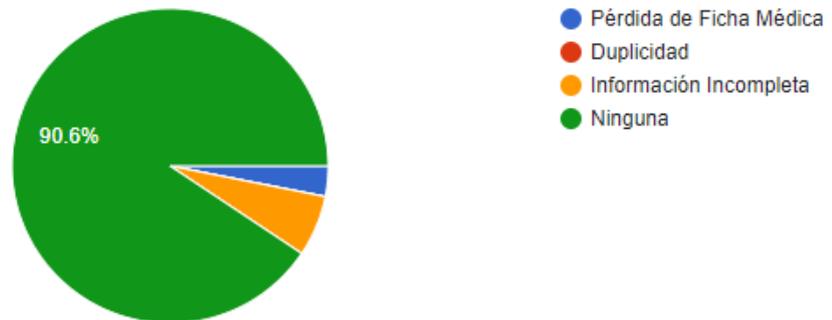


Figura 2.15: Inconvenientes con las Fichas Médicas.
Elaborado por: Investigador

Análisis e Interpretación de resultados

Según los resultados de la Figura 2.15. El 90.6% de los encuestados no tienen ningún problema con las fichas médicas, el 6.3% tienen fichas médicas incompletas y un 3.1% no posee ficha médica por su pérdida. Lo que evidencia que a la mayoría de docentes no tienen ningún inconveniente con las fichas médicas.

2.2.4. Procesamiento y Análisis de Datos

De acuerdo con la información recolectada se puede decir que

- La mayoría de estudiantes no cuentan con fichas medicas completas por lo que dificulta al médico de la institución actuar de manera eficiente y rápida.
- La atención de los estudiantes por parte del departamento médico es un poco demorosa ya que, al no contar con los datos suficientes de las fichas médicas no pueden actuar de manera rápida sin la aprobación de los padres de familia.
- Se debe mejorar la recolección de la información y el manejo de las fichas médicas estudiantiles.
- Es muy importante llevar un control de las fichas medicas tanto de docentes como estudiantes para en caso de una emergencia saber si tiene alguna enfermedad crónica o si es alérgico a algo.
- El manejo de la información actual de cada estudiante y docente no es la adecuada, debido que, se encuentra almacenada físicamente en archivadores lo que resulta difícil la búsqueda de las fichas médicas y su acceso a ellas.

- Según los resultados obtenidos se puede determinar en caso de los estudiantes los padres de familia no llenan la información completa o los estudiantes no hacen saber a los padres de familia que se les pidió esta información.
- Llevar las fichas de manera electrónica es muy eficiente ya que se puede obtener la información de forma rápida y precisa.

CAPÍTULO III.- RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de los resultados

3.1.1. Procesos para la toma de información de las fichas médicas.

Una vez obtenida la información mediante los instrumentos de recolección, se puede determinar los procesos que se utiliza dentro de la Unidad Educativa González Suárez para la creación de fichas medicas tanto de docentes como estudiantes, logrando así comprender el manejo de fichas médicas que mantiene la institución, se identificaron inconvenientes al rato de registrar las fichas médicas ya que existe duplicidad de información, falta de información o fichas sin llenar y dificultad en la búsqueda de fichas médicas, razón por la que la institución decidió gestionar las fichas medicas de manera virtual.

Proceso para la toma de información de fichas medicas de los docentes.

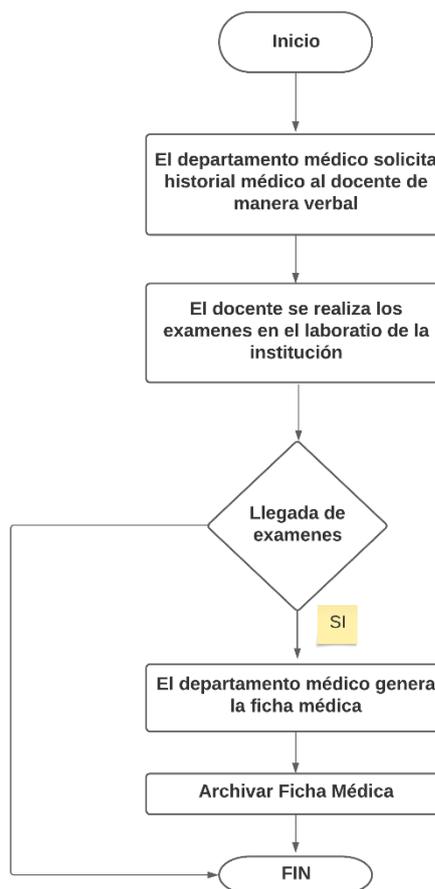


Figura 3.1: Proceso para llenado de fichas médicas de los docentes.

Elaborador por: Investigador.

- Se pide de manera verbal, a los docentes realizarse exámenes con el laboratorio que tiene convenio con la institución.

- Llegado los resultados los docentes deben entregar al departamento médico para que se pueda generar la ficha médica.
- La encargada del departamento médico llena las fichas medicas de manera manual.
- Archiva las fichas médicas de los docentes.

Proceso para la toma de información de fichas médicas actualizado de los docentes.

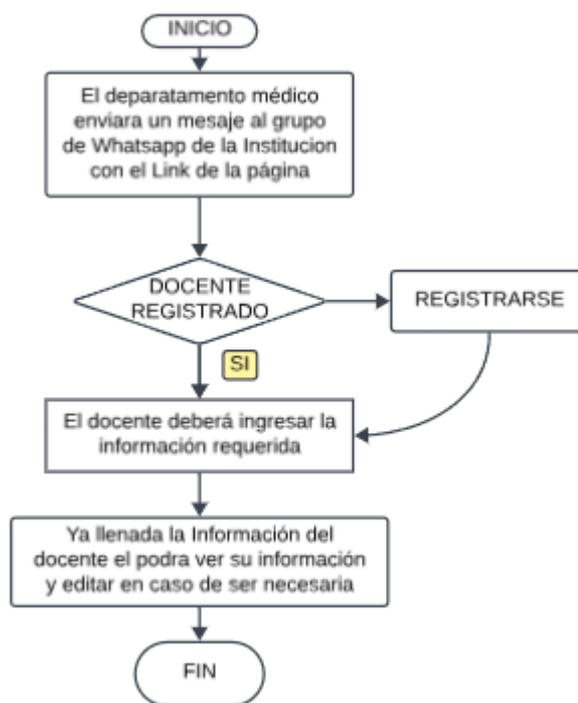


Figura 3.2: Proceso actualizado para llenado de fichas médicas de los docentes.

Elaborador por: Investigador.

- Se remite un mensaje de Whatsapp al grupo de la Institución para que registren la ficha médica.
- Los docentes ingresan al sistema si tienen ya una cuenta, caso contrario deberán registrarse.
- El docente deberá llenar la información requerida.

Proceso para la toma de información de fichas médicas de los estudiantes.

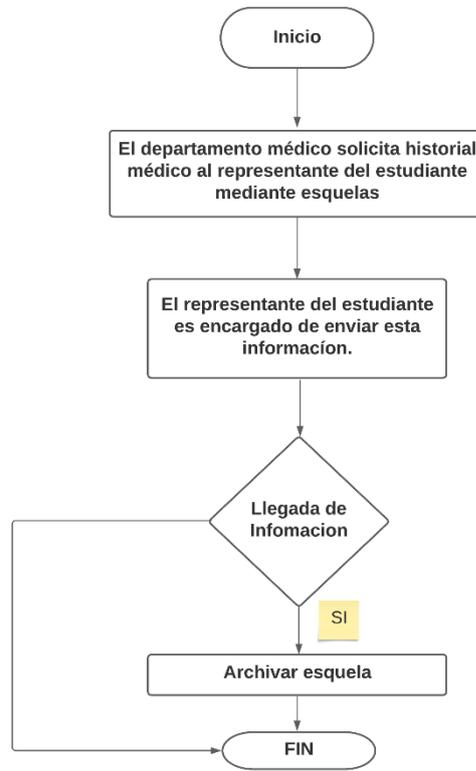


Figura 3.3: Proceso para llenar las fichas médicas de los estudiantes.

Elaborado por: Investigador

- Se solicita por esquela o medios tecnológicos a los padres de familia que otorgue la información.
- Llegada la información solicitada de los estudiantes se procede enviar al departamento médico.
- Archiva las esquelas porque no se dispone de un modelo de ficha médica para los estudiantes.

Proceso para la toma de información de fichas médicas actualizado de los estudiantes.

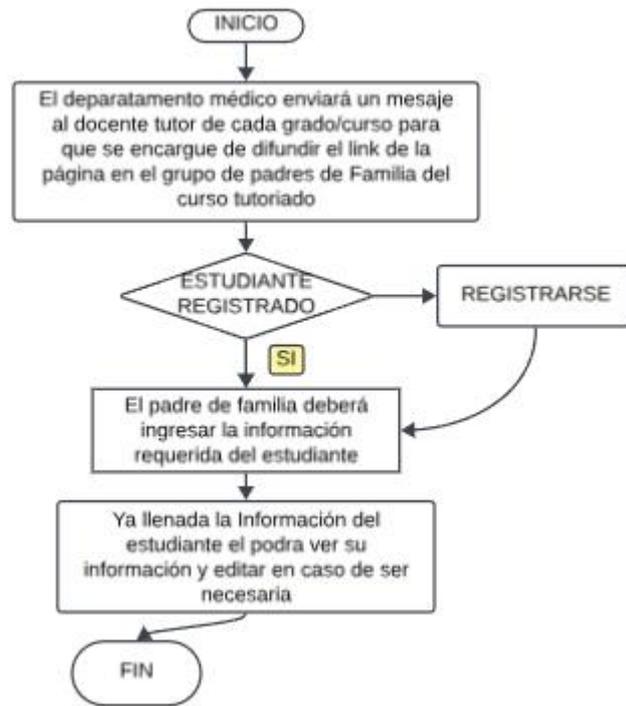


Figura 3.4: Proceso actualizado para llenar las fichas médicas de los estudiantes.

Elaborado por: Investigador

- El encargado del departamento médico remitirá un mensaje a los tutores de cada grado/curso el mismo que se encargará de enviar el link al grupo de padres de familia.
- Los padres de familia deberán iniciar sesión si el estudiante tiene una cuenta caso contrario deberán registrar al estudiante.
- Los padres de familia ingresaran la información requerida del estudiante.

3.1.2. Sistema web basado en la arquitectura MVC

Definición técnica de la arquitectura MVC

Según Manzano Rojas Edison Javier [3] se puede considerar que “La arquitectura MVC es la única solución de desarrollo Web en el lado servidor existente en la actualidad que permite separar la parte lógica de la parte de presentación de una aplicación Web. Esto es muy importante, porque permite que el Desarrollador y el Diseñador Web trabajen por separado en un proyecto, cada uno sobre sus archivos o componentes”.

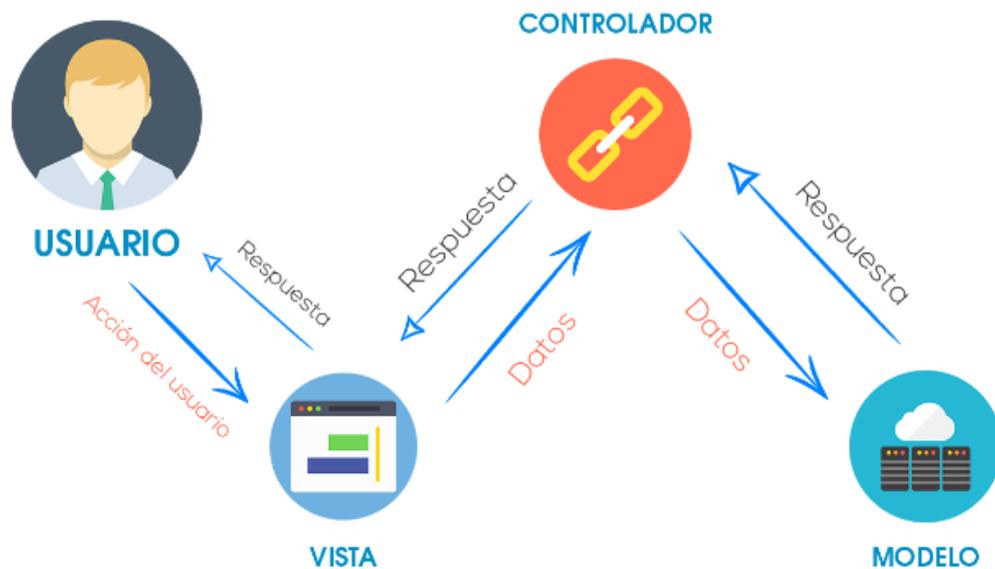


Figura 3.5: Arquitectura MVC.

Elaborado por: Investigador

Niveles de la arquitectura MVC

Modelo: Es la lógica de negocio de nuestro software, comúnmente hace referencia, pero no está limitado a, la base de datos, entidades, repositorios y consultas. Todo lo referente a manipulación de datos en el nivel más bajo, se encuentra en el modelo, lo que llamamos el CRUD de nuestro sistema (Create, Read, Update, Delete o en español: Crear, Leer, Actualizar, Eliminar), las cuatro operaciones esenciales que podemos realizar con nuestros datos, además de la validación de los mismos [14].

Vista: Es la interfaz de usuario, todo lo referente a la presentación de nuestra aplicación estará de forma organizada en múltiples vistas, en las cuales no habrá ningún tipo de lógica compleja. Las vistas deben estar limitadas a mostrar información únicamente, con lógica sencilla como: impresión de variables, condicionales y ciclos para repetir un elemento; toda esta información debe ser procesada y validada previamente por el controlador y el modelo, de tal forma que se reduzca al mínimo la programación en la vista [14] las consultas de base de datos utilizando técnicas como eager loading (carga ansiosa), selección selectiva de columnas y uso de índices adecuados.

3.1.3. Tecnologías web basadas en la arquitectura MVC

Característica	Tecnologías Web Basadas en la Arquitectura MVC		
	LARAVEL	ASP.NET MVC	RUBY ON RAILS
Lenguaje	PHP	C#	Ruby
Popularidad	Alta	Alta	Alta
Curva de Aprendizaje	Moderada	Requerir un tiempo adicional para aprender los conceptos y la sintaxis específica de C#	Es posible que te resulte relativamente fácil aprender Ruby on Rails
Arquitectura	MVC	MVC	MVC
Flexibilidad	Ofrece una arquitectura modular que permite a los desarrolladores elegir qué componentes utilizar y cómo estructurar su aplicación	Ofrece una buena flexibilidad en términos de estructura y configuración de la aplicación	Destaca por su enfoque en la convención sobre la configuración y la programación orientada a convenciones
Rendimiento	Es conocido por su rendimiento sólido y eficiente. El framework utiliza el motor de plantillas Blade y proporciona una estructura de archivos organizada que permite una carga rápida de páginas. Además, Laravel ofrece características	Impulsado por el framework .NET de Microsoft, es reconocido por su rendimiento rápido y de escalabilidad. El framework se beneficia del rendimiento del lenguaje C# y la infraestructura de .NET, que incluye la compilación JIT	Es conocido por su enfoque en la productividad del desarrollador, pero puede tener un rendimiento ligeramente inferior en comparación con otros frameworks. El rendimiento de Ruby on Rails puede verse afectado por la naturaleza interpretada del

	como el almacenamiento en caché, la optimización de consultas de bases de datos y el enrutamiento eficiente, lo que contribuye a mejorar el rendimiento de las aplicaciones.	(Just-In-Time) y la gestión avanzada de memoria. ASP.NET MVC también ofrece opciones de almacenamiento en caché, optimización de consultas y una amplia gama de herramientas de rendimiento y monitorización.	lenguaje Ruby y el uso de una capa de abstracción ORM (Mapeo Objeto-Relacional) llamada ActiveRecord.
Comunidad	Grande	Grande	Grande
Documentación	La documentación oficial de Laravel es muy completa y está bien organizada. Proporciona guías detalladas, ejemplos de código, explicaciones de conceptos clave y referencias a todas las características y componentes del framework.	La documentación oficial de ASP.NET MVC es proporcionada por Microsoft y ofrece una amplia gama de recursos, tutoriales y ejemplos. Cubre todos los aspectos del desarrollo en ASP.NET MVC, incluyendo el uso de controladores, vistas, enrutamiento, modelos, autenticación y más. La documentación está actualizada y es mantenida por Microsoft.	La documentación oficial de Ruby on Rails está disponible en varios formatos, incluyendo guías, referencia de API y tutoriales. Proporciona información detallada sobre cómo desarrollar aplicaciones web con Ruby on Rails, incluyendo aspectos como enrutamiento, modelos, vistas, controladores, bases de datos y más.
Soporte	Activo	Activo	Activo

Escalabilidad	Proporciona una estructura modular y flexible que permite a los desarrolladores escalar su aplicación de acuerdo a las necesidades específicas.	Es altamente escalable y está diseñado para manejar aplicaciones web de gran envergadura	Tiene una buena capacidad de escalabilidad y ha demostrado ser exitoso en el manejo de aplicaciones web de gran envergadura
ORM	Eloquent ORM	Entity Framework	ActiveRecord
Seguridad	<ol style="list-style-type: none"> 1. Protección contra ataques de CSRF (Cross-Site Request Forgery) mediante el uso de tokens CSRF. 2. Sistema de autenticación incorporado que facilita la implementación de la autenticación de usuarios. 3. Soporte para encriptación de contraseñas utilizando funciones hash seguras. 4. Validación de datos de entrada para prevenir ataques de inyección y XSS (Cross-Site Scripting). 	<ol style="list-style-type: none"> 1. Autenticación y autorización basada en roles integrada a través de ASP.NET Identity. 2. Soporte para protección contra ataques de CSRF y XSS. 3. Validación de modelos y datos de entrada para prevenir ataques de inyección y XSS. 4. Filtros de seguridad a nivel de acción y controlador para proteger los endpoints. 5. Soporte para encriptación de datos y almacenamiento seguro de contraseñas. 	<ol style="list-style-type: none"> 1. Protección contra ataques de CSRF y XSS a través de tokens CSRF y métodos de escape automático de datos. 2. Configuración automática de encabezados de seguridad en las respuestas HTTP para protección contra ataques de clickjacking y sniffing. 3. Validación y filtrado de datos de entrada para prevenir ataques de inyección y XSS. 4. Protección de rutas y endpoints mediante restricciones de acceso y filtros de seguridad.

	<p>5. Middleware para autorización y control de acceso basado en roles y permisos.</p> <p>6. Protección de rutas y endpoints mediante la definición de políticas de acceso.</p> <p>7. Soporte para almacenamiento seguro de datos confidenciales utilizando cifrado.</p> <p>8. Herramientas para manejar el filtrado y sanitización de datos.</p>	<p>6. Integración con el framework de seguridad de .NET, que ofrece características adicionales de seguridad, como el uso de políticas de seguridad y claims-based authentication.</p>	<p>5. Manejo seguro de sesiones y cookies utilizando mecanismos de cifrado y firmado.</p> <p>6. Soporte para autenticación y autorización mediante gemas como Devise y CanCanCan.</p> <p>7. Integración con sistemas de bases de datos seguros y encriptación de datos confidenciales.</p>
Ecosistemas	Laravel Forge, Laravel Mix, Laravel Valet, Laravel Horizon, Laravel Nova.	Visual Studio, NuGet, Microsoft Azure, ASP.NET Identity.	RubyGems
Integración	Buena	Buena	Buena
Frontend	Integración con tecnologías	Integración con tecnologías	Integración con tecnologías

Tabla 3.1: Comparativa entre las mejores tecnologías web basadas en la arquitectura MVC.

Elaborado por: Investigador

En base al análisis comparativo de las mejores tecnologías web basados en la arquitectura MVC realizado en la **Tabla 3.1**, se seleccionó la tecnología de Laravel porque es un framework de desarrollo web de código abierto construido sobre PHP, que sigue el patrón de diseño MVC, además Laravel ofrece una sintaxis elegante y expresiva, así como una estructura y un conjunto de herramientas para facilitar el desarrollo de aplicaciones web robustas y escalables. Laravel se centra en mejorar la productividad del desarrollador al proporcionar características como enrutamiento fácil de usar, manejo de solicitudes y respuestas, ORM (Object-Relational Mapping), la gestión de sesiones y la autenticación integrada, migraciones de base de datos, la seguridad ya que posee herramientas incorporada para mitigar ataques comunes, etc. Laravel cuenta con una popularidad sumamente alta, por lo que tiene una comunidad muy grande, activa y vibrante que ayuda a los desarrolladores a tener éxito, ya que existe una gran cantidad de recursos disponibles, desde tutoriales y documentación hasta paquetes de terceros y eventos en persona, lo que brinda a los desarrolladores una amplia gama de oportunidades para aprender, conectarse y crecer en el mundo de Laravel

3.1.4. Motor de base de datos para Laravel

Características	Motor de Base de Datos en Laravel		
	MySQL	PostgreSQL	SQLite
Tipo de base de datos	Relacional	Relacional	Relacional
Licencia	Código abierto	Código abierto	Dominio Público
Escalabilidad	Buena	Excelente	Limitada
Soporte ACID	Si	Si	Si
Transacciones	Si	Si	Si
Concurrencia	Bloqueo de nivel de fila	Multiversión y bloqueo de nivel de fila	Bloqueo de nivel de base de datos
Popularidad	Muy Popular	Ampliamente utilizado	Ampliamente utilizado en aplicaciones de

			desarrollo y pruebas
Rendimiento	Buen rendimiento general	Excelente rendimiento en operaciones complejas y grandes volúmenes de datos	Rendimiento rápido en operaciones pequeñas
Herramientas	Amplio conjunto de herramientas y administradores	Amplio conjunto de herramientas y administradores	Herramientas básicas
Comunidad	Gran comunidad y soporte activo	Gran comunidad y soporte activo	Comunidad y soporte limitados
Costo	Gratuito	Gratuito	Gratuito

Tabla 3.2: Comparativa de los mejores motores de base de datos para Laravel.

Elaborado por: Investigador

En base al análisis comparativo de los motores de base de datos para Laravel realizado en la **Tabla 3.2**, se seleccionó MySQL porque es una opción popular para el desarrollo de aplicaciones web debido a su fácil uso, escalabilidad, rendimiento rápido y amplia comunidad de usuarios. Su combinación de características lo convierte en una opción atractiva para una amplia gama de proyectos, desde pequeñas aplicaciones hasta sistemas empresariales de alto tráfico.

3.1.5. Metodología ágil para el desarrollo web de un gestor de fichas medicas

Características	Metodologías Ágiles		
	OOHDM	AUP	WSDM
Enfoque	Orientado a objetos	Orientado a objetos y enfoque ágil	Orientado a servicios web y SOA
Propósito	Diseño de aplicaciones hipermediales.	Desarrollo de software en general	Gestión de servicios web distribuidos
Iterativo e Incremental.	No	Si	Si
Planificación	Planificación inicial y fases predefinidas	Planificación adaptable y flexible	Planificación adaptable
Modelo	Modelado conceptual	Modelado orientado a objetos	Modelado de servicios web
Gestión de Requisitos	Identificación y análisis de requisitos	Identificación y priorización de requisitos	Identificación de servicios web
Flexibilidad	Menos flexible	Mayor flexibilidad	Moderada flexibilidad
Documentación	Documentación estructurada y detallada	Documentación ágil y adaptable	Documentación técnica y operativa
Enfoque de desarrollo	Metodología más tradicional	Metodología ágil	Metodología orientada a servicios
Adaptabilidad	Menos adaptable a cambios	Mayor adaptabilidad a cambios	Moderada adaptabilidad
Control de riesgos	Análisis de riesgos estructurado	Análisis de riesgos y enfoque ágil	Control de riesgos de servicios
Comunicación	Comunicación Formalizada y estructurada	Comunicación fluida y constante	Comunicación en servicios web
Fases	<ol style="list-style-type: none"> 1. Análisis de requerimientos 2. Diseño conceptual 3. Diseño navegacional 4. Diseño de presentación 5. Implementación 6. Evaluación y mantenimiento 	<ol style="list-style-type: none"> 1. Inicio (Inception) 2. Elaboración (Elaboration) 3. Construcción (Construction) 4. Transición (Transition) 	<ol style="list-style-type: none"> 1. Análisis de requerimientos 2. Diseño de la arquitectura 3. Diseño del contrato 4. Implementación 5. Pruebas y depuración 6. Despliegue y mantenimiento

Tabla 3.3: Comparativa de la mejor metodología ágil.

Elaborado por: Investigador

Según la **Tabla 3.3** se eligió la metodología AUP debido a que presenta ser la mejor opción para el tipo de proyecto, ya que es una metodología conocida por su gran enfoque adaptable y flexible, también cuenta con un gran enfoque equilibrado ya que combina elementos de desarrollo ágil con las mejores prácticas del proceso unificado de software (RUP) y esto proporciona un equilibrio entre la agilidad y disciplina, además está orientado hacia la arquitectura por lo que proporciona directrices claras para el diseño y construcción de una arquitectura sólida, siendo esto la mejor forma para coordinar con la empresa y así mejorar y potenciar los cambios, entregando así un proyecto con una metodología ágil de calidad.

3.2. Desarrollo de la propuesta

3.2.1. Fase 1: Inicio

En esta fase de análisis de requisitos, se realizó un estudio del proceso de recolección de información para llenar las fichas médicas de los docentes y estudiantes, con el propósito de obtener una lista de ideas sobre el diseño de la interfaz de usuario en el sistema de gestión de fichas médicas.

A. Alcance del Proyecto

El sistema para la gestión de fichas medicas de docentes y estudiantes permitirá:

- Al Administrador agregar, editar, eliminar y ver información de las tablas administrativas (niveleducacion, gradocurso, estadocivil, discapacidad, lateralidad, genero, identidadgenero, orinetacionsexual, religión), docentes o estudiantes, generar reportes y ver los gráficos.
- A la persona encargada del departamento médico registrar, editar, eliminar y ver la información de la ficha médica tanto de docentes como estudiantes, obtener reportes de estudiantes ya se en general o por grados/cursos.
- A los docentes y estudiantes registrar su información para la creación de la ficha médica en la institución.

El sistema no permitirá:

- A los docentes y estudiantes tener acceso a información de las fichas médicas de otros usuarios.

B. Análisis de requisitos

Dentro del análisis de requisitos, se encuentra los requerimientos funcionales y no funcionales del sistema de gestión de fichas médicas de docentes y estudiantes.

Requerimientos funcionales:

- El sistema permitirá tener acceso mediante un login y rol.
- El sistema permitirá restablecer la contraseña.

- El sistema permitirá a la encargada del departamento médico registrar información de las fichas médicas de docentes y estudiantes.
- El sistema permitirá a la encargada del departamento médico generar reporte de fichas médicas de estudiantes.
- El sistema permitirá a los docentes y estudiantes registrar información para su ficha médica en la institución.
- El sistema permitirá generar reportes en .pdf de las fichas medicas de docentes en caso de los estudiantes se podrá generar mediante filtros ya sea por grado/curso.

Requerimientos no funcionales:

- El sistema deberá funcionar y visualizarse correctamente en cualquier navegador como Microsoft Edge, Google Chrome, Mozilla, Opera, entre otros.
- El sistema debe responder de una forma rápida a la generación de reportes.
- El sistema deberá ser sencillo, intuitivo y de fácil aprendizaje, además de cumplir con la funcionalidad requerida.

C. Preparar el entorno de trabajo

Usuario/Rol	Software	Hardware
Administrador	Laravel	Hosting y Servidor de Hostinger.
Departamento Médico.	Bootstrap	
Docente	MySql	
Estudiante	Xampp	
	Apache	
	Composer	
	Arquitectura MVC	

Tabla 3.2.1: Entorno de Trabajo.

Elaborado por: Investigador.

D. Identificar los procesos de negocio

El proceso de negocio lo podemos identificar en la **Figuras 3.1** y en la **Figura 3.2** donde se encuentra detallado cual es el proceso que realiza la encargada del departamento médico de la institución para la toma de información tanto de docentes como estudiantes para crear su ficha médica dentro de la institución.

3.2.2. Fase 2: Elaboración

A. Usuarios del sistema

Una vez identificado los usuarios en el modelo del negocio y alineando los requerimientos solicitados, se procede a implementar los usuarios del sistema como se muestra en la **Figura 3.2.1**

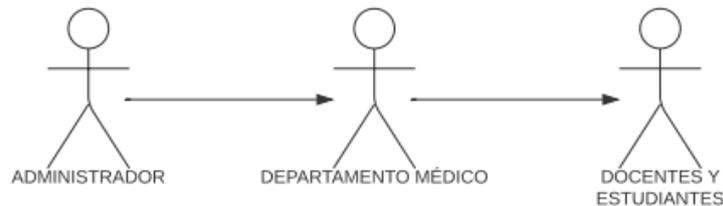


Figura 3.2.1: Usuarios del Sistema.

Elaborador por: Investigador.

B. Casos de Uso del Sistema

Diagrama de Casos de Uso

El proyecto está dividido en 9 casos de uso y se han identificado como usuarios de la aplicación en la **Figura 3.2.1**.

Inicio de Sesión: El propósito de este caso de uso es permitir el acceso a los usuarios.

Gestión Roles: El propósito de este caso de uso es crear roles que los usuarios tendrán en el registro, los roles pueden ser Administrador, Departamento Médico, Docente o Estudiante.

Gestión Usuarios: El propósito de este caso de uso es permitir el registro de nuevos usuarios según el rol asignado.

Gestión Tablas Administrativas: El propósito de este caso de uso es permitir agregar, eliminar, editar y ver los tipos que existen en cada tabla administrativa ejemplo: tabla administrativa género se puede agregar los tipos masculino y femenino.

Gestión Docentes: El propósito de este caso de uso es permitir agregar, editar, eliminar, ver y generar ficha médica de los docentes si se tiene un rol administrador o departamento médico caso contrario solo podrá registrar información.

Gestión Estudiantes: El propósito de este caso de uso es permitir agregar, editar, eliminar, ver y generar ficha médica de los estudiantes si se tiene un rol administrador o departamento médico caso contrario solo podrá registrar información.

Generar Reporte: El propósito de este caso de uso es permitir generar reportes de estudiantes ya sea por género, tipo de sangre o edades.

Cerrar Sesión: El propósito de este caso de uso es el de permitir a un usuario finalizar temporalmente la permanencia en el sistema.

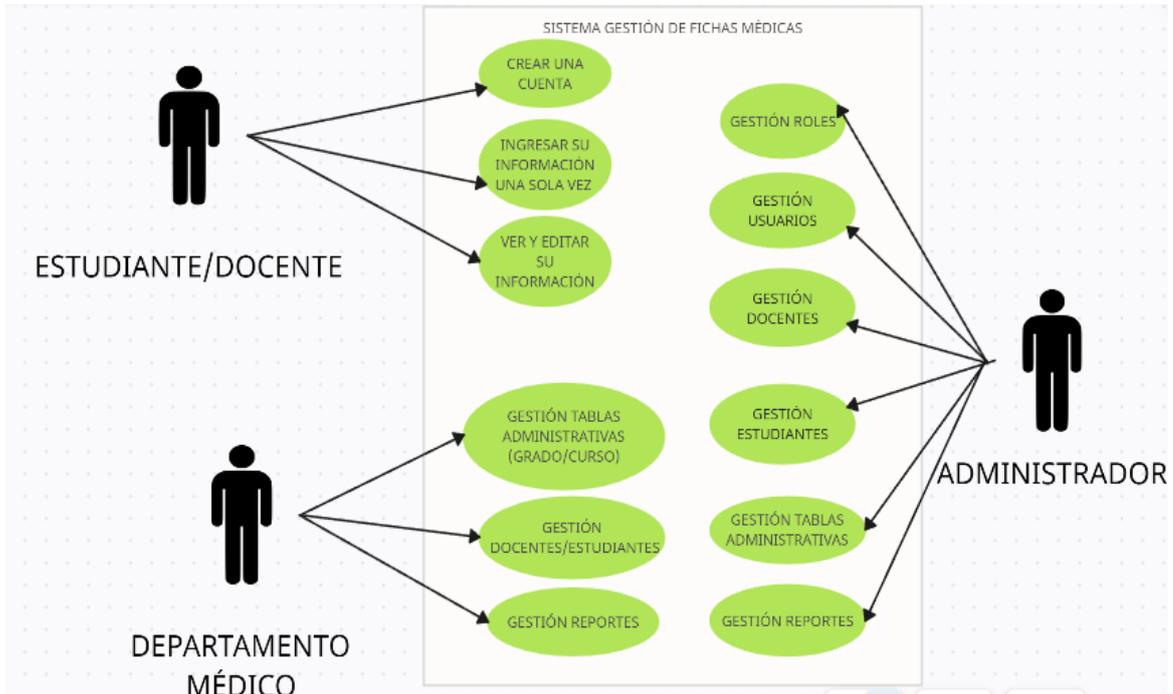


Figura 3.2.2: Diagrama de Casos de uso

Elaborado por: Investigador

C. Especificaciones de Casos de Uso

En esta parte, se realiza una descripción detallada de los casos de uso donde incluyen: precondiciones, postcondiciones, flujo de eventos.

Nombre de Caso de Uso	:	Inicio de Sesión
Objetivo	:	Permitir acceso al sistema.
Actor	:	Administrador, Departamento Médico, Docentes y Estudiantes
Precondiciones	:	Estar registrado en la base de datos.
Flujo Principal	:	<ul style="list-style-type: none"> El usuario ingresa los datos de usuario y contraseña. El sistema verifica si el usuario se encuentra registrado. En caso de estar registrado, el sistema permite visualizar al usuario los módulos correspondientes al rol, caso contrario, se mostrará un mensaje ‘Estas credenciales no coinciden con nuestros registros’.
Flujo Alternativo	:	Verificar que tenga conexión a Internet.
Postcondiciones	:	Si los datos ingresados por el usuario son validados por el sistema y se ha seguido el flujo principal de eventos, se visualiza los módulos del sistema según el rol asignado.

Tabla 3.2.2: Detalle Caso de Uso – Inicio de Sesión

Elaborado por: Investigador

Nombre de Caso de Uso	:	Gestión Roles
Objetivo	:	Registro de roles.
Actor	:	Administrador.
Precondiciones	:	Debe existir un usuario administrador registrado.
Flujo Principal	:	<ul style="list-style-type: none"> • El usuario ingresa al módulo de roles. • Para agregar un nuevo rol el usuario deberá presionar el botón 'AGREGAR ROL' para registrar el rol con sus respectivos permisos. • El usuario presionara el botón 'GUARDAR'. • El sistema verificará los datos registrados en la creación del rol y dirigirá a la pantalla donde se observan los roles registrados. • El usuario podrá editar, eliminar rol según sus necesidades.
Flujo Alternativo	:	Verificar que tenga conexión a Internet.
Postcondiciones	:	Si los datos ingresados son validados por el sistema y se ha seguido el flujo principal de eventos, el formulario queda habilitado para el registro de un nuevo rol.

Tabla 3.2.3: Detalle Caso de Uso – Gestión Roles

Elaborado por: Investigador

Nombre de Caso de Uso	:	Gestión Usuarios
Objetivo	:	Registro de usuarios.
Actor	:	Administrador.
Precondiciones	:	Debe existir un usuario administrador registrado.
Flujo Principal	:	<ul style="list-style-type: none"> • El usuario ingresa al módulo de usuarios. • Para agregar un nuevo usuario el usuario deberá presionar el botón 'AGREGAR USUARIO' para llenar los datos del usuario a registrar. • El usuario presionara el botón 'GUARDAR'. • El sistema verificará los datos registrados en la creación del usuario y dirigirá a la pantalla donde se observan los usuarios registrados. • El usuario podrá editar, eliminar usuario según sus necesidades.
Flujo Alternativo	:	Verificar que tenga conexión a Internet.
Postcondiciones	:	Si los datos ingresados son validados por el sistema y se ha seguido el flujo principal de eventos, el formulario queda habilitado para el registro de un nuevo usuario.

Tabla 3.2.4: Detalle Caso de Uso – Gestión Usuarios

Elaborado por: Investigador

Nombre de Caso de Uso	:	Gestión Tablas Administrativas
Objetivo	:	Registro de los tipos en las tablas administrativas.
Actor	:	Administrador.
Precondiciones	:	Debe existir un usuario administrador registrado.
Flujo Principal	:	<ul style="list-style-type: none"> • El usuario ingresa al módulo de la tabla administrativa que desee agregar el tipo. • Para agregar un nuevo tipo en la tabla administrativa seleccionada el usuario deberá presionar el botón ‘AGREGAR’ para llenar los datos del tipo a registrar. • El usuario presionara el botón ‘GUARDAR’. • El sistema verificará los datos registrados en la creación del tipo y dirigirá a la pantalla donde se observan los tipos registrados. • El usuario podrá editar, eliminar, ver los tipos según sus necesidades.
Flujo Alternativo	:	Verificar que tenga conexión a Internet.
Postcondiciones	:	Si los datos ingresados son validados por el sistema y se ha seguido el flujo principal de eventos, el formulario queda habilitado para el registro de un de tipos en la tabla administrativa.

Tabla 3.2.5: Detalle Caso de Uso – Gestión Tablas Administrativas

Elaborado por: Investigador.

Nombre de Caso de Uso	:	Gestión Docentes
Objetivo	:	Registro de docentes.
Actor	:	Administrador, Departamento Médico, Docentes.
Precondiciones	:	Debe existir un usuario administrador, departamento médico y docente registrado.
Flujo Principal	:	<ul style="list-style-type: none"> • El usuario ingresa al módulo de docentes. • En el caso del rol Administrador y Departamento Médico para agregar un nuevo docente el usuario deberá presionar el botón ‘AGREGAR DOCENTE’ para registrar la información, en el caso contrario este solo deberá dar clic en la pestaña de Docente. • El sistema validará los campos que sean obligatorios en caso de no ingresar los campos obligatorios se mostrará ‘El campo nombre_campo es necesario’. • Si los datos están correctamente ingresados el usuario presionara el botón ‘GUARDAR’. • Si esta logeado como Administrador o Departamento Médico el sistema verificará los datos registrados en la creación del usuario y dirigirá a la pantalla donde se observan los usuarios

		<p>registrados, caso contrario se dirigirá a la información del docente.</p> <ul style="list-style-type: none"> • El usuario podrá si esta logeado con el rol Administrador o Departamento Médico crear, editar, eliminar, ver, generar reporte de ficha medica del docente, caso contrario solo podrá ver su información y editarla en caso de ser necesario.
Flujo Alternativo	:	Verificar que tenga conexión a Internet.
Postcondiciones	:	Si los datos ingresados son validados por el sistema y se ha seguido el flujo principal de eventos, el formulario queda habilitado para el registro de un nuevo docente en el caso de Administrador o Departamento Médico.

Tabla 3.2.6: Detalle Caso de Uso – Gestión Docentes

Elaborado por: Investigador.

Nombre de Caso de Uso	:	Gestión Estudiante
Objetivo	:	Registro de estudiantes.
Actor	:	Administrador, Departamento Médico, Estudiante.
Precondiciones	:	Debe existir un usuario administrador, departamento médico, estudiante registrado.
Flujo Principal	:	<ul style="list-style-type: none"> • El usuario ingresa al módulo de estudiante. • En el caso del rol Administrador y Departamento Médico para agregar un nuevo estudiante el usuario deberá presionar el botón ‘AGREGAR ESTUDIANTE’ para registrar la información, en el caso contrario este solo deberá dar clic en la pestaña de Estudiante. • El sistema validará los campos que sean obligatorios en caso de no ingresar los campos obligatorios se mostrará ‘El campo nombre_campo es necesario’. • Si los datos están correctamente ingresados el usuario presionara el botón ‘GUARDAR’. • Si esta logeado como Administrador o Departamento Médico el sistema verificará los datos registrados en la creación del usuario y dirigirá a la pantalla donde se observan los usuarios registrados, caso contrario se dirigirá a la información del estudiante. • El usuario podrá si esta logeado con el rol Administrador o Departamento Médico editar, eliminar, ver, generar reporte de ficha medica del estudiante, caso contrario solo podrá ver su información y editarla en caso de ser necesario.
Flujo Alternativo	:	Verificar que tenga conexión a Internet.

Postcondiciones	:	Si los datos ingresados son validados por el sistema y se ha seguido el flujo principal de eventos, el formulario queda habilitado para el registro de un nuevo docente.
------------------------	---	--

Tabla 3.2.7: Detalle Caso de Uso – Gestión Usuarios

Elaborado por: Investigador

Nombre de Caso de Uso	:	Generar Reporte
Objetivo	:	Generar reportes de estudiantes en .pdf.
Actor	:	Administrador, Departamento Médico.
Precondiciones	:	Debe existir un usuario administrador y departamento medico registrado.
Flujo Principal	:	<ul style="list-style-type: none"> • El usuario ingresa al módulo de reportes. • Para generar un reporte el usuario deberá seleccionar ‘POR TIPO DE SANGRE’, ‘POR GÉNERO’, ‘POR EDAD’. • El usuario deberá filtrar la información según sus necesidades y presionar el botón ‘FILTRAR’. • Se mostrarán en una tabla los datos filtrados y el usuario deberá presionar el botón ‘GENERAR PDF’. • Se le enviará al usuario al .pdf generado donde podrá descargar o imprimir.
Flujo Alternativo	:	Verificar que tenga conexión a Internet.
Postcondiciones	:	Si se ha seguido el flujo principal de eventos, el generar reporte .pdf queda habilitado para su descarga o impresión.

Tabla 3.2.8: Detalle Caso de Uso – Gestión Usuarios

Elaborado por: Investigador.

Nombre de Caso de Uso	:	Cerrar Sesión
Objetivo	:	Permite el cambio de usuario para inicio de sesión.
Actor	:	Administrador, Departamento Médico, Docentes y Estudiantes.
Precondiciones	:	Haber iniciado Sesión.
Flujo Principal	:	<ul style="list-style-type: none"> • El usuario deberá dirigirse a la parte superior izquierda donde se encuentra su nombre de usuario y deberá presionar en ‘Salir’.
Flujo Alternativo	:	Verificar que tenga conexión a Internet.
Postcondiciones	:	Ninguna.

Tabla 3.2.9: Detalle Caso de Uso – Gestión Usuarios

Elaborado por: Investigador.

D. Diagrama de la base de datos del sistema.

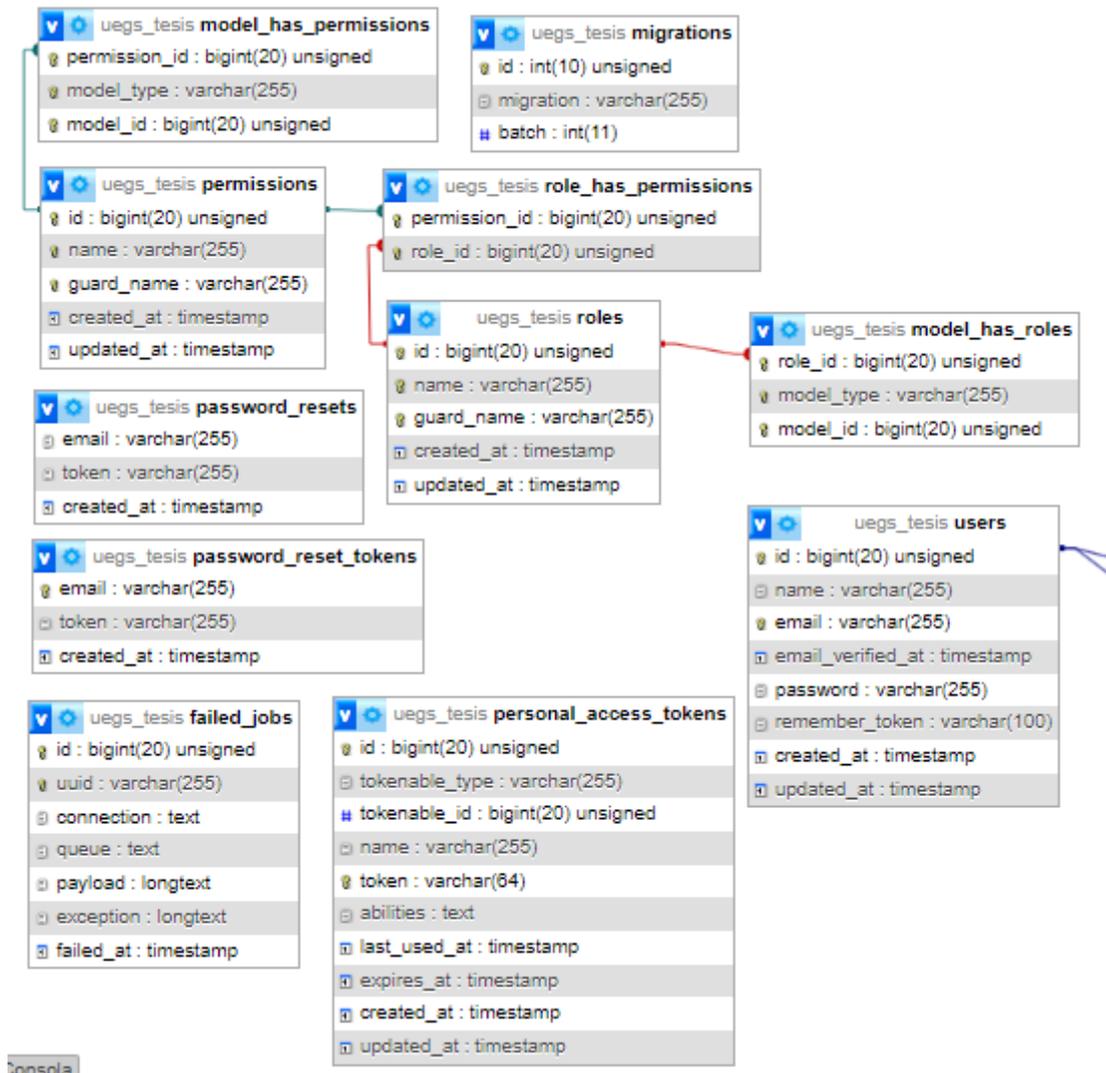


Figura 3.2.3: Diagrama de la Base de Datos para la autenticación de usuarios.

Elaborado por: Investigador.

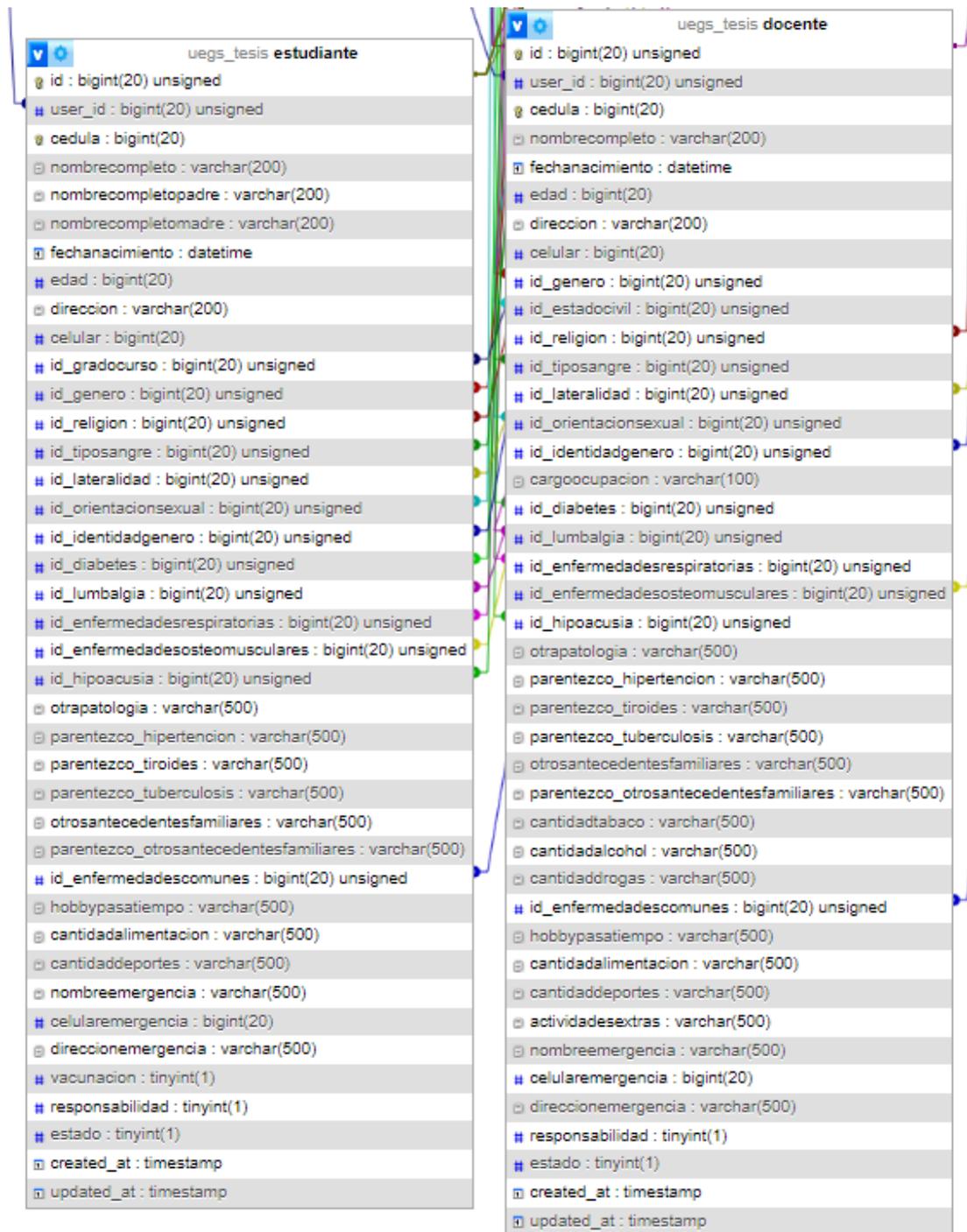


Figura 3.2.4: Tablas de docentes y estudiantes.

Elaborado por: Investigador.

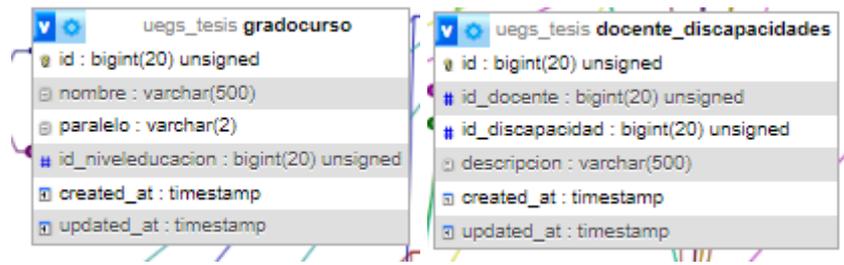


Figura 3.2.5: Ejemplo Tablas Administrativas y Tablas Intermedias (pivot).

Elaborado por: Investigador.

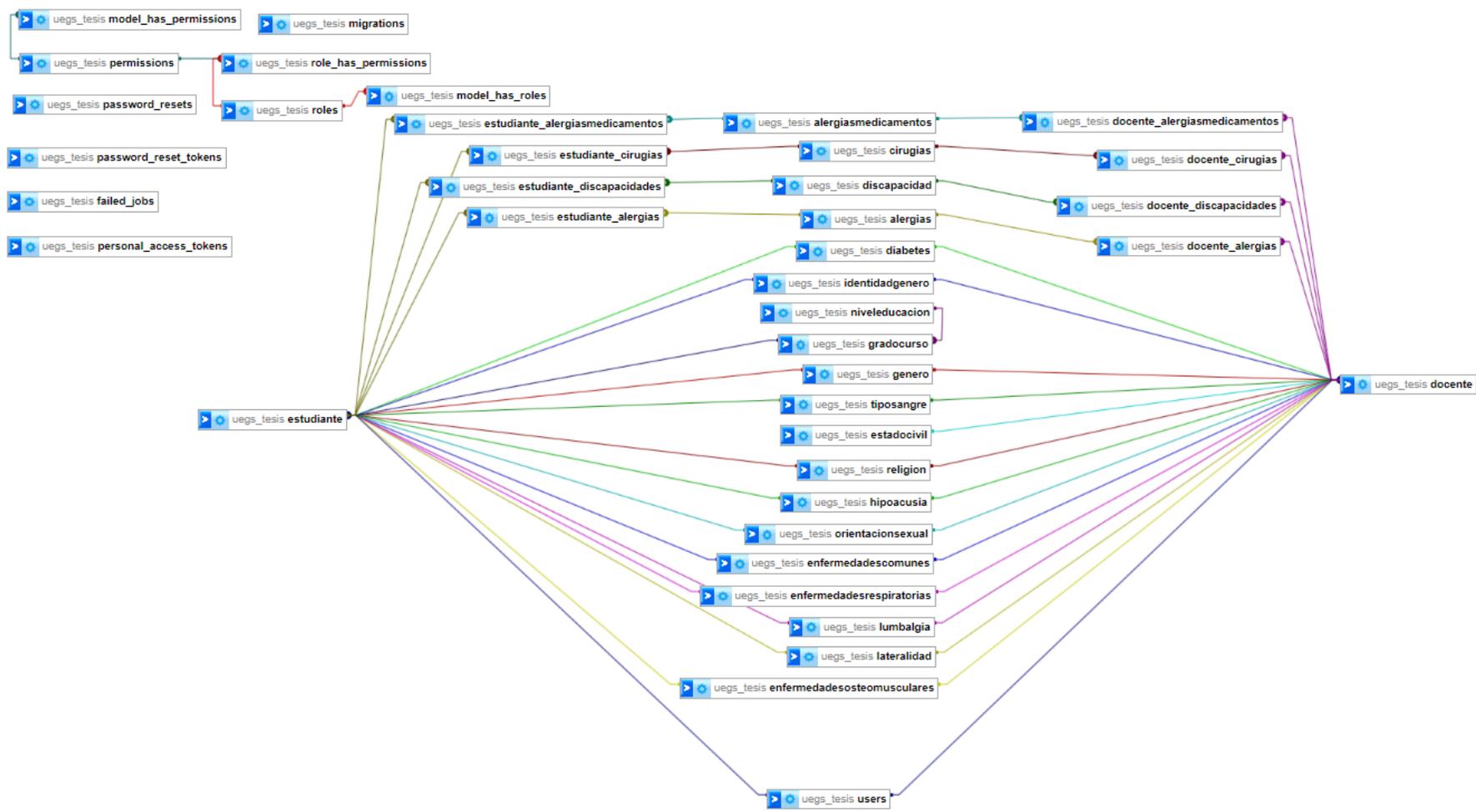


Figura 3.2.6: Diagrama de base de datos para registrar la información de docentes y estudiantes

Elaborado por: Investigador.

E. Arquitectura Lógica.

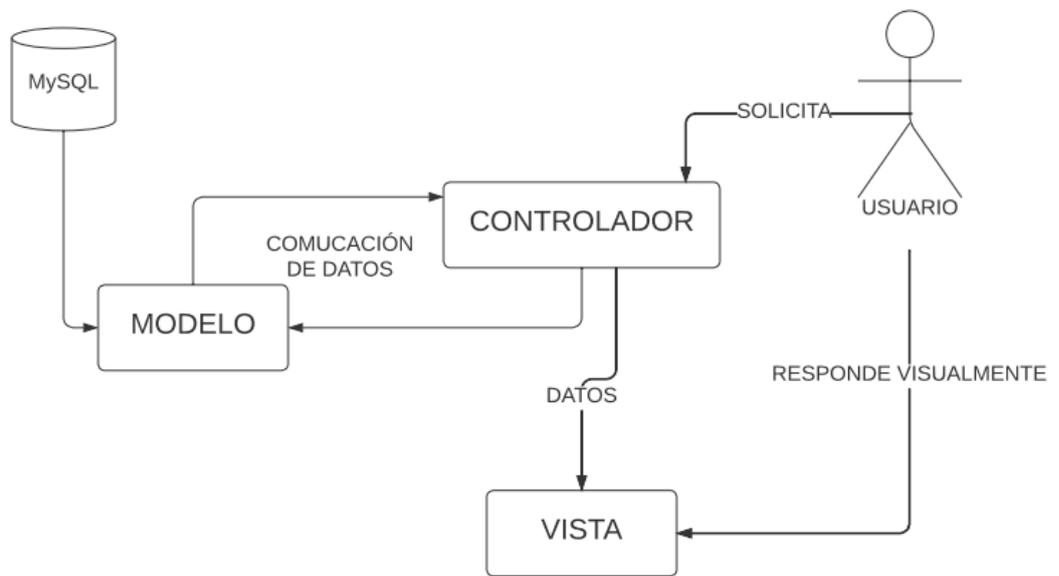


Figura 3.2.7: Arquitectura MVC con Laravel.

Elaborado por: Investigador.

F. Iteraciones del sistema

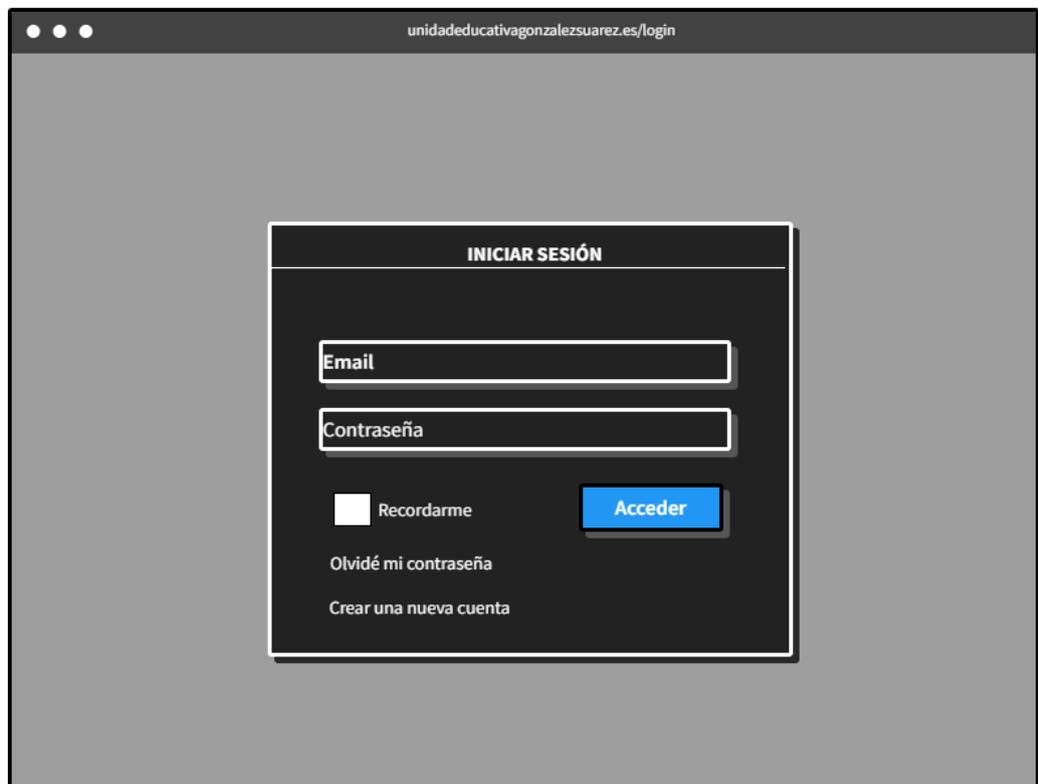


Figura 3.2.8: Inicio de Sesión.

Elaborador por: Investigador.

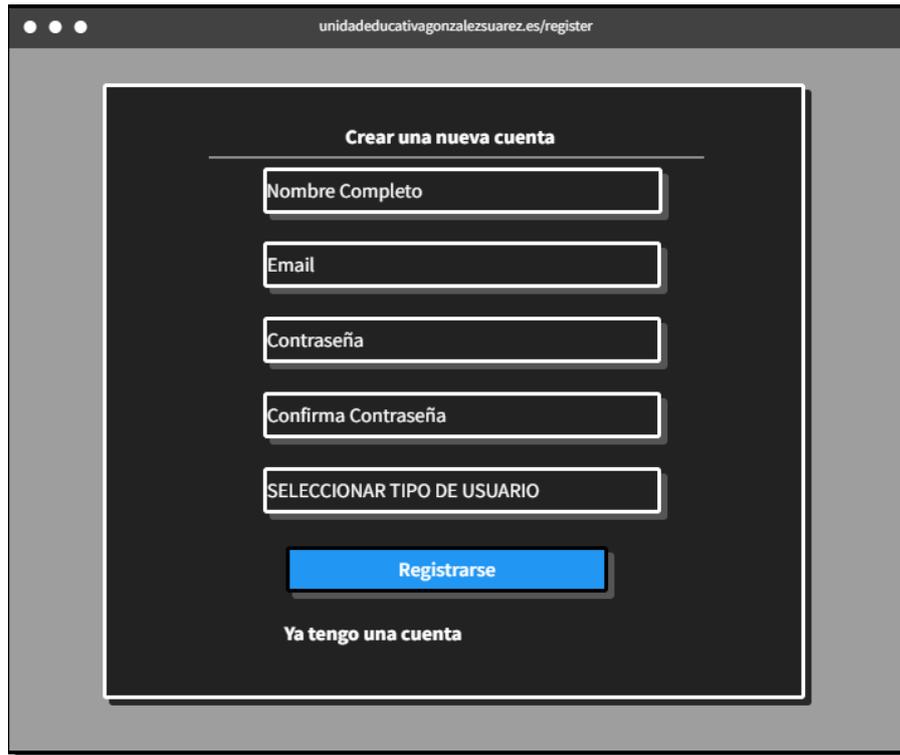


Figura 3.2.9: Registro de usuario.

Elaborado por: Investigador.

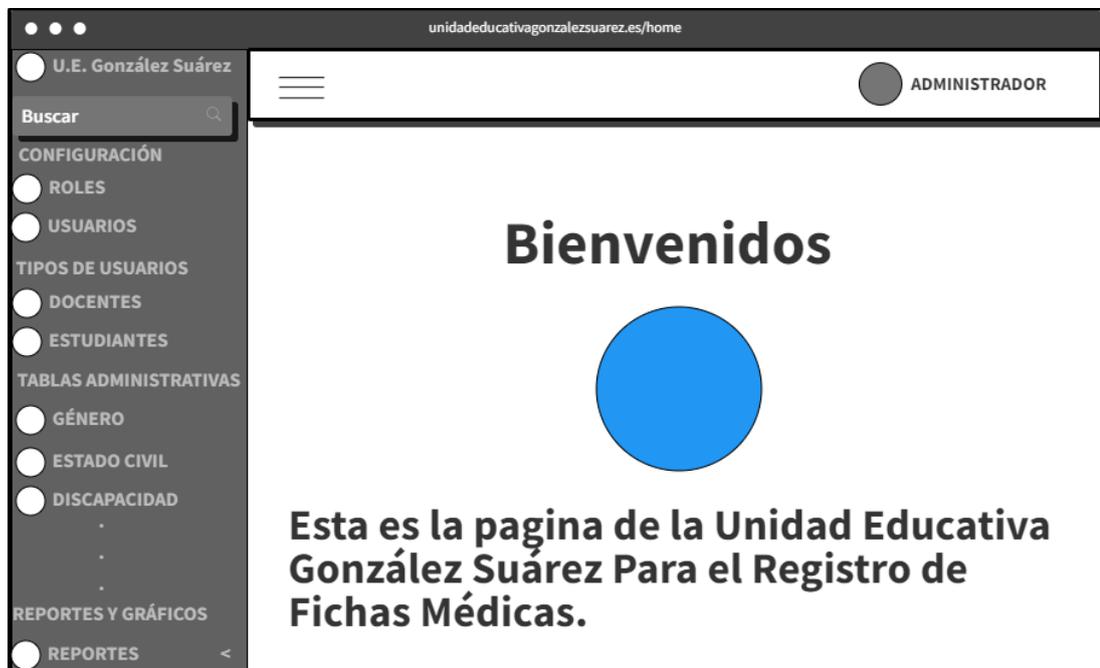


Figura 3.2.10: Vista inicio/home.

Elaborado por: Investigador.

El registro de roles y usuarios dentro del sistema solo lo podrá realizar el administrador.



Figura 3.2.11: Roles de usuario.

Elaborado por: Investigador.

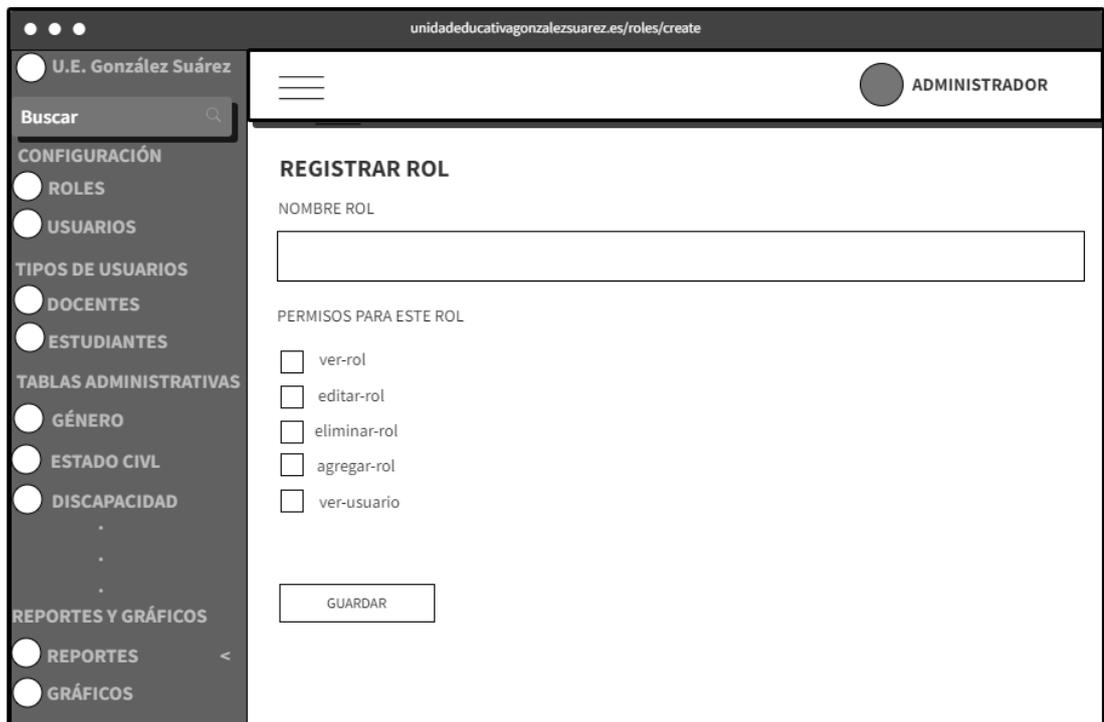


Figura 3.2.12: Registro de roles de usuario con sus respectivos permisos.

Elaborado por: Investigador.

unidadeducativagonzalezsuarez.es/usuarios

U.E. González Suárez

ADMINISTRADOR

Buscar

CONFIGURACIÓN

- ROLES
- USUARIOS
- TIPOS DE USUARIOS
- DOCENTES
- ESTUDIANTES
- TABLAS ADMINISTRATIVAS
- GÉNERO
- ESTADO CIVIL
- DISCAPACIDAD

REPORTES Y GRÁFICOS

- REPORTES

USUARIOS AGREGADOS

AGREGAR USUARIOS

Search...

NOMBRE	EMAIL	ROL	ACCIONES
ADMINISTRADOR	admin@gmail.com	ADMINISTRADOR	EDITAR ELIMINAR
DEPARTAMENTO MÉDICO	departamentomedico@gmail	DEPARTAMENTO MÉDICO	EDITAR ELIMINAR
JUAN GUERRERO	juanandres@gmail.com	DOCENTE	EDITAR ELIMINAR
CARLOS MARTINEZ	carlospepe@gmail.com	ESTUDIANTE	EDITAR ELIMINAR

Figura 3.2.13: Usuarios Registrados

Elaborado por: Investigador.

unidadeducativagonzalezsuarez.es/usuarios/create

U.E. González Suárez

ADMINISTRADOR

Buscar

CONFIGURACIÓN

- ROLES
- USUARIOS
- TIPOS DE USUARIOS
- DOCENTES
- ESTUDIANTES
- TABLAS ADMINISTRATIVAS
- GÉNERO
- ESTADO CIVIL
- DISCAPACIDAD

REPORTES Y GRÁFICOS

- REPORTES

REGISTRAR USUARIO

NOMBRE

EMAIL

CONTRASEÑA

CONFIRMAR CONTRASEÑA

ROL

Figura 3.2.14: Registro de usuarios.

Elaborado por: Investigador.

Para la gestión de docentes y estudiantes tendrán el mismo formato de las Figuras 3.2.12 – 3.2.14 y según el rol como se haya registrado el usuario podrá ver, editar, eliminar y generar ficha médica.

unidadeducativagonzalezsuarez.es/profesor

U.E. González Suárez

ADMINISTRADOR

DOCENTE

AGREGAR DOCENTE

Search...

N°	CEDULA	NOMBRE COMPLETO	DIRECCION	CELULAR	GENERO	ESTADO CIVIL	ACCIONES
1	1805120910	JONATHAN DANIEL NUÑEZ GUERRERO	AMBATO	0987120698	MASCULINO	SOLTERO	VER EDITAR ELIMINAR FICHA MÉDICA
2	1804752010	JUAN ANDRES PALACIOS PEREZ	PELILEO	0987145263	MASCULINO	SOLTERO	VER EDITAR ELIMINAR FICHA MÉDICA
3	1020528520	KAREN GEOMARA QUITO PAZMIÑO	QUERO	0985214754	FEMENINO	CASADO	VER EDITAR ELIMINAR FICHA MÉDICA
4	1258963014	MARIA ELENA CATILLO GUERRERO	CEVALLOS	0952147852	FEMENINO	DIVORCIADA	VER EDITAR ELIMINAR FICHA MÉDICA

Figura 3.2.15: Docentes o Estudiantes Registrados.

Elaborado por: Investigador.

unidadeducativagonzalezsuarez.es/usuarios/create

U.E. González Suárez

ADMINISTRADOR

REGISTRAR DOCENTE

CÉDULA

NOMBRE COMPLETO

EDAD

FECHA NACIMIENTO

Figura 3.2.16: Registro de docentes o estudiantes.

Elaborador por: Investigador.

UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ
FICHA MÉDICA DOCENTE

A. DATOS PERSONALES DEL DOCENTE				
INSTITUCIÓN	CÉDULA	NOMBRES Y APELLIDOS	GÉNERO	CARGO/OCUPACIÓN
UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ	1805120910	JONATHAN DANIEL NUÑEZ GUERRERO	MASCULINO	DOCENTE
B. DATOS ADICIONALES DEL DOCENTE				
FECHA DE NACIMIENTO	EDAD	DIRECCIÓN	RELIGIÓN	
16/05/1998	23	AMBATO	CATÓLICO	
TIPO DE SANGRE	LATERALIDAD	ORIENTACION SEXUAL	IDENTIDAD DE GÉNERO	
A+	IZQUIERDO/A	HETEROSEXUAL	HOMBRE	
C. ANTECEDENTES PATOLÓGICOS				
ANTECEDENTES PATOLÓGICOS PERSONALES				
PATOLOGÍA	SÍ/NO	DESCRIPCIÓN		
DIABETES	NO	NINGUNA		
LUMBALGIA	NO	NINGUNA		
ENF. RESPIRATORIAS	NO	NINGUNA		
ENF. OSTEOMUSCULARES	NO	NINGUNA		
HIPOCUSIA	NO	NINGUNA		
CIRUGÍA	NO	NINGUNA		
DISCAPACIDAD	NINGUNA	NINGUNA		
ALERGIA	NO	NINGUNA		

Figura 3.2.17: Formato ficha médica docentes o estudiantes.

Elaborador por: Investigador.

Para la gestión de las tablas administrativas tendrán el mismo formato de las Figuras 3.2.15 – 3.2.16 a estas tablas solo tendrá acceso el administrador.

unidadeducativagonzalezsuarez.es/profesor

ADMINISTRADOR

GÉNERO

AGREGAR GÉNERO

Search...

N°	GÉNERO	ACCIONES
1	MACULINO	VER EDITAR ELIMINAR
2	FEMENINO	VER EDITAR ELIMINAR

Figura 3.2.18: Listado de tipos registrados en las tablas administrativas.

Elaborador por: Investigador.

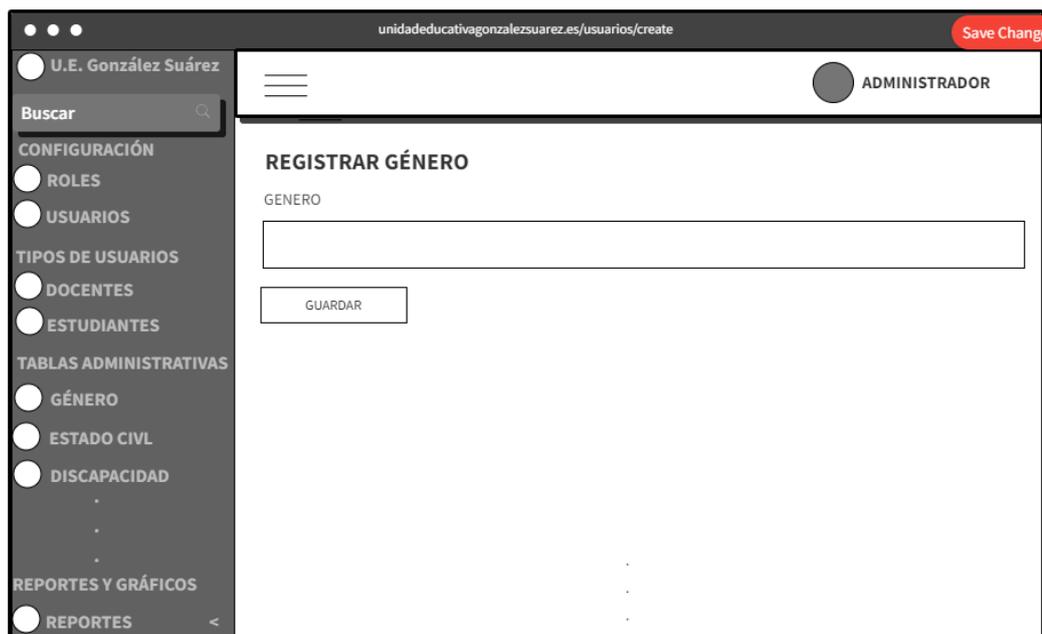


Figura 3.2.19: Registrar el tipo en la tabla administrativa.

Elaborado por: Investigador.

Para generar reportes se tendrá el mismo formato de las Figuras 3.2.18 - 3.2.19 para todos los reportes. Para poder generar un reporte se deberá tener el rol de administrador o departamento médico.



Figura 3.2.20: Filtrado de datos para los reportes.

Elaborado por: Investigador.

unidadeducativagonzalezsuarez.es/estudiante/tiposangregrado

REPORTE DE ESTUDIANTES POR CURSO/GRADO Y POR GÉNERO 1 / 1 100%

UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ
REPORTE DE ESTUDIANTES

INICIAL 1 - A

N°	NOMBRES/APELLIDOS DEL ESTUDIANTE	TIPO SANGRE
1	ANTHONY ALEJANDRO NUÑEZ GUERRERO	A+
2	CARLOS IVAN GUERRERO NUÑEZ	A+

Figura 3.2.21: Reporte del estudiante por tipo de sangre y grado o curso.

Elaborador por: Investigador.

3.2.3. Fase 3: Construcción

Estructura de Laravel

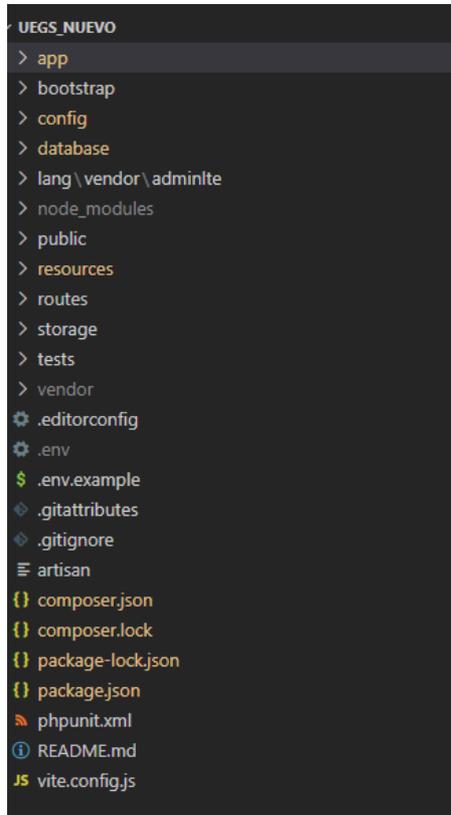


Figura 3.2.22: Estructura de Laravel.

Elaborador por: Investigador.

Conexión Base de datos

Para tener acceso a la base de datos MySQL de XAMPP se debe configurar el archivo .env, en el cual se debe registrar el nombre de base de datos, usuario y contraseña.

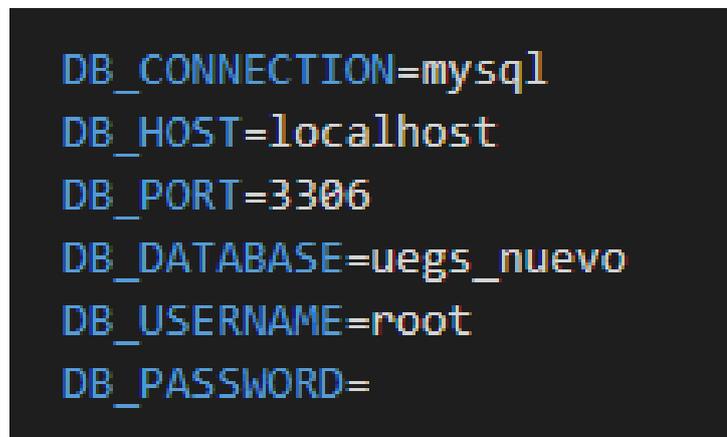


Figura 3.2.23: Conexión a la base de datos MySQL de XAMPP.

Elaborado por: Investigador.

Migraciones en Laravel

Sirve para crear tablas en la base de datos permitiendo definir y versionar los cambios en el código fuente del sistema.

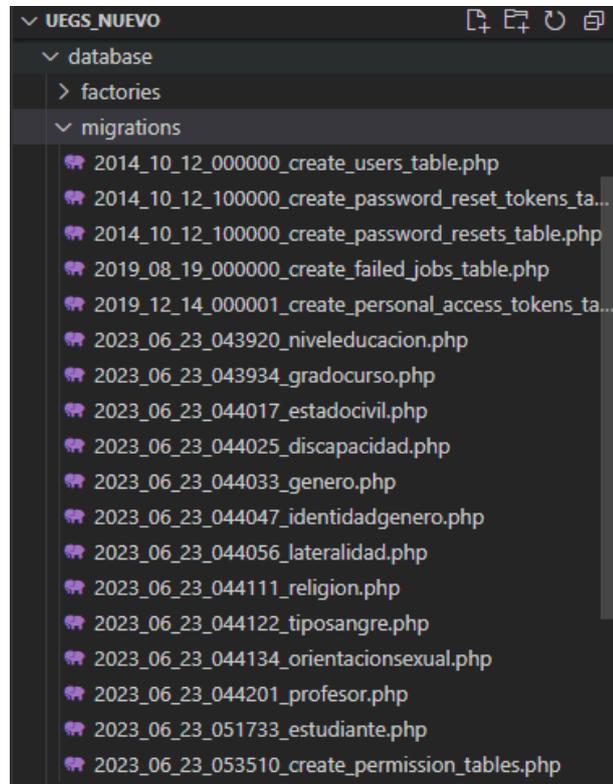


Figura 3.2.24: Migraciones del Sistema.

Elaborado por: Investigador.

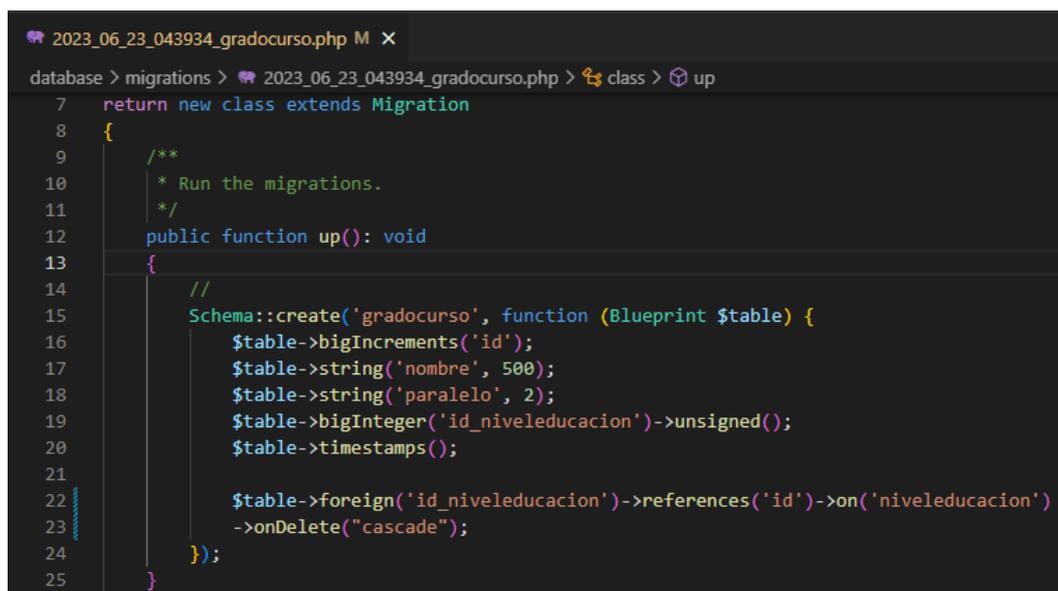


Figura 3.2.25: Creación de Migración de la tabla gradocurso con una relación a la tabla niveleduación.

Elaborado por: Investigador.

Instalación AdminLTE y Laravel Auth

AdminLTE ayudara a crear interfaces de usuario elegantes y receptivas basadas en Bootstrap, mientras que Laravel Auth se encargara de la autenticación de usuarios que ingresen al sistema, además brinda servicios como el inicio de sesión, la recuperación de contraseñas, el registro de usuarios, entre otros.

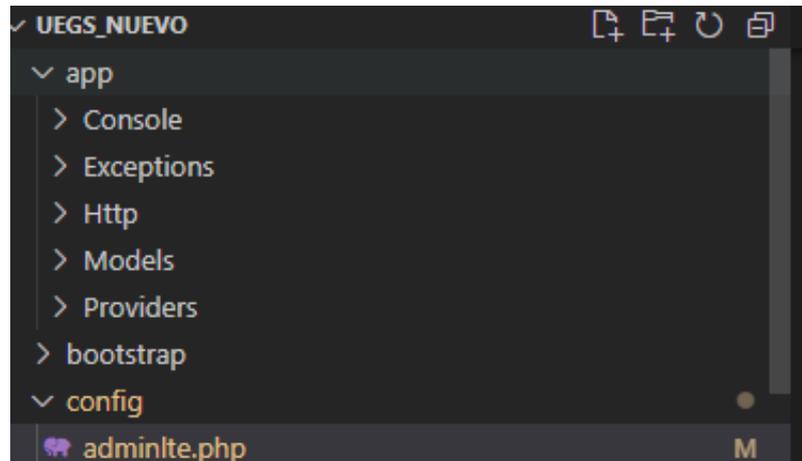


Figura 3.2.26: Archivo de Configuración AdminLTE.

Elaborador por: Investigador.

```
@extends('adminlte::page')

@section('title', 'Dashboard')

@section('content_header')
    <h1>Dashboard</h1>
@stop

@section('content')
    <p>Welcome to this beautiful admin panel.</p>
@stop

@section('css')
    <link rel="stylesheet" href="/css/admin_custom.css">
@stop

@section('js')
    <script> console.log('Hi!'); </script>
@stop
```

Figura 3.2.27: Plantilla general para la creación de archivos nombre_archivo.blade.php.

Elaborador por: Investigador

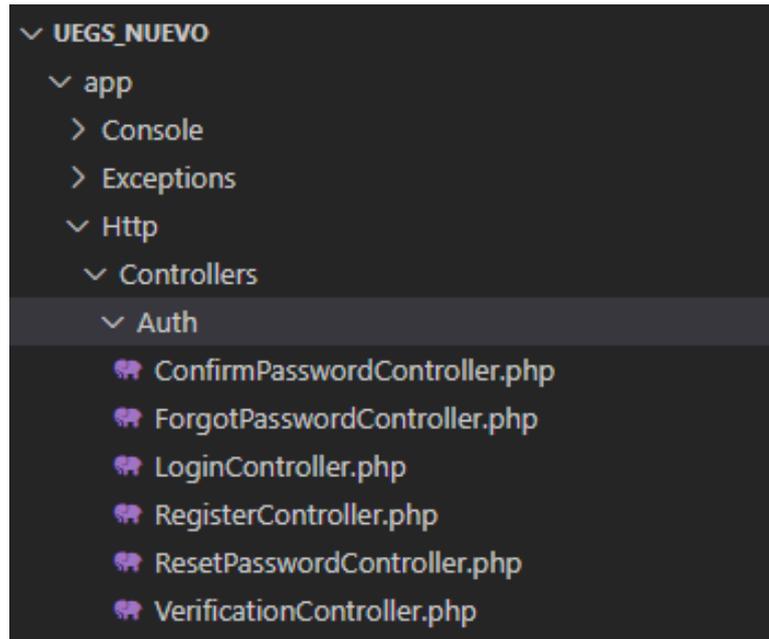


Figura 3.2.28: Controladores definidos por Laravel Auth.

Elaborador por: Investigador.

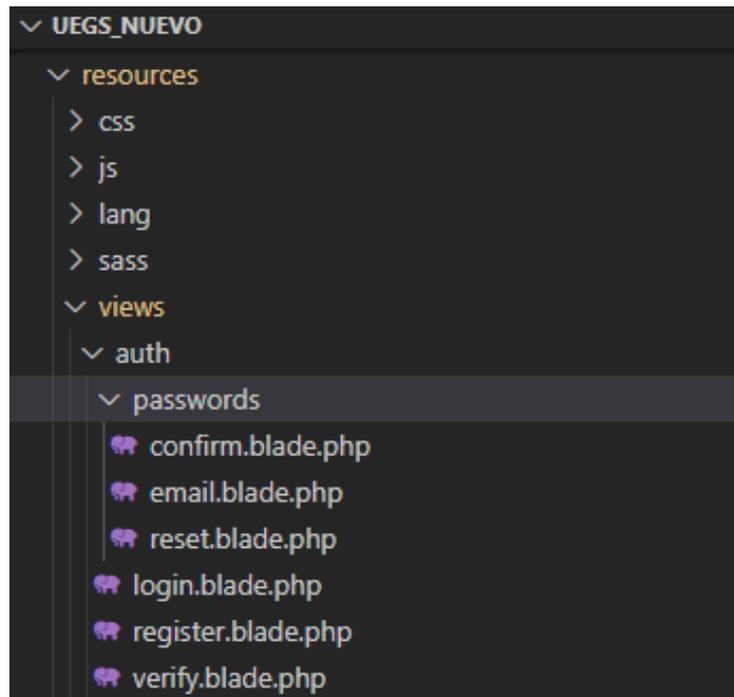


Figura 3.2.29: Vistas definidas por Laravel Auth.

Elaborado por: Investigador.

Instalación Laravel Permission

Se implementa para dar funcionalidades de control de acceso al sistema como asignación de roles y permisos a los usuarios y la protección de rutas.

```
app.php X
config > app.php
157
158     'providers' => ServiceProvider::defaultProviders()->merge([
159         /*
160          * Package Service Providers...
161          */
162
163         /*
164          * Application Service Providers...
165          */
166         App\Providers\AppServiceProvider::class,
167         App\Providers\AuthServiceProvider::class,
168         // App\Providers\BroadcastServiceProvider::class,
169         App\Providers\EventServiceProvider::class,
170         App\Providers\RouteServiceProvider::class,
171
172         //JeroenNoten\LaravelAdminLte\ServiceProviders::class,
173         // ...
174         Spatie\Permission\PermissionServiceProvider::class,
175     ])->toArray(),
```

Figura 3.2.30: Asignación de proveedor de servicios Spatie.

Elaborado por: Investigador.

```
Kernel.php X
app > Http > Kernel.php > ...
55     protected $middlewareAliases = [
56         'auth' => \App\Http\Middleware\Authenticate::class,
57         'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
58         'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
59         'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
60         'can' => \Illuminate\Auth\Middleware\Authorize::class,
61         'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
62         'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
63         'precognitive' => \Illuminate\Foundation\Http\Middleware\HandlePrecognitiveRequests::class,
64         'signed' => \App\Http\Middleware\ValidateSignature::class,
65         'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
66         'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
67
68         // ...
69         'role' => \Spatie\Permission\Middlewares\RoleMiddleware::class,
70         'permission' => \Spatie\Permission\Middlewares\PermissionMiddleware::class,
71         'role_or_permission' => \Spatie\Permission\Middlewares\RoleOrPermissionMiddleware::class,
72     ];
73 }
```

Figura 3.2.31: Asignación de Middleware Aliases para Spatie.

Elaborador por: Investigador.

```
User.php X
app > Models > User.php > User > adminlte_image
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 Illuminate\Support\Facades\Auth;
10 use Illuminate\Testing\Fluent\Concerns\Has;
11 use Laravel\Sanctum\HasApiTokens;
12 use Spatie\Permission\Traits\HasRoles;
13
14
15 class User extends Authenticatable
16 {
17     use HasApiTokens, HasFactory, Notifiable, HasRoles ;
18 }
```

Figura 3.2.32: Asignación Has Roles al Modelo User.php.

Elaborador por: Investigador.

```
RegisterController.php X
app > Http > Controllers > Auth > RegisterController.php > ...
66     protected function create(array $data)
67     {
68         $roles = Role::all();
69         $user = User::create([
70             'name' => $data['name'],
71             'email' => $data['email'],
72             'password' => Hash::make($data['password']),
73         ]);
74         $user->assignRole($data['role']);
75         return $user;
76     }
77 }
78 }
```

Figura 3.2.33: Método para asignar un rol al registrar un usuario.

Elaborado por: Investigador.

```

SeederTablaPermisos.php X
database > seeders > SeederTablaPermisos.php > SeederTablaPermisos > run
1  <?php
2
3  namespace Database\Seeders;
4
5  use Illuminate\Database\Console\Seeds\WithoutModelEvents;
6  use Illuminate\Database\Seeder;
7
8  use Spatie\Permission\Models\Permission;
9  class SeederTablaPermisos extends Seeder
10 {
11     /**
12      * Run the database seeds.
13      */
14     public function run(): void
15     {
16         $permisos = [
17             'dashboard-administrador',
18             'dashboard-docentes',
19             'pdf-edades'
20         ];
21         foreach($permisos as $permiso){
22             Permission::create(['name'=>$permiso]);
23         }
24     }
25 }
26

```

Figura 3.2.34: Ejemplo Seeder para la creación de permisos.

Elaborador por: Investigador.

```

register.blade.php M X
resources > views > vendor > adminlte > auth > register.blade.php > form
92  {{{!-- Asignación de Usuario y Roles --}}}
93  <div class="input-group mb-3">
94
95      <select class="form-control" name="role" aria-placeholder="Seleccionar Rol">
96          <option value="#" hidden>SELECCIONAR TIPO DE USUARIO</option>
97          <option value="3">DOCENTE</option>
98          <option value="4">ESTUDIANTE</option>
99      </select>
100     @if ($errors->has('role'))
101         <span class="help-block">
102             <strong>{{ $errors->first('role') }}</strong>
103         </span>
104     @endif
105
106     <div class="input-group-append">
107         <div class="input-group-text">
108             <span class="fas fa-user-lock {{ config('adminlte.classes_auth_icon', '') }}">
109         </div>
110     </div>
111
112     @error('name')
113         <span class="invalid-feedback" role="alert">
114             <strong>{{ $message }}</strong>
115         </span>
116     @enderror
117 </div>
118 <div class="form-group{{ $errors->has('role') ? ' has-error' : '' }}">
119

```

Figura 3.2.35: Edición vista register.blade.php para asignar un rol al usuario.

Elaborado por: Investigador.

Gestión Roles

Para la creación de roles Laravel Permission relaciona al rol a cada permiso que este puede tener.

```
RolController.php X
app > Http > Controllers > RolController.php > RolController > store
44 | /
45 | public function store(Request $request)
46 | {
47 |     //
48 |     $this->validate($request, ['name' => 'required', 'permission'=> 'required']);
49 |     $role = Role::create(['name' => $request->input('name')]);
50 |     $role->syncPermissions($request->input('permission'));
51 |
52 |     return redirect()->route('roles.index');
53 | }
54 |
```

Figura 3.2.36: Controlador Rol y función store para la creación de Rol y sus permisos.

Elaborado por: Investigador.

Gestión Usuarios

Para la creación de usuarios Laravel Permission relaciona un usuario al rol al que pertenece permitiendo tener un buen control al rato que un usuario sea registrado o creado por el administrador con un rol en específico.

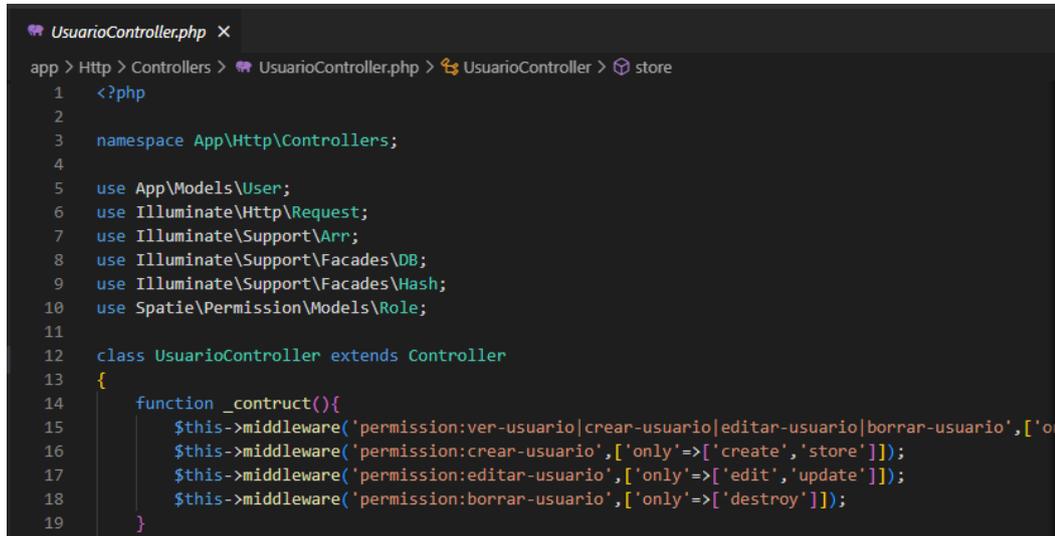
```
UsuarioController.php X
app > Http > Controllers > UsuarioController.php > UsuarioController > store
42 | */
43 | public function store(Request $request)
44 | {
45 |     //
46 |     $this->validate($request, [
47 |         'name' => 'required',
48 |         'email' => 'required|unique:users,email',
49 |         'password' => 'required|same:confirm-password',
50 |         'roles' => 'required'
51 |     ]);
52 |
53 |     $input = $request->all();
54 |     $input['password'] = Hash::make($input['password']);
55 |
56 |     $user = User::create($input);
57 |     $user->assignRole($request->input('roles'));
58 |
59 |     return redirect()->route('usuarios.index');
60 | }
61 |
```

Figura 3.2.37: Controlador Usuario y función store para la creación de usuarios con su rol.

Elaborado por: Investigador.

Middleware de Laravel Permission

Se lo puede utilizar para restringir el acceso a ciertas acciones del controlador, estas acciones son las que están asignadas a los permisos que tiene los usuarios cuando son asignados a un rol, estas acciones son las que están creadas en el seeder ver **Figura 3.2.33**.



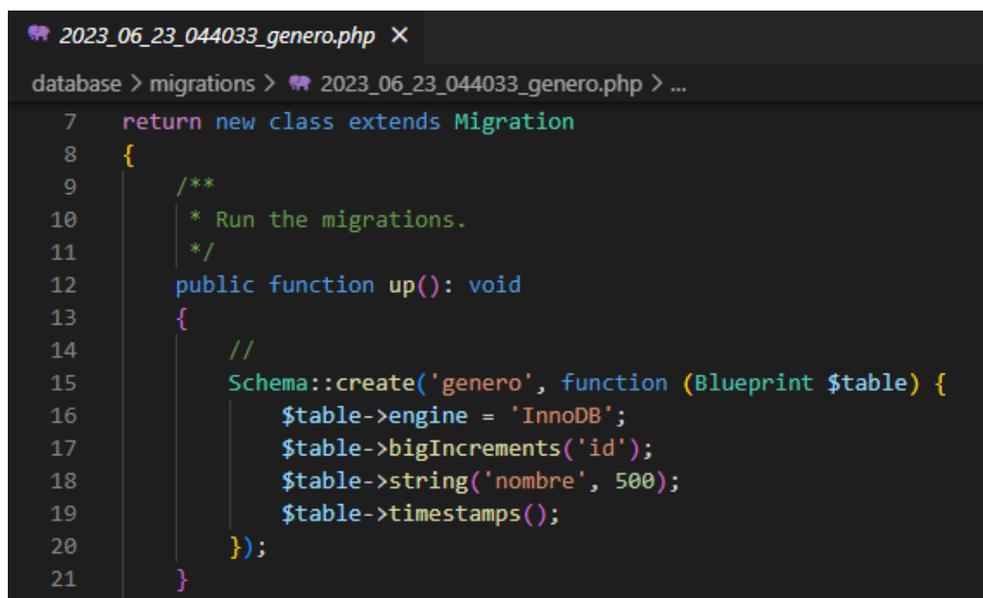
```
UsuarioController.php X
app > Http > Controllers > UsuarioController.php > UsuarioController > store
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\User;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Arr;
8  use Illuminate\Support\Facades\DB;
9  use Illuminate\Support\Facades\Hash;
10 use Spatie\Permission\Models\Role;
11
12 class UsuarioController extends Controller
13 {
14     function __construct(){
15         $this->middleware('permission:ver-usuario|crear-usuario|editar-usuario|borrar-usuario',['o
16         $this->middleware('permission:crear-usuario',['only'=>['create','store']]);
17         $this->middleware('permission:editar-usuario',['only'=>['edit','update']]);
18         $this->middleware('permission:borrar-usuario',['only'=>['destroy']]);
19     }
}
```

Figura 3.2.38: Middleware en controlador usuario apuntando a una función.

Elaborado por: Investigador.

Gestión Tablas Administrativas

Sirve agregar la información de los tipos que pueden existir en cada una de las tablas administrativas.



```
2023_06_23_044033_genero.php X
database > migrations > 2023_06_23_044033_genero.php > ...
7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     */
12     public function up(): void
13     {
14         //
15         Schema::create('genero', function (Blueprint $table) {
16             $table->engine = 'InnoDB';
17             $table->bigIncrements('id');
18             $table->string('nombre', 500);
19             $table->timestamps();
20         });
21     }
}
```

Figura 3.2.39: Migración de la tabla administrativa género.

Elaborador por: Investigador.

```

Genero.php X
app > Models > Genero.php > Genero
20 class Genero extends Model
21 {
22     protected $table = "genero";
23     static $rules = [
24         'nombre' => 'required',
25     ];
26
27     protected $perPage = 20;
28
29     /**
30      * Attributes that should be mass-assignable.
31      *
32      * @var array
33      */
34     protected $fillable = ['nombre'];
35
36
37     /**
38      * @return \Illuminate\Database\Eloquent\Relations\HasMany
39      */
40     public function estudiantes()
41     {
42         return $this->hasMany('App\Models\Estudiante', 'id_genero', 'id');
43     }
44 }

```

Figura 3.2.40: Modelo tabla Administrativa Género.

Elaborado por: Investigador.

```

GeneroController.php X
app > Http > Controllers > GeneroController.php > ...
12 class GeneroController extends Controller
13 {
14     function __construct(){
15         $this->middleware('permission:ver-genero|crear-genero|editar-genero|borrar-genero',['only'
16         $this->middleware('permission:crear-genero',['only'=>['create','store']]);
17         $this->middleware('permission:editar-genero',['only'=>['edit','update']]);
18         $this->middleware('permission:borrar-genero',['only'=>['destroy']]);
19     }

```

Figura 3.2.41: Controlador Género y constructor con permisos.

Elaborado por: Investigador.

```

GeneroController.php X
app > Http > Controllers > GeneroController.php > ...
25 public function index()
26 {
27     $generos = Genero::paginate();
28
29     return view('genero.index', compact('generos'))
30         ->with('i', (request()->input('page', 1) - 1) * $generos->perPage());
31 }
32
33 /**
34  * Show the form for creating a new resource.
35  *
36  * @return \Illuminate\Http\Response
37  */
38 public function create()
39 {
40     $genero = new Genero();
41     return view('genero.create', compact('genero'));
42 }

```

Figura 3.2.42: Función index y créate del controlador Género.

Elaborado por: Investigador.

```
GeneroController.php X
app > Http > Controllers > GeneroController.php > ...
50 public function store(Request $request)
51 {
52     request()->validate(Genero::$rules);
53
54     $genero = Genero::create($request->all());
55
56     return redirect()->route('genero.index')
57         ->with('success', 'Genero created successfully.');
```

```
58 }
59
60 /**
61  * Display the specified resource.
62  *
63  * @param int $id
64  * @return \Illuminate\Http\Response
65  */
66 public function show($id)
67 {
68     $genero = Genero::find($id);
69
70     return view('genero.show', compact('genero'));
71 }
```

Figura 3.2.43: Función store y show del controlador Género.

Elaborador por: Investigador.

```
GeneroController.php X
app > Http > Controllers > GeneroController.php > GeneroController > update
79 public function edit($id)
80 {
81     $genero = Genero::find($id);
82
83     return view('genero.edit', compact('genero'));
84 }
85
86 /**
87  * Update the specified resource in storage.
88  *
89  * @param \Illuminate\Http\Request $request
90  * @param Genero $genero
91  * @return \Illuminate\Http\Response
92  */
93 public function update(Request $request, Genero $genero)
94 {
95     request()->validate(Genero::$rules);
96
97     $genero->update($request->all());
98
99     return redirect()->route('genero.index')
100         ->with('success', 'Genero updated successfully');
```

```
101 }
```

Figura 3.2.44: Función edit y update del controlador Género.

Elaborado por: Investigador.

```

GeneroController.php X
app > Http > Controllers > GeneroController.php > GeneroController > index
108     public function destroy($id)
109     {
110         $genero = Genero::find($id)->delete();
111
112         return redirect()->route('genero.index')
113             ->with('success', 'Genero deleted successfully');
114     }
115 }

```

Figura 3.2.45: Función destroy del controlador Género.

Elaborado por: Investigador.

La migración, el modelo y las funciones de la tabla administrativa genero indicadas en la **Figura 3.2.38** hasta la **Figura 3.2.44** son una guía para la creación y gestión de todas las tablas administrativas necesarias del sistema.

Gestión Docentes y Estudiantes

Sirve para agregar la información la cual ayudara a crear la ficha médica de los docentes y estudiantes.

```

database > migrations > 2023_06_23_043934_gradocurso.php > class > up
6
7     return new class extends Migration
8     {
9         /**
10          * Run the migrations.
11          */
12         public function up(): void
13         {
14             //
15             Schema::create('gradocurso', function (Blueprint $table) {
16                 $table->bigIncrements('id');
17                 $table->string('nombre', 500);
18                 $table->string('paralelo', 2);
19                 $table->bigInteger('id_niveleduccion')->unsigned();
20                 $table->timestamps();
21
22                 $table->foreign('id_niveleduccion')->references('id')->on('niveleduccion')
23                 ->onDelete("cascade");
24             });
25         }
26     }

```

Figura 3.2.46: Migración de la tabla gradocurso con su relación.

Elaborado por: Investigador.

La migración de la **Figura 3.2.45** es una guía la cual se puede usar para crear el esquema y migración tanto de la tabla estudiante como docente.

```

Estudiante.php X
app > Models > Estudiante.php > ...
74 class Estudiante extends Model
75 {
76     protected $table='estudiante';
77     static $rules = [ ...
111 ];
112
113     protected $perPage = 20;
114
115     /** ...
120     protected $fillable = ['cedula','nombrecompletoestudiante','nombrecompletopadre','nombrecompletomadre','edad','fechanacimient
121
122
123     /** ...
126     public function discapacidad()
127     {
128         return $this->hasOne('App\Models\Discapacidad', 'id', 'id_discapacidad');
129     }
130
131     /** ...
134     public function genero()
135     {
136         return $this->hasOne('App\Models\Genero', 'id', 'id_genero');
137     }
138
139     /** ...
142     public function gradocurso()
143     {
144         return $this->hasOne('App\Models\Gradocurso', 'id', 'id_gradocurso');
145     }
146
147     /** ...
150     public function identidadgenero()
151     {
152         return $this->hasOne('App\Models\Identidadgenero', 'id', 'id_identidadgenero');
153     }

```

Figura 3.2.47: Modelo de estudiante con las funciones de las tablas administrativas que están relacionadas al modelo.

Elaborador por: Investigador.

```

EstudianteController.php M X
app > Http > Controllers > EstudianteController.php > EstudianteController > update
26 //
27 class EstudianteController extends Controller
28 {
29     function __construct(){ ...
40 }
41     /** ...
46     public function index()
47     { ...
52 }
53     /** ...
58     public function create()
59     { ...
83 }
84     /** ...
90     public function store(Request $request)
91     { ...
98 }
99     /** ...
105     public function show($id)
106     { ...
113 }
114     /** ...
120     public function edit($id)
121     { ...
148 }
149     /** ...
156     public function update(Request $request, Estudiante $estudiante)
157     { ...
164 }
165     /** ...
170     public function destroy($id)
171     { ...
178 }
179     public function pdf($id){ ...
187 }
188 }

```

Figura 3.2.48: Funciones del controlador estudiante para el crud y para la generación de ficha médica.

Elaborador por: Investigador.

```

EstudianteController.php M X
app > Http > Controllers > EstudianteController.php > EstudianteController
179 public function pdf($id){
180
181     $estudiante = Estudiante::find($id);
182     $fechaformato = $estudiante->fechanacimiento;
183     $fecharest = Carbon::createFromFormat('Y-m-d H:i:s', $fechaformato);
184     $fecha = $fecharest->format('d/m/Y');
185     $pdf = Pdf::loadView('estudiante.pdf', compact('estudiante','fecha'));
186     return $pdf->stream();
187 }

```

Figura 3.2.49: Función pdf para obtener los datos de estudiante y generar la ficha médica.

Elaborador por: Investigador.

Las migraciones, el modelo y funciones del estudiante mostradas en la **Figura 3.2.45** hasta la **Figura 3.2.48** son una guía para crear y gestionar al docente.

Gestión Reportes

Se encarga en la obtención de información mediante filtros para obtener reportes por género, grado o curso y por edades.

```

EstudianteController.php M X
app > Http > Controllers > EstudianteController.php > EstudianteController
189 public function tiposangrea(Request $request){
190
191     // Obtener los parámetros de filtro del request
192     //return view('estudiante.tiposangrea', compact('estudianteamas','tiposangreamas',
193     $sangres = Tiposangre::pluck('nombre', 'id');
194     $tipoSangre = $request->input('tipo_sangre');
195
196     // Realizar la consulta con el filtro aplicado
197     $estudiantes = Estudiante::where('id_tiposangre', $tipoSangre)->get();
198
199     // Retornar la vista que muestra los resultados del filtro
200     return view('estudiante.tiposangrea', compact('estudiantes','sangres'));
201 }
202
203 public function pdftiposangre(Request $request)
204 {
205     $tipoSangre = $request->input('tipo_sangre');
206     $estudiantes = Estudiante::where('id_tiposangre', $tipoSangre)->get();
207     $pdf = Pdf::loadView('estudiante.pdftiposangre', compact('estudiantes'));
208     return $pdf->stream();
209 }

```

Figura 3.2.50: Función tiposangrea y pdftiposangre para generar reportes por tipo de sangre de los estudiantes de la institución.

Elaborador por: Investigador.

```

EstudianteController.php M X
app > Http > Controllers > EstudianteController.php > EstudianteController > tiposangregrado
210
211 public function tiposangregrado(Request $request){
212     $sangres = Tiposangre::pluck('nombre', 'id');
213     $registros = Gradocurso::all();
214     $grados = [];
215     foreach ($registros as $registro) {
216         $grados[$registro->id] = $registro->nombre . ' - ' . $registro->paralelo;
217     }
218     $sangre = $request->input('sangre');
219     $grado = $request->input('grado');
220     // Obtener los estudiantes filtrados según los valores seleccionados
221     $estudiantes = Estudiante::where('id_tiposangre', $sangre)->where('id_gradocurso', $grado)->where('estado', '1')->get();
222     $i=0;
223     // Retornar la vista que muestra los resultados del filtro
224     return view('estudiante.tiposangregrado', compact('estudiantes','sangres','grados','i'));
225 }
226
227 public function pdftiposangregrado(Request $request)
228 {
229     $sangres = Tiposangre::pluck('nombre', 'id');
230     $grados = Gradocurso::pluck('nombre', 'id');
231     $sangre = $request->input('sangre');
232     $grado = $request->input('grado');
233     // Obtener los estudiantes filtrados según los valores seleccionados
234     $estudiantes = Estudiante::where('id_tiposangre', $sangre)->where('id_gradocurso', $grado)->where('estado', '1')->get();
235     $i=0;
236     // Generar el PDF con los datos filtrados
237     $pdf = PDF::loadView('estudiante.pdftiposangregrado', compact('estudiantes','grados','i'));
238     // Retornar el PDF como respuesta para descargar o visualizar
239     return $pdf->stream();
240 }
241

```

Figura 3.2.51: Función tiposangregrado y pdftiposangregrado para generar reporte del tipo de sangre de los estudiantes por cursos o grados.

Elaborado por: Investigador.

```

EstudianteController.php M X
app > Http > Controllers > EstudianteController.php > EstudianteController > pdfgenero
242 public function genero(Request $request){
243     $generos = Genero::pluck('nombre', 'id');
244     $genero = $request->input('genero');
245
246     // Realizar la consulta con el filtro aplicado
247     $estudiantes = Estudiante::where('id_genero', $genero)->where('estado', '1')->get();
248
249     // Retornar la vista que muestra los resultados del filtro
250     return view('estudiante.genero', compact('estudiantes','generos'));
251 }
252
253 public function pdfgenero(Request $request)
254 {
255     $genero = $request->input('genero');
256     // Realizar la consulta nuevamente con el filtro aplicado
257     $estudiantes = Estudiante::where('id_genero', $genero)->where('estado', '1')->get();
258     // Generar el PDF con los datos filtrados
259     $pdf = PDF::loadView('estudiante.pdfgenero', compact('estudiantes'));
260
261     // Retornar el PDF como respuesta para descargar o visualizar
262     return $pdf->stream();
263 }
264

```

Figura 3.2.52: Función genero y pdfgenero para generar reporte por genero de los estudiantes de la institución.

Elaborado por: Investigador.

```

EstudianteController.php M X
app > Http > Controllers > EstudianteController.php > EstudianteController > pdfgenerogrado
265 public function generogrado(Request $request){
266     $generos = Genero::pluck('nombre', 'id');
267     $registros = Gradocurso::all();
268     $grados = [];
269     foreach ($registros as $registro) {
270         $grados[$registro->id] = $registro->nombre . ' - ' . $registro->paralelo;
271     }
272     $genero = $request->input('genero');
273     $grado = $request->input('grado');
274     // Obtener los estudiantes filtrados según los valores seleccionados
275     $estudiantes = Estudiante::where('id_genero', $genero)->where('id_gradocurso', $grado)->where('estado', '1')->get();
276     $i=0;
277     // Retornar la vista que muestra los resultados del filtro
278     return view('estudiante.generogrado', compact('estudiantes','generos','grados','i'));
279 }
280
281 public function pdfgenerogrado(Request $request)
282 {
283     $generos = Genero::pluck('nombre', 'id');
284     $grados = Gradocurso::pluck('nombre', 'id');
285     $genero = $request->input('genero');
286     $grado = $request->input('grado');
287     // Obtener los estudiantes filtrados según los valores seleccionados
288     $estudiantes = Estudiante::where('id_genero', $genero)->where('id_gradocurso', $grado)->where('estado', '1')->get();
289     $i=0;
290     // Generar el PDF con los datos filtrados
291     $pdf = PDF::loadView('estudiante.pdfgenerogrado', compact('estudiantes','grados','i'));
292     // Retornar el PDF como respuesta para descargar o visualizar
293     return $pdf->stream();
294 }

```

Figura 3.2.53: Función generogrado y pdfgenerogrado para generar reporte por genero y grado del estudiante.

```

EstudianteController.php M X
app > Http > Controllers > EstudianteController.php > EstudianteController > pdfedadcurso
296 public function edadcurso(Request $request){
297     $registros = Gradocurso::all();
298     $grados = [];
299
300     foreach ($registros as $registro) {
301         $grados[$registro->id] = $registro->nombre . ' - ' . $registro->paralelo;
302     }
303     $grado = $request->input('grado');
304
305     // Obtener los estudiantes filtrados según los valores seleccionados
306     $estudiantes = Estudiante::where('id_gradocurso', $grado)->where('estado', '1')->get();
307
308     $i=0;
309     // Retornar la vista que muestra los resultados del filtro
310     return view('estudiante.edadcurso', compact('estudiantes','grados','i'));
311 }
312
313 public function pdfedadcurso(Request $request)
314 {
315     $grados = Gradocurso::pluck('nombre', 'id');
316     $grado = $request->input('grado');
317     // Obtener los estudiantes filtrados según los valores seleccionados
318     $estudiantes = Estudiante::where('id_gradocurso', $grado)->where('estado', '1')->get();
319     $i=0;
320     // Generar el PDF con los datos filtrados
321     $pdf = PDF::loadView('estudiante.pdfedadcurso', compact('estudiantes','grados','i'));
322
323     // Retornar el PDF como respuesta para descargar o visualizar
324     return $pdf->stream();
325 }

```

Figura 3.2.54: Función edadcurso y pdfedadcurso para genera reporte de estudiantes por edad y curso o grado al que pertenece.

Elaborado por: Investigador.

3.2.4. Fase 4: Transición

A. Pruebas del Sistema

Pruebas Unitarias

El objetivo de las pruebas unitarias consiste en validar el funcionamiento correcto de los flujos que se encuentran presente en el sistema; es decir, que están dirigidas a componentes menores como lo son los módulos de un sistema.

A continuación, se muestran las siguientes pruebas unitarias a realizar en los casos de uso del proyecto:

Caso de Uso Inicio de Sesión

Código	Objetivo de la Prueba
PU-001	Verificar en el sistema las validaciones cuando un usuario ingresa credenciales incorrectas o no registradas.

Tabla 3.2.10: Prueba Unitaria Inicio de Sesión.

Elaborado por: Investigador.

Caso de Uso Gestión Roles

Código	Objetivo de la Prueba
PU-002	Verificar que se pueda agregar roles con sus respectivos permisos al sistema.
PU-003	Verificar que se pueda editar un rol para agregar o quitar permisos.
PU-004	Verificar que se pueda eliminar un rol con sus permisos en casos de ser necesario.
PU-005	Verificar que no se pueda eliminar un rol si esta referenciado a un usuario.

Tabla 3.2.11: Prueba Unitaria Gestión Roles.

Elaborado por: Investigador.

Caso de Uso Gestión Usuarios

Código	Objetivo de la Prueba
PU-006	Verificar que se pueda agregar usuarios con un rol.
PU-007	Verificar que se pueda editar un usuario.
PU-008	Verificar que no se pueda agregar un usuario con un registro ya existente.
PU-009	Verificar en el sistema la validación en el campo email si no se ingresa un correo.
PU-010	Verificar en el sistema la validación del campo contraseña si no es mayor a 8 caracteres.

Tabla 3.2.12: Prueba Unitaria Gestión Usuarios.

Elaborado por: Investigador.

Caso de Uso Tablas Administrativas

Código	Objetivo de la Prueba
PU-011	Verificar que se pueda agregar los tipos en las tablas administrativas.
PU-012	Verificar que se pueda editar los tipos en las tablas administrativas.
PU-013	Verificar que se pueda eliminar un tipo en las tablas administrativas.
PU-014	Verificar en el sistema la validación de los campos necesarios.

Tabla 3.2.13: Pruebas Unitarias Tablas Administrativas.

Elaborado por: Investigador.

Caso de Uso Gestión Docentes

Código	Objetivo de la Prueba
PU-015	Verificar que se pueda agregar la información de los docentes.
PU-016	Verificar que se pueda editar la información de los docentes.
PU-017	Verificar que se pueda eliminar o desactivar un docente que ya no pertenezca a la institución.
PU-018	Verificar en el sistema la validación de los campos necesarios de la información del docente.
PU-019	Verificar la creación de ficha médica.
PU-020	Verificar en la ficha médica que los campos estén correctamente ubicados.

Tabla 3.2.14: Prueba Unitaria Gestión Docentes.

Elaborado por: Investigador.

Caso de Uso Gestión Estudiantes

Código	Objetivo de la Prueba
PU-021	Verificar que se pueda agregar la información de los estudiantes.
PU-022	Verificar que se pueda editar la información de los estudiantes.
PU-023	Verificar que se pueda eliminar o desactivar un estudiante que ya no pertenezca a la institución.
PU-024	Verificar en el sistema la validación de los campos necesarios de la información del estudiante.
PU-025	Verificar la creación de ficha médica.
PU-026	Verificar en la ficha médica que los campos estén correctamente ubicados.

Tabla 3.2.15: Prueba Unitaria Gestión Estudiantes.

Elaborado por: Investigador.

Caso de Uso Gestión Reporte

Código	Objetivo de la Prueba
PU-027	Verificar que se pueda filtrar la información requerida.
PU-028	Verificar que se muestre correctamente la información filtrada.
PU-029	Verificar que genere reporte de la información filtrada.
PU-030	Verificar en el sistema la validación de los campos necesarios para filtrar la información.

PU-031	Verificar que los campos de la información filtrada estén correctamente ubicados.
--------	---

Tabla 3.2.16: Prueba Unitaria Gestión Reporte.

Elaborado por: Investigador.

Caso de Uso Cerrar Sesión

Código	Objetivo de la Prueba
PU-032	Verificar en el sistema que el usuario finalice la sesión.

Tabla 3.2.17: Prueba Unitaria Cerrar Sesión.

Elaborado por: Investigador.

B. Técnica Utilizada

Pruebas de Caja Negra

Las pruebas de caja negra son realizadas en base a entradas que recibe y las salidas que produce sin tener en cuenta el funcionamiento interno, es decir, están enfocados en los requerimientos establecidos y en la funcionalidad del sistema.

Código	Descripción	Resultado Obtenido
PU-001	Ingresar las credenciales de inicio de sesión.	Se muestra un aviso que los datos son incorrectos o que no se encuentra registrado.
PU-002	Seleccionar el botón agregar rol. Llenar los campos necesarios y otorgarle permisos.	Se registró un nuevo rol en el sistema.
PU-003	Seleccionar el rol a editar, seleccionar el botón editar, editar los campos y permisos necesarios.	Se ha editado los campos y permisos del rol.
PU-004	Seleccionar el rol a eliminar, seleccionar el botón eliminar.	Se ha eliminado un rol con sus permisos.
PU-005	Seleccionar el rol a eliminar, seleccionar el botón eliminar,	El rol no se puede eliminar.
PU-006	Seleccionar agregar usuario, Llenar los campos necesarios y asignar un rol.	Se agrego un nuevo usuario al sistema.
PU-007	Seleccionar el usuario a editar, seleccionar el botón editar, editar los campos necesarios.	Se ha editado un usuario.
PU-008	Seleccionar agregar usuario, ingresar los campos.	Se muestra un aviso que el usuario ya ha sido registrado.
PU-009	Seleccionar agregar usuario, Ingresar los campos.	Se muestra un aviso en caso de no ser un email.
PU-010	Seleccionar agregar usuario, ingresar los campos.	Se muestra un aviso que la contraseña debe ser mayor a 8 caracteres.
PU-011	Seleccionar agregar, ingresar los campos obligatorios.	Se ha agregado el tipo en la tabla administrativa.

PU-012	Seleccionar el tipo a editar, seleccionar el botón editar, editar los campos necesarios.	Se ha editado el tipo de la tabla administrativa.
PU-013	Seleccionar el tipo a eliminar, seleccionar el botón eliminar.	Se ha eliminado el tipo de la tabla administrativa.
PU-014	Seleccionar el agregar, ingresar los campos, seleccionar el botón guardar.	Se muestra un aviso que los campos del tipo en la tabla administrativa son necesarios.
PU-015	Seleccionar el botón agregar docente, ingresar la información, dar clic en guardar.	Se ha agregado la información del docente.
PU-016	Seleccionar el docente a editar, dar clic en editar, editar los campos, y dar clic en guardar.	Se ha editado la información del docente.
PU-017	Seleccionar el docente a eliminar o desactivar, dar clic en el botón eliminar.	Se ha eliminado o desactivado el docente.
PU-018	Seleccionar el botón agregar docente, ingresar la información, dar clic en guardar.	Muestra avisos en cada campo obligatorio.
PU-019	Seleccionar el docente, dar clic en el botón ficha médica.	Despliega la ficha médica del docente seleccionado.
PU-020	Seleccionar el docente, dar clic en el botón ficha médica.	Despliega la ficha médica del docente seleccionado con la información ubicada correcta.
PU-021	Seleccionar el botón agregar estudiante, ingresar la información obligatoria, dar clic en guardar.	Se ha agregado la información del estudiante.
PU-022	Seleccionar el estudiante a editar, dar clic en editar, editar los campos, y dar clic en guardar.	Se ha editado la información del estudiante.
PU-023	Seleccionar el estudiante a eliminar o desactivar, dar clic en el botón eliminar.	Se ha eliminado o desactivado el estudiante.
PU-024	Seleccionar el botón agregar estudiante, ingresar la información obligatoria, dar clic en guardar.	Muestra avisos en cada campo obligatorio.
PU-025	Seleccionar el estudiante, dar clic en el botón ficha médica.	Despliega la ficha médica del estudiante seleccionado.
PU-026	Seleccionar el estudiante, dar clic en el botón ficha médica.	Despliega la ficha médica del estudiante seleccionado con la información ubicada correcta.
PU-027	Seleccionar filtro y dar clic en el botón filtrar.	Los datos se filtraron correctamente.
PU-028	Seleccionar filtro y dar clic en el botón filtrar.	Muestra una lista de los datos que pertenecen al filtro.
PU-029	Dar clic en generar PDF	Muestra el reporte de los datos filtrados.

PU-030	Seleccionar filtro y dar clic en filtrar.	Se mostrará un mensaje diciendo no se encontraron registros.
PU-031	Dar clic en generar PDF	Muestra el reporte de los datos filtrados ubicados correctamente.
PU-032	Seleccionar la opción salir dando clic en el nombre de usuario del sistema.	Se muestra la ventana de Login para el ingreso y registro de usuarios.

Tabla 3.2.18: Pruebas caja negra con las pruebas unitarias.

Elaborado por: Investigador.

C. Implantación

Para la implantación del sistema web se adquirió un servidor en Hostinger.es en el plan sencillo el cual otorga una capacidad de 50 GB en SSD, además Hostinger ofrece dominios el cual se obtuvo uno como unidadeducativagonzalezsuarez.es

Proceso de Configuración

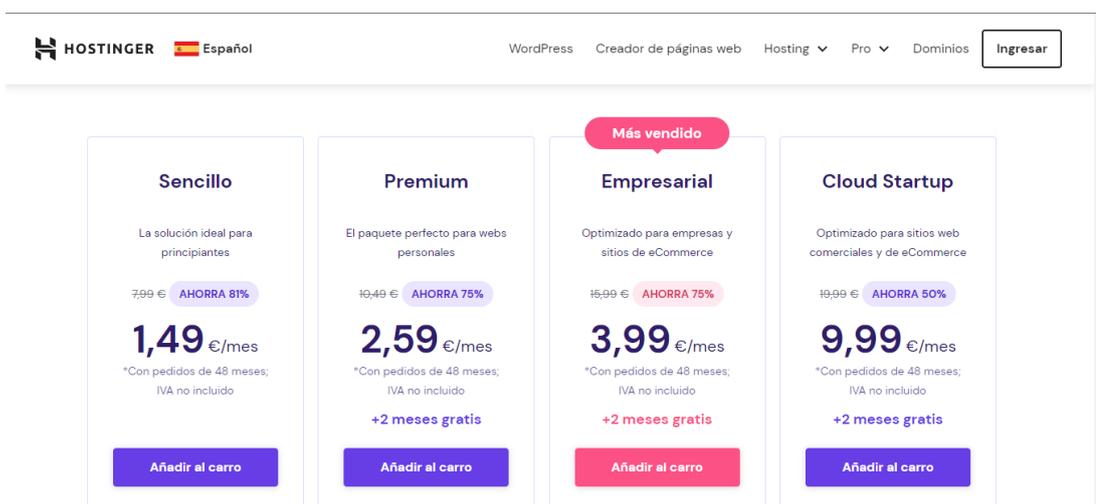


Figura 3.2.55: Contratar plan en Hostinger.es

Elaborado por: Investigador.



Figura 3.2.56: Contratar dominio en Hostinger.

Elaborado por: Investigador.

Una vez contratado tanto el dominio como plan del servidor ingresamos al panel de control con las credenciales o correo con el cual se realizó el registro, como el dominio y el servidor son de la misma empresa estos se conectan automáticamente.

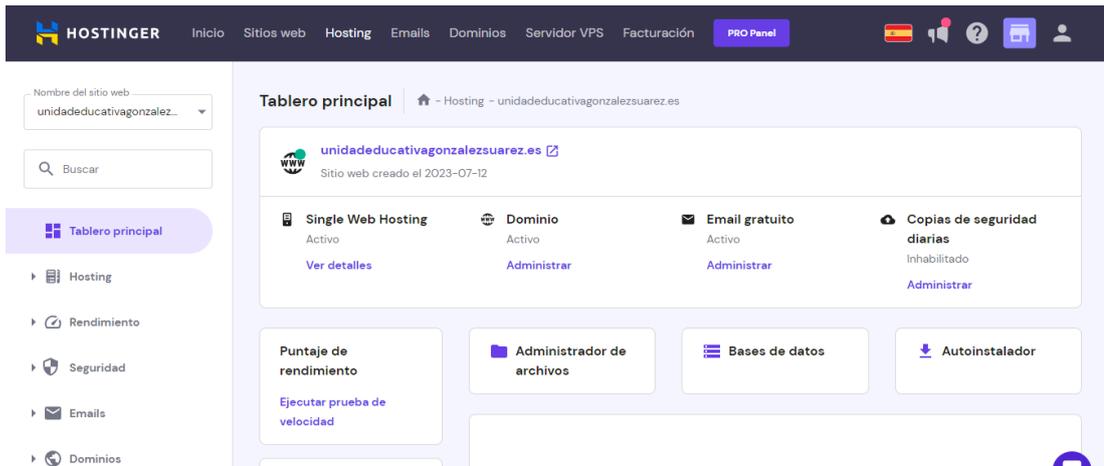


Figura 3.2.57: Panel principal de Hostinger.

Elaborado por: Investigador.

Publicación de la base de datos local a base de datos de Hostinger.

Como Hostinger ocupa un motor de base de datos MySQL se importa la base de datos local.

Administración | 🏠 - Hosting - unidadeducativagonzalezsuarez.es - Bases de datos - Administración

+ Crear nueva base de datos MySQL y usuario de base de datos

Nombre de la base de datos MySQL:

Nombre de usuario MySQL:

Contraseña:

Figura 3.2.58: Creación de la base de datos.

Elaborado por: Investigador.

⚙️ Lista de bases de datos y usuarios actuales de MySQL

Base de Datos MySQL ↕	Usuario MySQL ↕	Creado en ↕	Sitio web
u255927839_uegs_nuevo 3 MB	u255927839_uegs_nuevo	2023-07-13	unidadeducativagonzalezsuarez.es <input type="button" value="Ingresar a phpMyAdmin"/>

Figura 3.2.59: Listado de base de datos en Hostinger.

Elaborado por: Investigador.

Servidor: 127.0.0.1

Bases de datos | SQL | Estado actual | Cuentas de usuarios | Exportar | Importar | Configuración | Replicación | Más

Exportar bases de datos del servidor actual

Exportar plantillas:

Nueva plantilla: Nombre de plantilla:

Plantillas existentes: Plantilla: -- Seleccionar plantilla --

Método de exportación:

Rápido - mostrar sólo el mínimo de opciones de configuración

Personalizado - mostrar todas las opciones de configuración posibles

Formato:

Figura 3.2.60: Exportación de la base de datos local.

Elaborado por: Investigador.

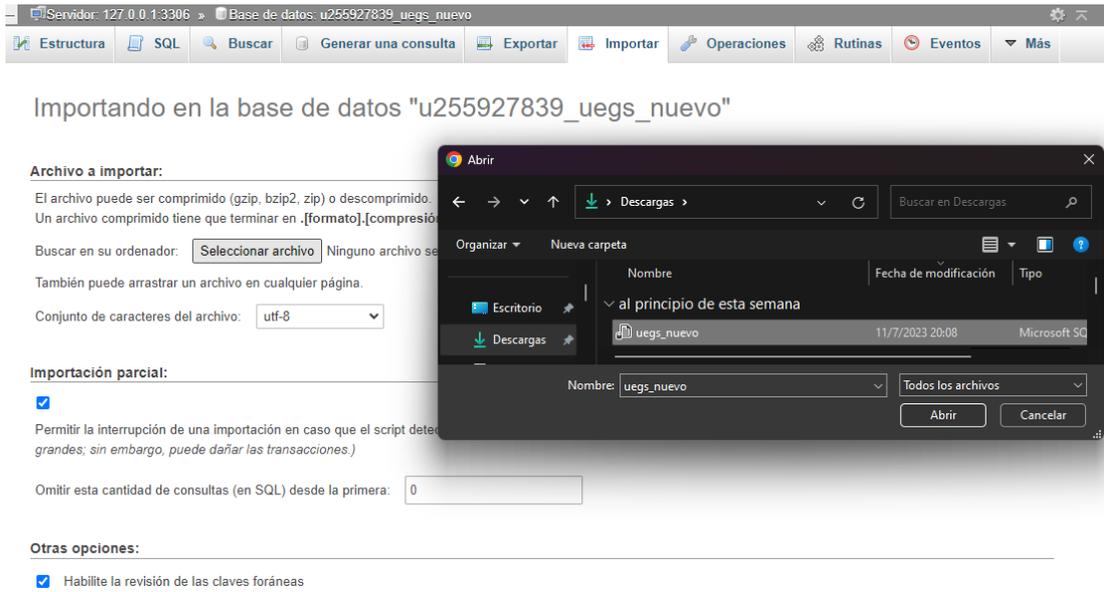


Figura 3.2.61: Importar base de datos a Hostinger.

Elaborador por: Investigador.

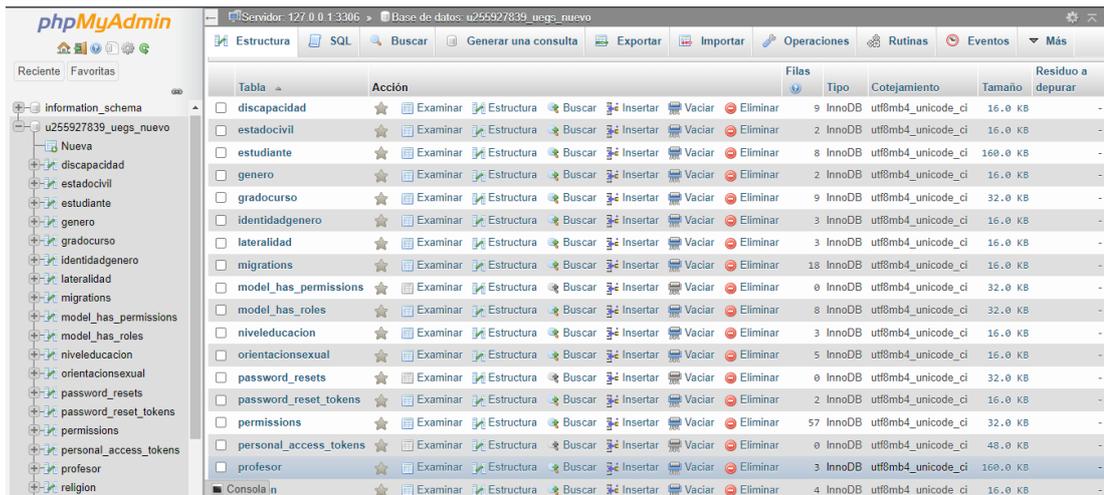


Figura 3.2.62: Base de datos importada a Hostinger correctamente.

Elaborado por: Investigador.

Publicación del proyecto Laravel a Hostinger.

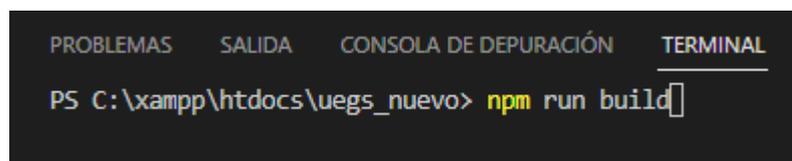


Figura 3.2.63: Comando para puesta en producción del proyecto Laravel.

Elaborado por: Investigador.

Realizar un archivo comprimido en .zip que contenga todas las carpetas y archivos del proyecto Laravel.

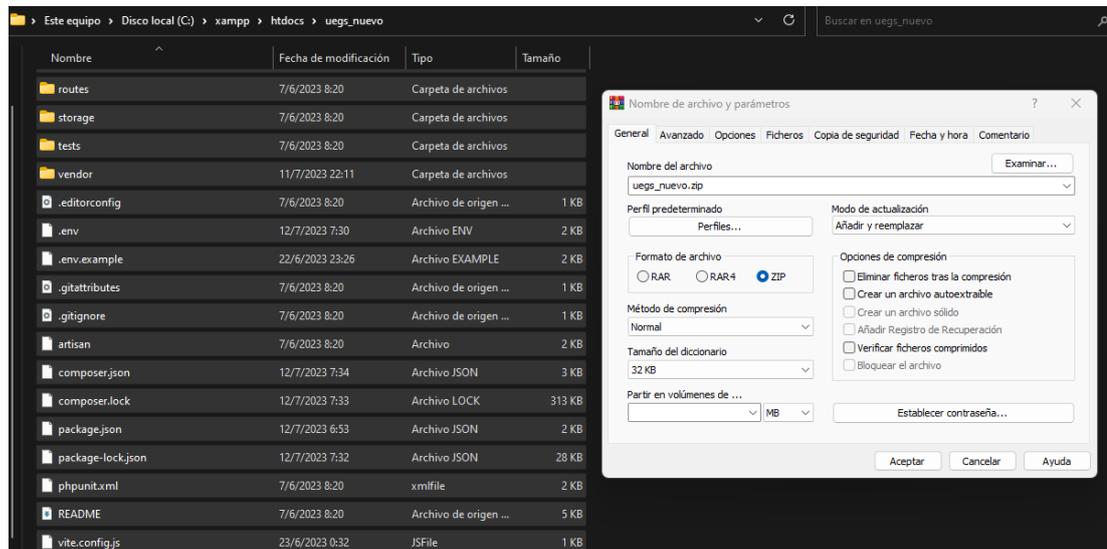


Figura 3.2.64: Creación archivo .zip del proyecto Laravel.

Elaborado por: Investigador.

En Hostinger en el panel de control seleccionar administrador de archivos y subir el archivo .zip y descomprimirlo en la carpeta public_html.

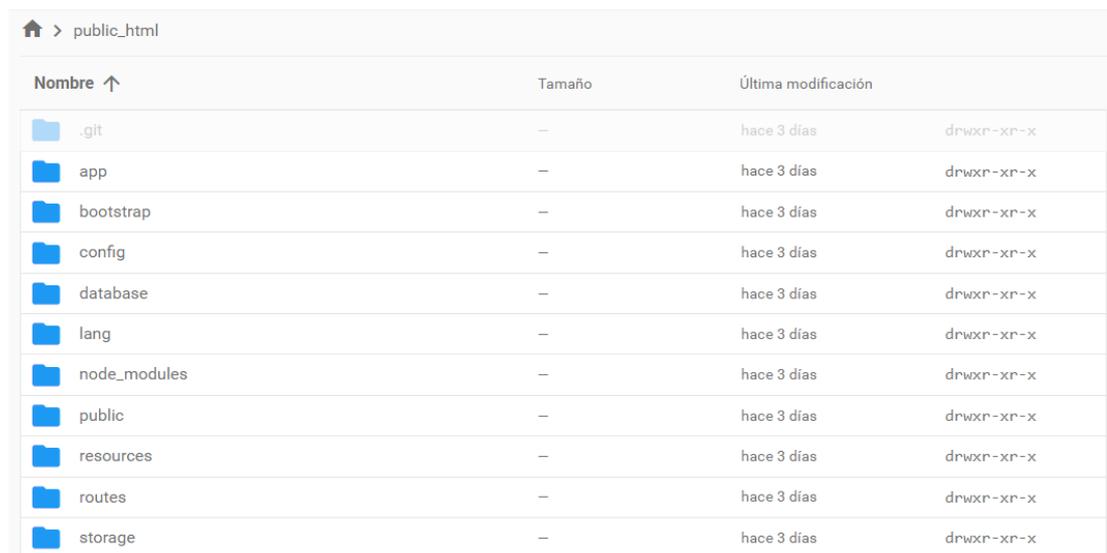


Figura 3.2.65: Archivo .zip descomprimido en la carpeta public_html.

Elaborado por: Investigador.

Configurar archivo .env para la conexión a la base de datos de Hostinger y crear archivo .htaccess para tener acceso a la carpeta public del proyecto Laravel.

```
✕ .env

🏠 > public_html > .env
10
11 DB_CONNECTION=mysql
12 DB_HOST=localhost
13 DB_PORT=3306
14 DB_DATABASE=u255927839_uegs_nuevo
15 DB_USERNAME=u255927839_uegs_nuevo
16 DB_PASSWORD=Danie11805120910*
17
```

Figura 3.2.66: Configuración archivo .env.

Elaborador por: Investigador.

```
✕ .htaccess

🏠 > public_html > .htaccess
1 <IfModule mod_rewrite.c>
2 RewriteEngine On
3 RewriteBase /
4 RewriteRule ^$ public/index.php [L]
5 RewriteRule ^(?:!pubic/).* $ public/$1 [L,NC]
6 </IfModule>|
```

Figura 3.2.67: Configuración archivo .htaccess.

Elaborado por: Investigador.

Realizada las configuraciones de la Figura 3.2.49 y de la Figura 3.2.50 podemos ingresar al dominio y verificar la publicación del proyecto.



Figura 3.2.68: Inicio del sistema con dominio unidadeducativagonzalezsuarez.es

Elaborador por: Investigador.

Posterior a la implantación del sistema, se procedió a la capacitación del departamento médico. La capacitación estuvo conformada por distintos temas los cuales se detallan en la **Tabla 3.2.21**.

Tema	Instructor	Dirigido a:	Fecha Inicio	Fecha Fin	07/07/2023
Registro de Usuarios al Sistema	Jonathan Nuñez	Departamento Médico	07/07/2023	07/07/2023	
Ingreso al sistema	Jonathan Nuñez	Departamento Médico	07/07/2023	07/07/2023	
Restablecimiento de contraseñas	Jonathan Nuñez	Departamento Médico	07/07/2023	07/07/2023	
Gestión de Docentes	Jonathan Nuñez	Departamento Médico	07/07/2023	07/07/2023	
Gestión de Estudiantes	Jonathan Nuñez	Departamento Médico	07/07/2023	07/07/2023	
Gestión tablas Administrativas	Jonathan Nuñez	Departamento Médico	07/07/2023	07/07/2023	
Reportes por Filtrado	Jonathan Nuñez	Departamento Médico	07/07/2023	07/07/2023	
Cierre del Sistema	Jonathan Nuñez	Departamento Médico	07/07/2023	07/07/2023	

Tabla 3.2.19: Cronograma capacitación del sistema.

Elaborador por: Investigador.

Además, se cuenta con un manual de usuario, en el cual se indica las funcionalidades del sistema. (ver Anexo A)

CAPITULO IV.- CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- Aplicadas las herramientas de recolección de la información al departamento médico, docentes y estudiantes de la Unidad Educativa González Suárez se evidenció los problemas que se tiene al archivar la ficha médica, en el caso de los docentes se cuenta con un formato ya establecido pero existe perdida, duplicidad y falta de información, en el caso de los estudiantes no se cuenta con un formato establecido para la ficha médica por lo que se dificulta la atención al estudiante ya que no se tiene la información necesaria para tratarlo en caso de alguna emergencia dentro de la institución. Para esto se actualizo los procesos para llenar la información de las fichas médicas tanto de docentes como estudiantes.
- La tecnología empleada para el desarrollo fue con el Framework Laravel el mismo que se basa en la arquitectura MVC lo que promueve una separación clara de la lógica de negocio y la presentación, esto facilita la organización y el mantenimiento del código, así como la reutilización de componentes. Además, se cuenta con una gran variedad de paquetes y bibliotecas que permite ahorrar tiempo al rato de codificar.
- La metodología AUP establecida para gestionar las actividades del proyecto fue la más óptima ya que se trabajó en conjunto con el departamento médico de la institución y en caso de existir cambios en los entregables se los realizó de manera rápida logrando así cumplir con los requerimientos del usuario.

Recomendaciones

- Se recomienda utilizar técnicas de recolección de datos aplicadas en el proyecto ya que así se puede tener información necesaria para la implementación.
- Se recomienda revisar la documentación de Laravel ya que ayuda a entender la estructura y funcionamiento de los paquetes, Además brinda ayuda detallada sobre el uso y configuración para las bibliotecas y componentes, por ejemplo, integrar una base de datos por migraciones.
- Se recomienda subir a un hosting el proyecto y la base de datos, modificar el archivo .env con las credenciales de la base de datos del servidor para así evitar errores de permisos o compatibilidad con el sitio web.

- Se recomienda trabajar en conjunto con los usuarios el diseño de interfaces y reportes ya que en caso de existir cambios en el sistema se lo pueda realizar de manera rápida.

Referencias Bibliográficas

- [1] D. Luna, E. Soriano, y F. González, «Historia clínica electrónica», Buenos Aires, dic. 2007. Accedido: 16 de noviembre de 2022. [En línea]. Disponible en: https://www.hospitalitaliano.org.ar/multimedia/archivos/servicios_attachs/5056.pdf
- [2] E. L. Seguimiento *et al.*, «UNIVERSIDAD TÉCNICA DE AMBATO FACULTAD DE CIENCIAS DE LA SALUD CARRERA DE MEDICINA INFORME DE INVESTIGACIÓN SOBRE Requisito previo para optar el Título de Médico. "BENEFICIOS DEL USO DE LA HISTORIA CLÍNICA ELECTRÓNICA EN», 2013.
- [3] E. Manzano, «SISTEMA WEB BASADO EN ARQUITECTURA MVC PARA LA GESTIÓN DE INSUMOS PRIMARIOS PARA LA PLANIFICACIÓN DEL SISTEMA ELÉCTRICO DE DISTRIBUCIÓN DE LA EMPRESA ELÉCTRICA AMBATO (EEASA)», Universidad Técnica de Ambato, Ambato, 2022. Accedido: 23 de noviembre de 2022. [En línea]. Disponible en: <https://repositorio.uta.edu.ec/bitstream/123456789/36638/1/t2139ti.pdf>
- [4] K. Chacha, «SISTEMA WEB APLICANDO ARQUITECTURA MODELO VISTA CONTROLADOR (MVC) PARA EL CONTROL DE INVENTARIO Y PRODUCCIÓN EN LA PASTEURIZADORA J.S DEL CANTÓN SALCEDO.», Universidad Técnica de Ambato, Ambato, 2022. Accedido: 23 de noviembre de 2022. [En línea]. Disponible en: <https://repositorio.uta.edu.ec/bitstream/123456789/36643/1/t2141ti.pdf>
- [5] E. Jiménez, «APLICACIÓN WEB DE CATÁLOGOS PRIVADOS UTILIZANDO LA ARQUITECTURA MVC PARA LA EMPRESA TURBOMEKANICS S.A. DE LA CIUDAD DE AMBATO», Universidad Técnica de Ambato, Ambato, 2017. Accedido: 23 de noviembre de 2022. [En línea]. Disponible en: https://repositorio.uta.edu.ec/bitstream/123456789/26218/1/Tesis_t1293si.pdf
- [6] E. Chanatasig, «SISTEMA WEB MULTIPLATAFORMA BASADO EN LA ARQUITECTURA MODELO VISTA CONTROLADOR (MVC) Y CLOUD COMPUTING PARA LA GESTIÓN DE ACTIVIDADES DE NOVADENT-ESPECIALIDADES ODONTOLÓGICAS EN LA CIUDAD DE LATACUNGA», Universidad Técnica de Ambato, Ambato, 2022. Accedido: 23 de noviembre de 2022. [En línea]. Disponible en: <https://repositorio.uta.edu.ec/bitstream/123456789/36647/1/t2144ti.pdf>
- [7] «Modelo vista controlador (MVC). Servicio de Informática ASP.NET MVC 3 Framework». <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html> (accedido 30 de noviembre de 2022).
- [8] M. DE Desarrollo De Software y E. Gabriel Pacienza, «FACULTAD DE QUÍMICA E INGENIERIA "FRAY ROGELIO BACON" PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA SANTA MARIA DE LOS BUENOS AIRES Cátedra Seminario de Sistemas».

- [9] L. R. Capote, «METODOLOGÍAS TRADICIONALES DE DESARROLLO DE SOFTWARE VS METODOLOGÍAS ÁGILES».
- [10] E. Flores y J. Cordero, «UNIVERSIDAD UNION BOLIVARIANA CARRERA DE INGENIERIA DE SISTEMAS METODOLOGIAS AGILES PROCESO UNIFICADO AGIL (AUP) MATERIA : INGENIERIA SOFTWARE», Accedido: 18 de julio de 2023. [En línea]. Disponible en: <https://ingenieriadesoftware.mex.tl/images/18149/METODOLOGIAS%20AGILES.pdf>
- [11] M. Centeno y A. Manuel, «UNIDAD ACADÉMICA DE INGENIERÍA CIVIL CARRERA DE INGENIERÍA DE SISTEMAS MACHALA 2018», Accedido: 18 de julio de 2023. [En línea]. Disponible en: http://repositorio.utmachala.edu.ec/bitstream/48000/12240/1/TUAIC_2017_IS_CD0045.pdf
- [12] Ramos. Benito, *Control de Calidad de la Atención de Salud*, 2.ª ed. 2011. Accedido: 30 de noviembre de 2022. [En línea]. Disponible en: https://files.sld.cu/scap/files/2012/01/control_calidad.pdf
- [13] D. De, A. De, L. A. Calidad, y S. Planificacion, «MINISTERIO DE SALUD PUBLICA MANUAL DEL MANEJO, ARCHIVO DE LAS HISTORIAS CLINICAS», 2007.
- [14] A. TACURI, «SISTEMA DE INFORMACIÓN UTILIZANDO LA ARQUITECTURA MVC PARA EL SEGUIMIENTO Y MONITOREO DE LOS CRÉDITOS DE DESARROLLO HUMANO EN EL MINISTERIO DE INCLUSIÓN ECONÓMICA Y SOCIAL (MIES) DE LA PROVINCIA DE PASTAZA.», UNIVERSIDAD REGIONAL AUTÓNOMA DE LOS ANDES UNIANDES, PUYO, 2018. Accedido: 11 de mayo de 2023. [En línea]. Disponible en: <https://dspace.uniandes.edu.ec/bitstream/123456789/7750/1/PIUPSIS0002-2018.pdf>

ANEXOS

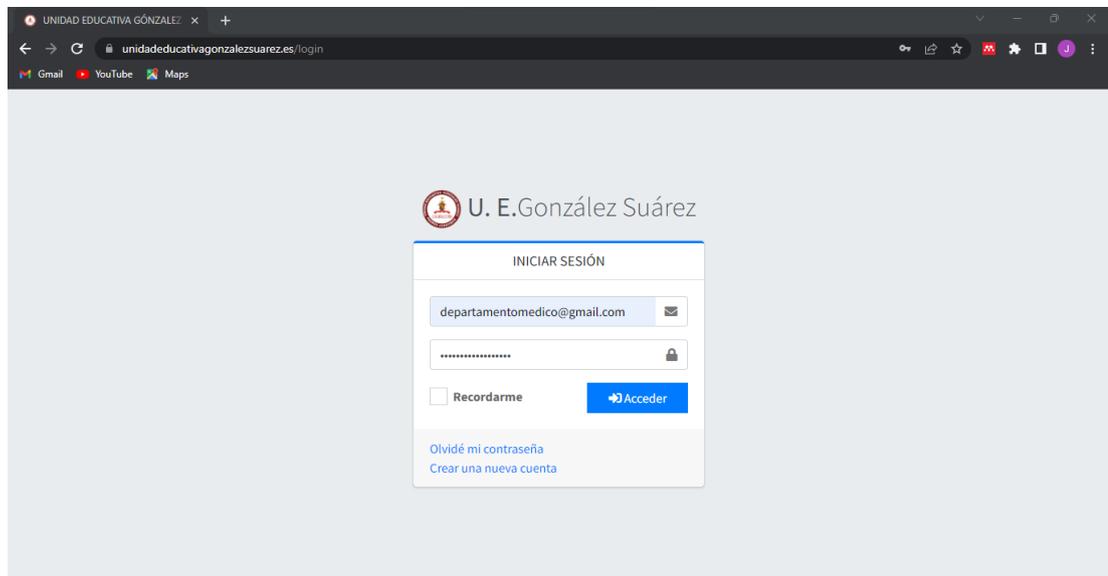
ANEXO A. Manual de Usuario

Departamento Médico (gestión docente, gestión estudiante, generar reporte, visualizar gráficas).

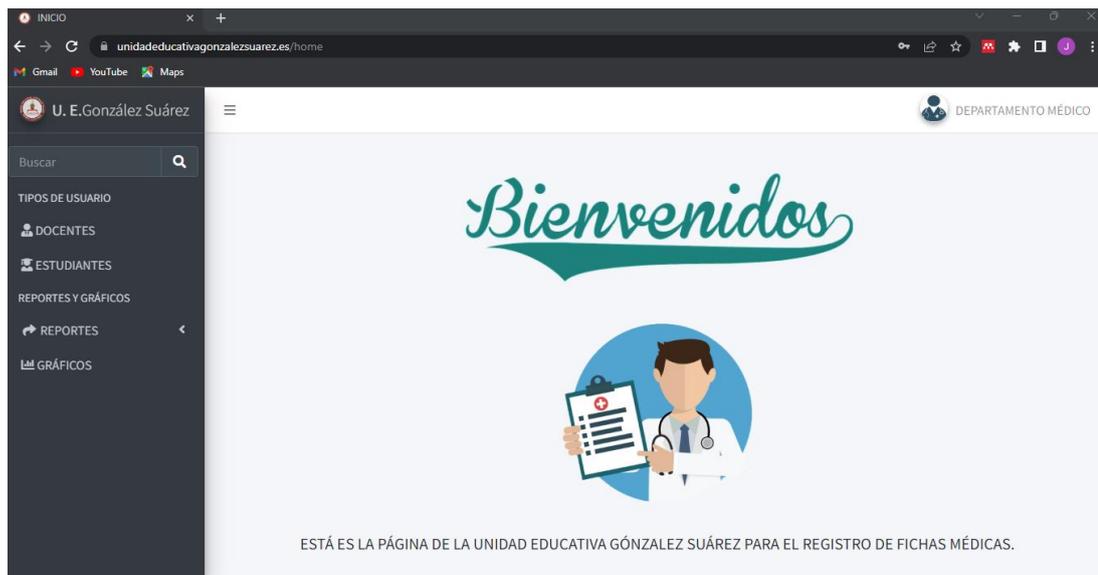
1. Ingresar a unidadeducativagonzalezsuarez.es.



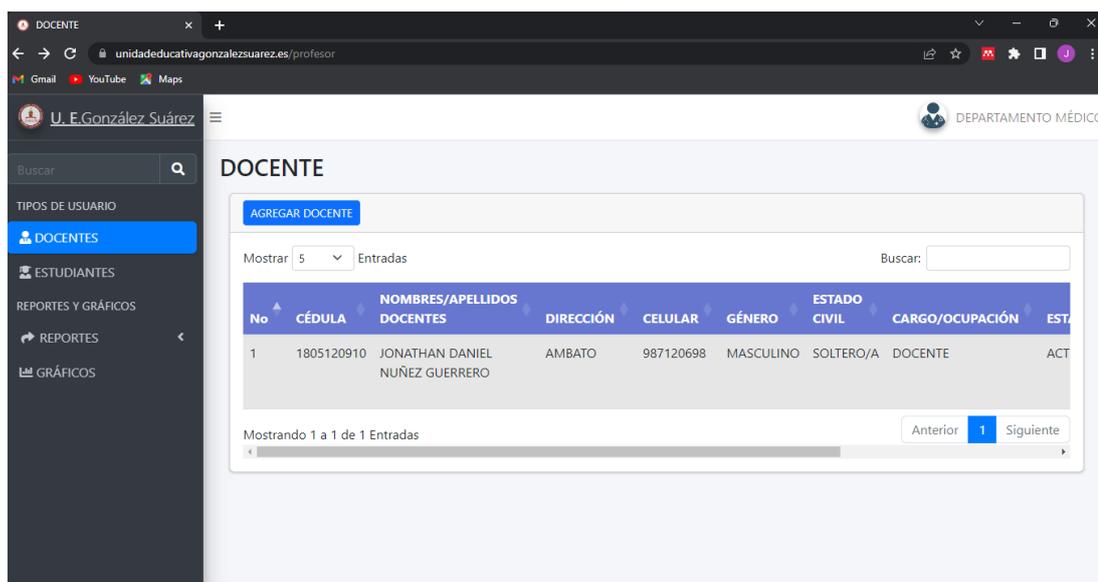
2. Dar clic en Iniciar Sesión e ingresar sus credenciales.



3. Ingresado al sistema podrá ver la pagina de inicio y en la parte izquierda los módulos con los que podrá trabajar.



4. Al ingresar al módulo de docentes o estudiantes podrá ver una lista de los docentes o estudiantes que se han registrado en el sistema.



5. Para registrar la información de nuevos docentes o estudiantes dar clic en agregar docente o estudiante, el cual redireccionará a un formulario.

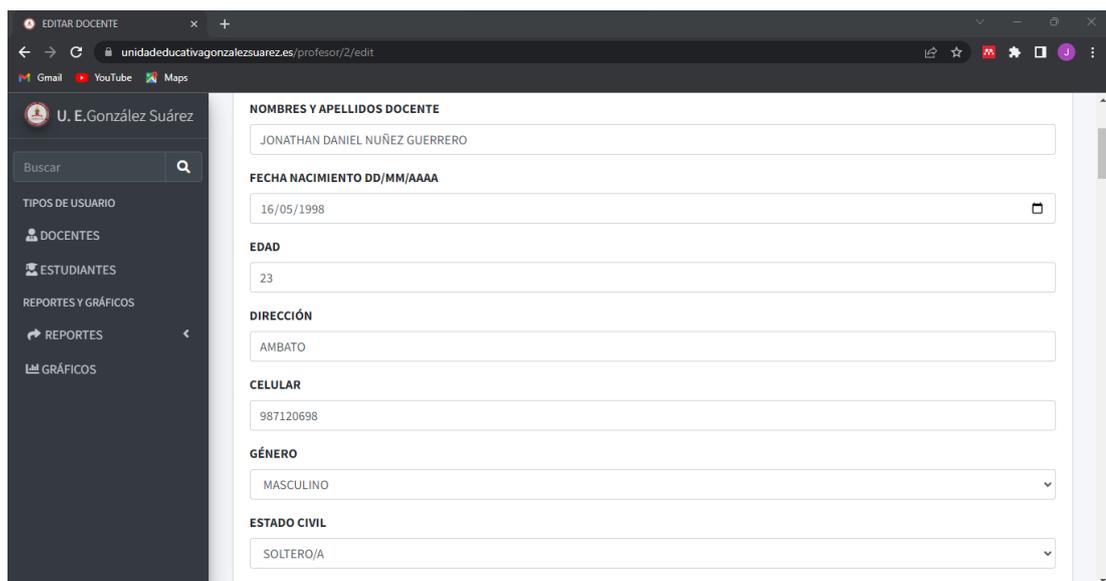
6. Dar clic en Guardar y se redireccionara a la pantalla donde se puede observar el listado de docentes o estudiantes registrados.

DIRECCIÓN	CELULAR	GÉNERO	ESTADO CIVIL	CARGO/OCUPACIÓN	ESTADO	VER	EDITAR	FICHA MÉDICA	ELIMINAR
AMBATO	987120698	MASCULINO	SOLTERO/A	DOCENTE	ACTIVO	VER	EDITAR	FICHA MÉDICA	ELIMINAR

7. Dando clic en el botón ver se podrá observar toda la información del docente o estudiante.



8. Dando clic en el botón editar se abrirá un formulario el cual tendrá toda la información del docente o estudiante, aquí se podrá modificar los campos según como sea necesario.



9. Dando clic en el botón de ficha médica se genera el documento con el formato de ficha médica del docente o estudiante.

FICHA MÉDICA JONATHAN DANIEL NUÑEZ GUERRERO

unidadeducativagonzalezsuarez.es/profesor/pdf/2

FICHA MÉDICA JONATHAN DANIEL NUÑEZ GUERRERO

UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ

FICHA MÉDICA DOCENTE

A. DATOS PERSONALES DEL DOCENTE				
INSTITUCIÓN	CÉDULA	NOMBRES Y APELLIDOS	GÉNERO	CARGO/OCUPACIÓN
UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ	185120910	JONATHAN DANIEL NUÑEZ GUERRERO	MASCULINO	DOCENTE
B. DATOS ADICIONALES DEL DOCENTE				
FECHA DE NACIMIENTO	EDAD	DIRECCIÓN	RELIGIÓN	
16/05/1998	23	AMBATO	CATÓLICO	
TIPO DE SANGRE	LATERALIDAD	ORIENTACION SEXUAL	IDENTIDAD DE GÉNERO	
A+	IZQUIERDO/A	HETEROSEXUAL	HOMBRE	
C. ANTECEDENTES PATOLÓGICOS				
ANTECEDENTES PATOLÓGICOS PERSONALES				
PATOLOGÍA	SÍ/NO	DESCRIPCIÓN		
DIABETES	NO	NINGUNA		
LUMBALGIA	NO	NINGUNA		
ENF. RESPIRATORIAS	NO	NINGUNA		
ENF. OSTIOMESCLARES	NO	NINGUNA		
HIPOCUSIA	NO	NINGUNA		
CEJUGIA	NO	NINGUNA		
DISCAPACIDAD	NINGUNA	NINGUNA		
ALERGIA	NO	NINGUNA		
ALERGIA MEDICAMENTO	NO	NINGUNA		
ENF. COMUNES	NO	NINGUNA		
OTRAS PATOLOGÍAS	NO	NINGUNA		

10. Dando clic en el botón Eliminar el estado del docente o estudiante cambia a inactivo ya que una ficha médica en la institución no puede ser eliminada en su totalidad.

DOCENTE

unidadeducativagonzalezsuarez.es/profesor

U. E. González Suárez

DEPARTAMENTO MÉDICO

DOCENTE

AGREGAR DOCENTE

Buscar:

DIRECCIÓN	CELULAR	GÉNERO	ESTADO CIVIL	CARGO/OCUPACIÓN	ESTADO				
AMBATO	987120698	MASCULINO	SOLTERO/A	DOCENTE	INACTIVO	VER	EDITAR	FICHA MÉDICA	ELIMINAR

Anterior 1 Siguiente

11. En el módulo de reportes se podrá genera reportes con datos filtrados según sea necesario para lo cual tenemos las opciones de tipo de sangre, género y edad.

U. E. González Suárez DEPARTAMENTO MÉDICO

ESTUDIANTES X TIPO DE SANGRE

A+

Filtrar

NOMBRES	TIPO SANGRE
JONATHAN DANIEL NÚÑEZ GUERRERO	A+
ANTHONY ALEJANDRO NÚÑEZ GUERRERO	A+
CARLOS IVAN GUERRERO NÚÑEZ	A+

Generar PDF

Una vez filtrados los datos se podrá generar el reporte deseado.

UNIDAD EDUCATIVA GÓNZALEZ SUÁREZ
REPORTE ESTUDIANTES

NOMBRES/APELLIDOS DEL ESTUDIANTE	TIPO SANGRE
JONATHAN DANIEL NÚÑEZ GUERRERO	A+
ANTHONY ALEJANDRO NÚÑEZ GUERRERO	A+
CARLOS IVAN GUERRERO NÚÑEZ	A+

Reporte con filtros de grado y tipo de sangre

U. E. González Suárez DEPARTAMENTO MÉDICO

ESTUDIANTES X GRADO/CURSO X TIPO DE SANGRE

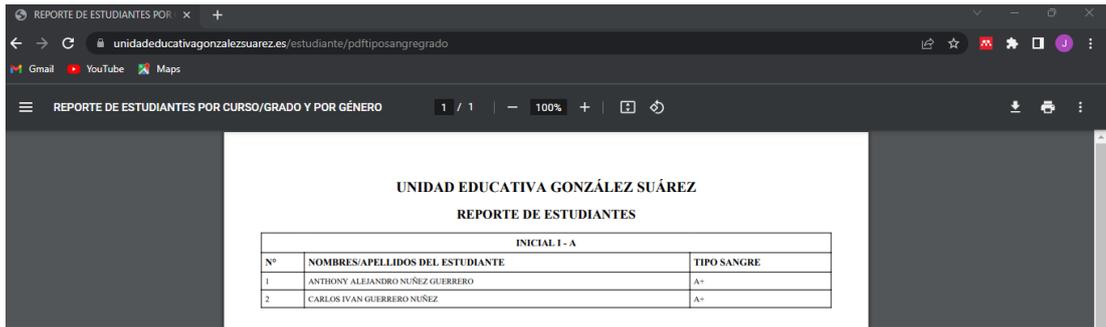
INICIAL I - A

A+

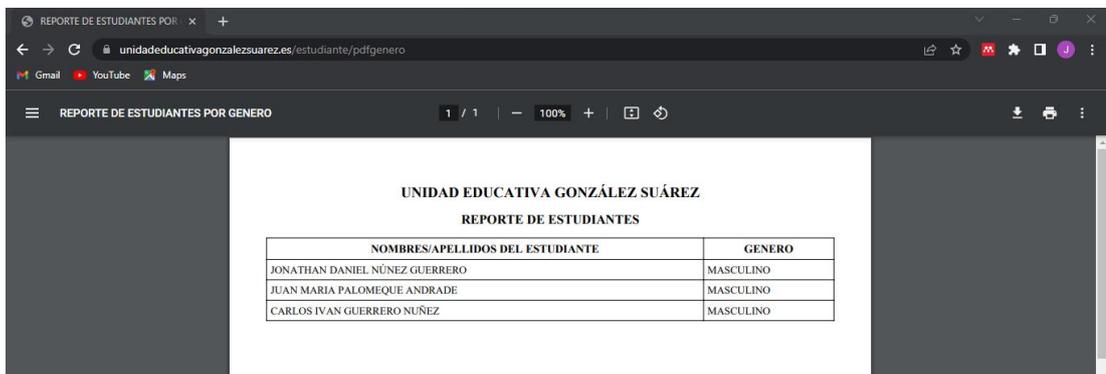
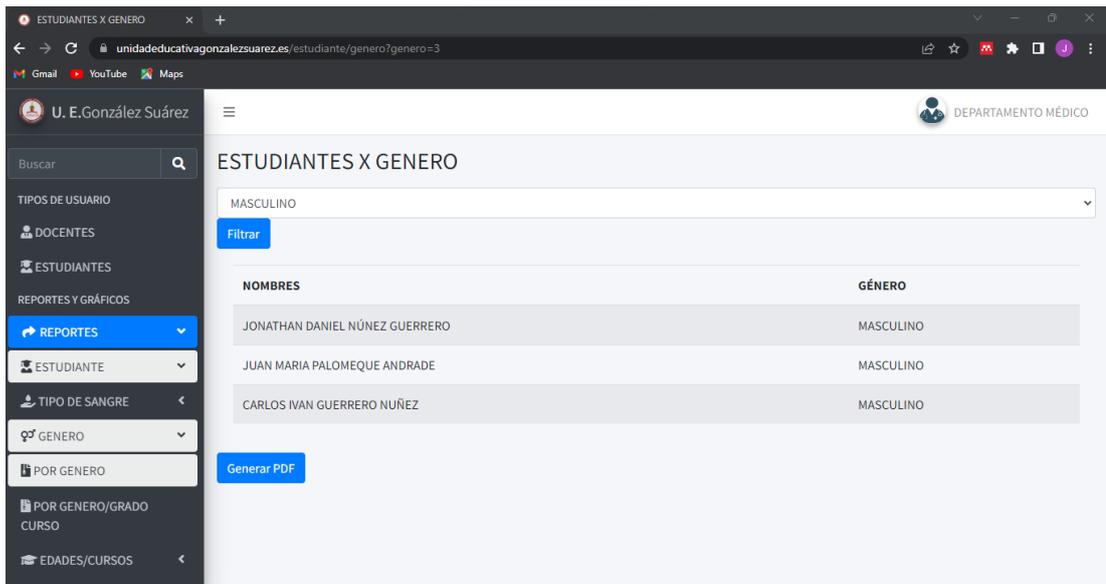
Filtrar

N°	NOMBRES	GRADO/CURSO	TIPO SANGRE
1	ANTHONY ALEJANDRO NÚÑEZ GUERRERO	INICIAL I - A	A+
2	CARLOS IVAN GUERRERO NÚÑEZ	INICIAL I - A	A+

Generar PDF



Reporte por genero



Reporte de estudiantes por género y curso.

ESTUDIANTES X GENERO

unidadeducativagonzalezsuarez.es/estudiante/genero?grado=1&genero=3

U. E. González Suárez DEPARTAMENTO MÉDICO

ESTUDIANTES X GRADO/CURSO X GENERO

INICIAL I - A

MASCULINO

Filtrar

Nº	NOMBRES	GRADO/CURSO	GÉNERO
1	CARLOS IVAN GUERRERO NUÑEZ	INICIAL I - A	MASCULINO

Generar PDF

REPORTE DE ESTUDIANTES POR

unidadeducativagonzalezsuarez.es/estudiante/pdfgenero?grado

REPORTE DE ESTUDIANTES POR CURSO/GRADO Y POR GÉNERO 1 / 1 100%

UNIDAD EDUCATIVA GÓNZALEZ SUÁREZ

REPORTE ESTUDIANTES

INICIAL I - A

Nº	NOMBRES/APELLIDOS DEL ESTUDIANTE	GENERO
1	CARLOS IVAN GUERRERO NUÑEZ	MASCULINO

Reporte por grado o curso y edades de los estudiantes

ESTUDIANTES EDADES X CURSO

unidadeducativagonzalezsuarez.es/estudiante/edadcurso?grado=1

U. E. González Suárez DEPARTAMENTO MÉDICO

ESTUDIANTES X CURSO X EDAD

Seleccionar grado/curso

Filtrar

Nº	NOMBRES	EDAD
1	PEPE MANOLO ALEJANDRO NUÑEZ	25
2	ANTHONY ALEJANDRO NUÑEZ GUERRERO	21
3	CARLOS IVAN GUERRERO NUÑEZ	26

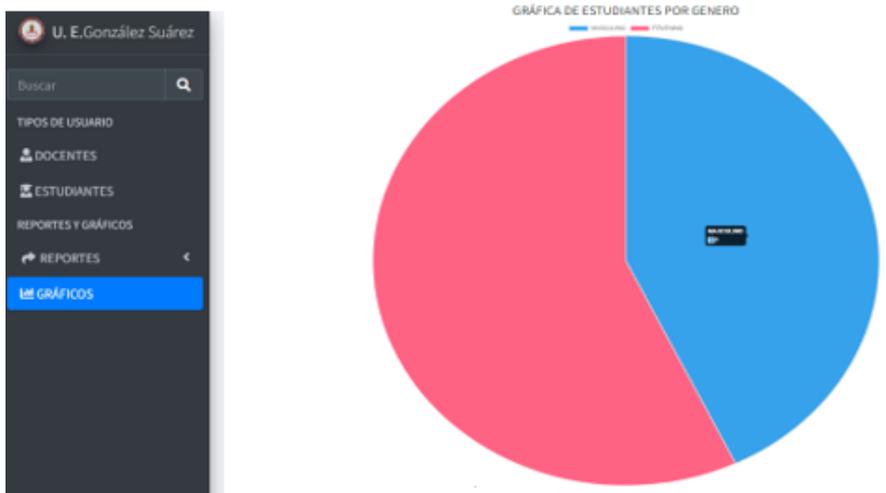
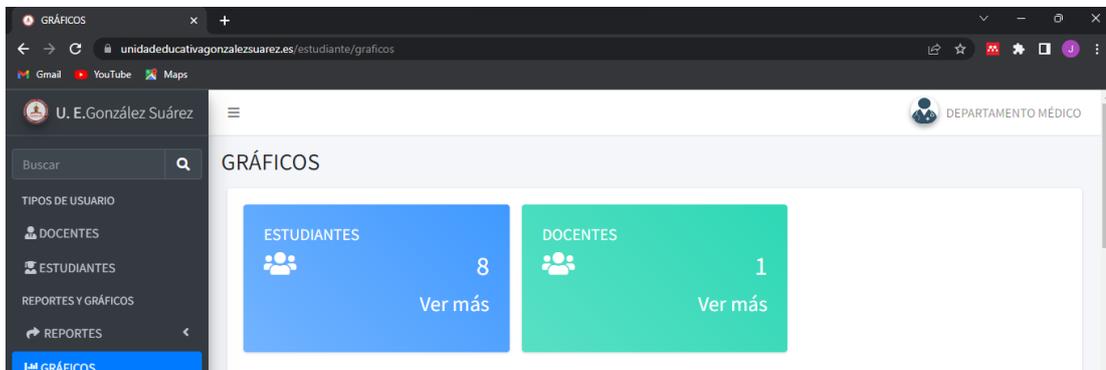
Generar PDF

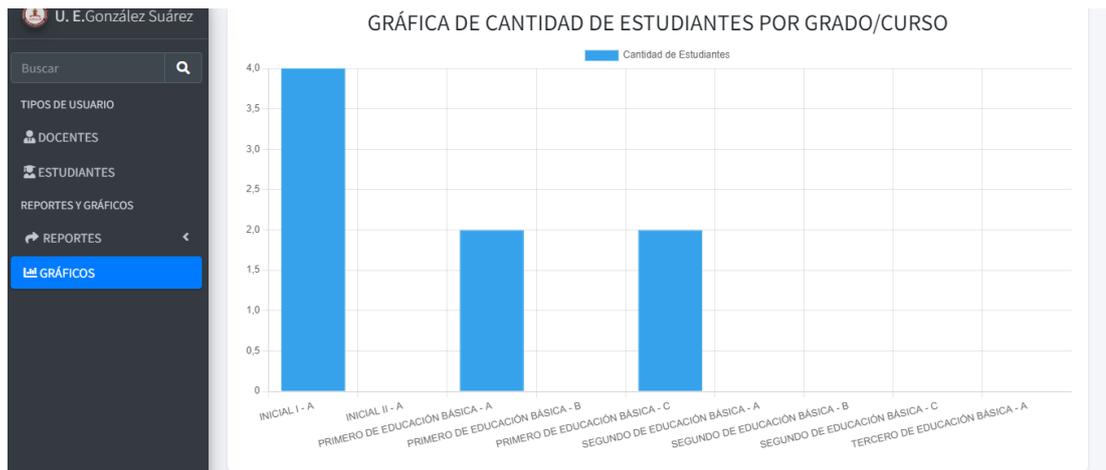
REPORTE DE ESTUDIANTES EDADES POR CURSO/GRADO

UNIDAD EDUCATIVA GONZÁLEZ SUÁREZ
 REPORTE DE ESTUDIANTES

INICIAL I - A		
Nº	NOMBRES/APELLIDOS DEL ESTUDIANTE	EDAD
1	PEPE MANOLO ALEJANDRO NUÑEZ	25
2	ANTHONY ALEJANDRO NUÑEZ GUERRERO	21
3	CARLOS IVAN GUERRERO NUÑEZ	26

12. En el módulo de gráficos se podrá observar la cantidad de estudiantes y docentes registrados, así como gráficos de estudiantes por genero y otro de cantidad de estudiantes por grado o curso.





Administrador (gestión tablas administrativas, gestión docente, gestión estudiante, generar reporte, visualizar gráficos).

13. Para la gestión de docentes, estudiantes, generar reportes, y visualizar gráficos seguir los pasos de departamento médico, pero con el registro de administrador.

14. Registro de roles y usuarios.

UNIDAD EDUCATIVA GÓNZALEZ: x +

unidadeducativagonzalezsuarez.es/roles

U. E. González Suárez ADMINISTRADOR

ROLES REGISTRADOS

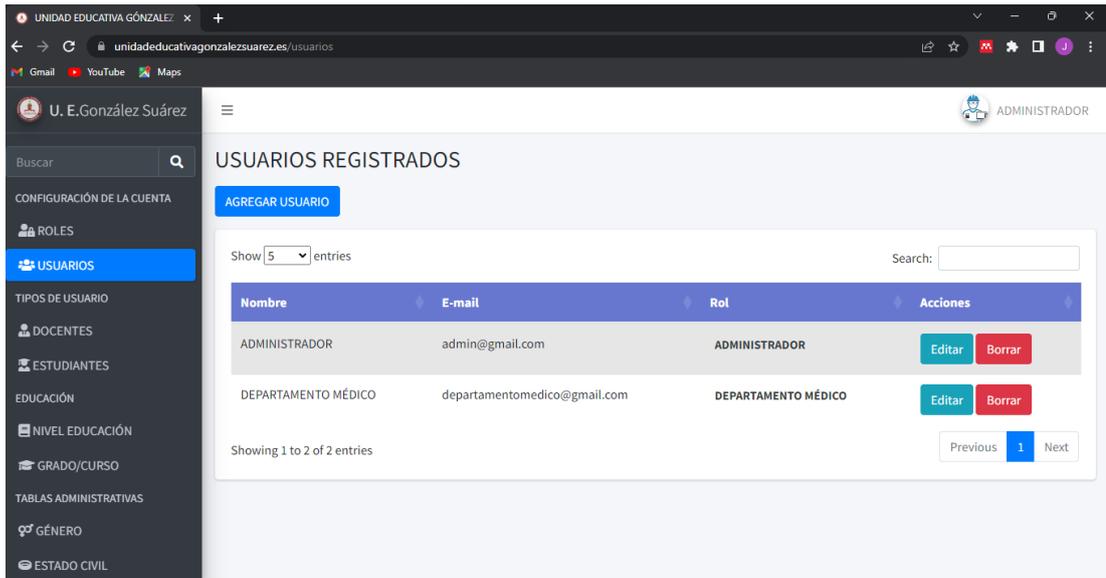
AGREGAR ROL

Mostrar 5 Entradas

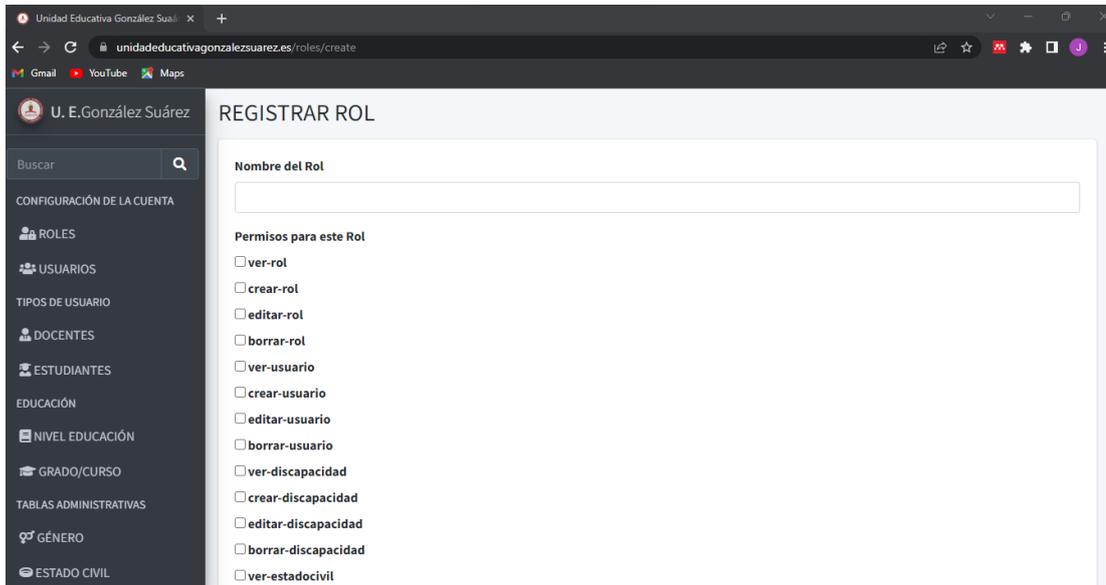
ROL	ACCIONES
ADMINISTRADOR	Editar ELIMINAR
DEPARTAMENTO MÉDICO	Editar ELIMINAR
DOCENTE	Editar ELIMINAR
ESTUDIANTE	Editar ELIMINAR

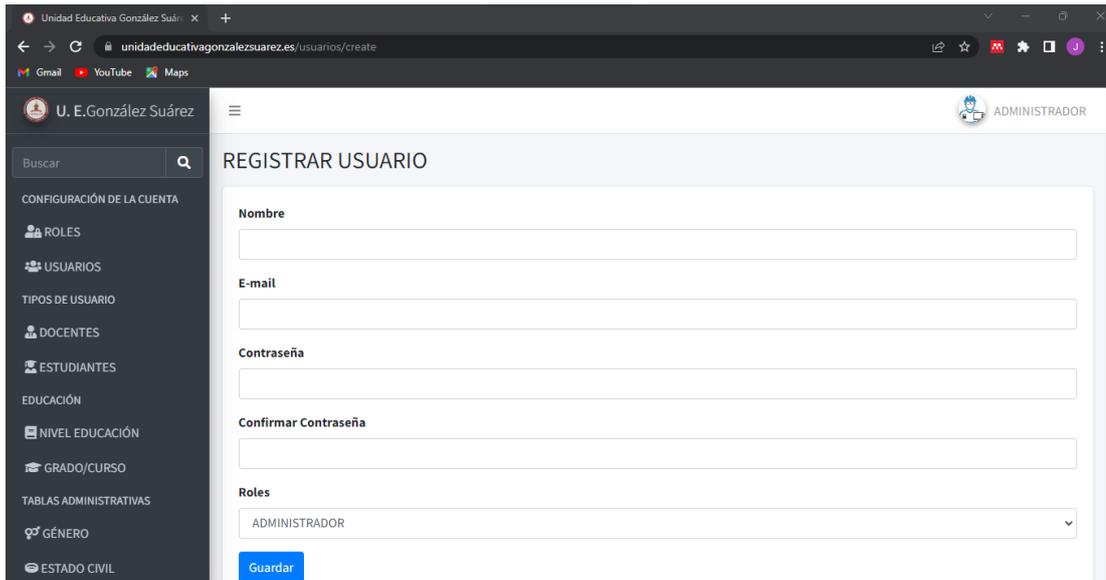
Mostrando 1 a 4 de 4 Entradas

Anterior 1 Siguiente

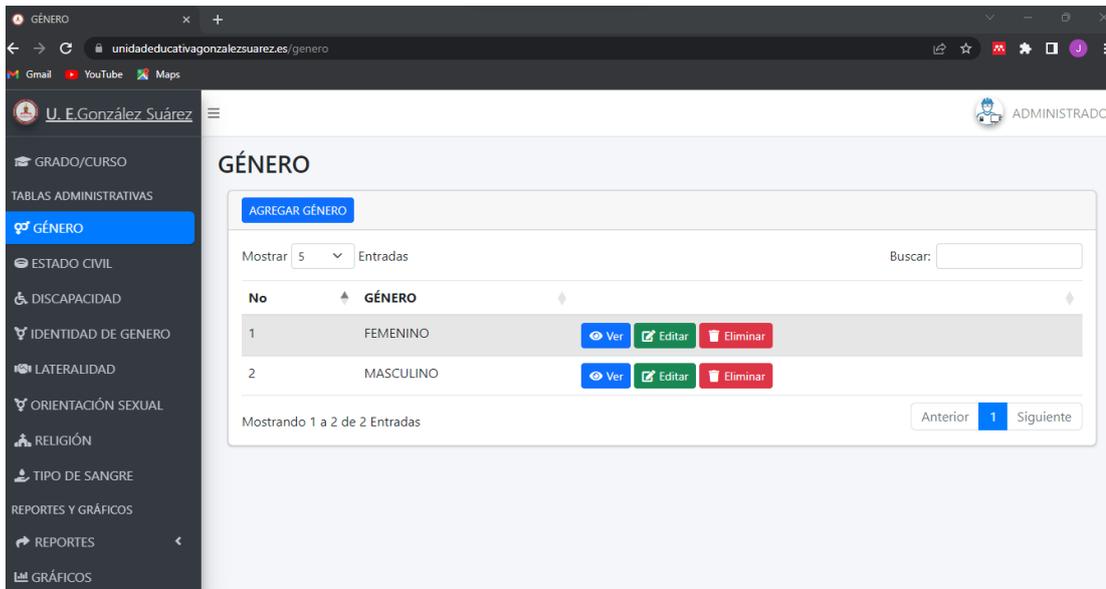


15. Creación de roles con sus permisos y usuarios.





16. Tablas Administrativas



U. E. González Suárez

DISCAPACIDAD

AGREGAR DISCAPACIDAD

Mostrar: Todo Entradas

Buscar:

No	TIPO DE DISCAPACIDAD	Ver	Editar	Eliminar
1	DISCAPACIDAD FÍSICA	Ver	Editar	Eliminar
2	DISCAPACIDAD INTELECTUAL	Ver	Editar	Eliminar
3	DISCAPACIDAD MENTAL	Ver	Editar	Eliminar
4	DISCAPACIDAD PSICOSOCIAL	Ver	Editar	Eliminar
5	DISCAPACIDAD MÚLTIPLE	Ver	Editar	Eliminar
6	DISCAPACIDAD SENSORIAL	Ver	Editar	Eliminar
7	DISCAPACIDAD AUDITIVA	Ver	Editar	Eliminar
8	DISCAPACIDAD VISUAL	Ver	Editar	Eliminar

17. Agregar tipos en tablas administrativas.

U. E. González Suárez

AGREGAR GÉNERO

GÉNERO

EJEMPLO: MASCULINO

[GUARDAR](#)

U. E. González Suárez

AGREGAR DISCAPACIDAD

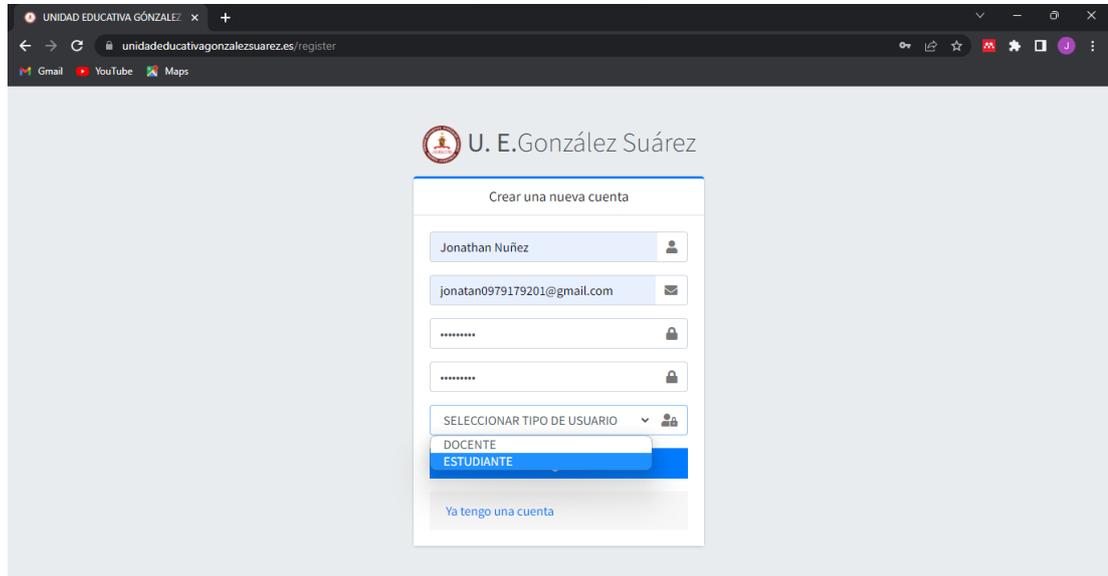
TIPO DE DISCAPACIDAD

EJEMPLO: DISCAPACIDAD VISUAL

[GUARDAR](#)

Docentes y Estudiantes (gestión docente y estudiante solo con permisos de registro de información).

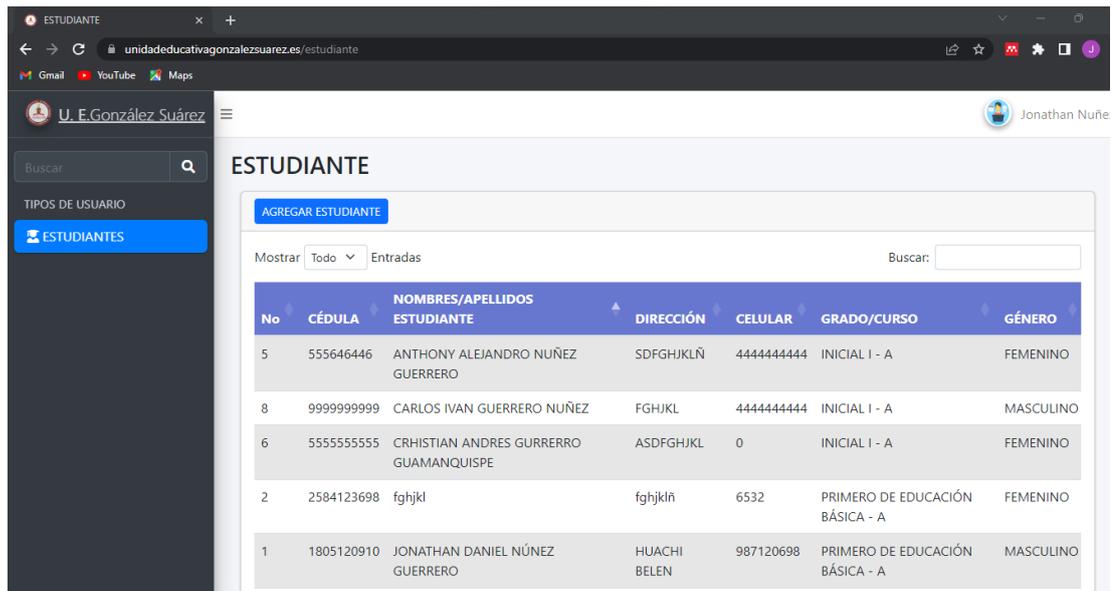
18. Registrarse como docente o estudiante.



19. Ingresar al sistema con el correo y contraseña del registro.



20. Seleccionar el modulo de docente o estudiante según el usuario que se haya registrado.



21. Seleccionar Agregar docente o estudiante según el usuario que se haya seleccionado, agregar la información correspondiente y dar clic en el botón guardar.

